Федеральное агентство по рыболовству
БГАРФ ФГБОУ ВО «КГТУ»
Калининградский морской рыбопромышленный колледж

Утверждаю
Заместитель начальника колледжа
по учебно-методической работе
А.И.Колесниченко

Подписано цифровой подписью: Колесниченко Андрей Игоревич
Причина: Я являюсь автором этого документа
Дата: 19 Август 2025 г. 12:52:50
DN: E=uc_fk@roskazna.ru, S=77 Москва, ИНН ЮЛ=7710568760, ОГРН=1047797019830, STREET="Большой Златоустинский переулок, д. 6, строение 1", L=г. Москва, C=RU, O=Казначейство России, CN=Федеральное казначейство

ООД.08 ИНФОРМАТИКА

Методические указания для выполнения практических занятий
по специальности

35.02.10 Обработка водных биоресурсов

2 часть

МО – 35 02 10-ООД.08.ПЗ

РАЗРАБОТЧИКИ	Кривонос Е.В., Сукорская А.О., Халина Е.Н.
ЗАВЕДУЮЩИЙ ОТДЕЛЕНИЕМ	Судьбина Н.А.

ГОД РАЗРАБОТКИ	2023
ГОД ОБНОВЛЕНИЯ	2025

Содержание

Практическое занятие № 1 Структура информации. Графы. Введение и понятия. Способы задания графов. Алгоритм построения дерева решений

Цель занятия: познакомить обучающихся с основными понятиями теории графов.

Исходные данные: теоретический материал

Содержание и порядок выполнения:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

Понятие графа

Граф – это множество точек или вершин и множество линий или ребер, соединяющих между собой все или часть этих точек.

Вершины, прилегающие к одному и тому же ребру, называются смежными.

Два ребра, у которых есть общая вершина, также называются смежными (или соседними).

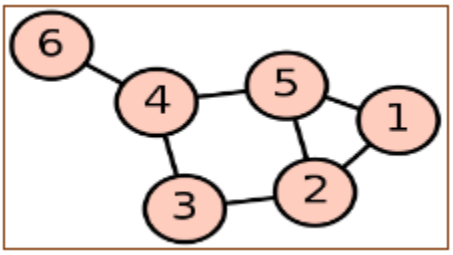


Рис. 1. Граф с шестью вершинами и семью ребрами

Элементы графа

Петля – это дуга, начальная и конечная вершина которой совпадают.

Пустым (нулевым) называется граф без ребер.

Полным называется граф, в котором каждые две вершины смежные.

Нулевой граф – граф, состоящий из «изолированных» вершин.

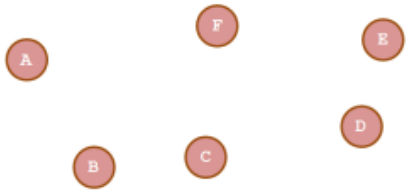


Рис. 2. Нулевой граф

Неполный граф – графы, в которых не построены все возможные ребра.

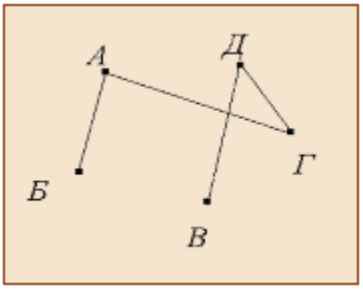


Рис. 3 Неполный граф

Степень графа

Количество рёбер, выходящих из вершины графа, называется степенью вершины. Вершина графа, имеющая нечётную степень, называется нечетной, а чётную степень – чётной.

Если степени всех вершин графа равны, то граф называется однородным. Таким образом, любой полный граф — однородный.

Кружки называются вершинами графа, линии со стрелками - дугами, без стрелок - ребрами. Граф, в котором направление линий не выделяется (все линии являются ребрами),

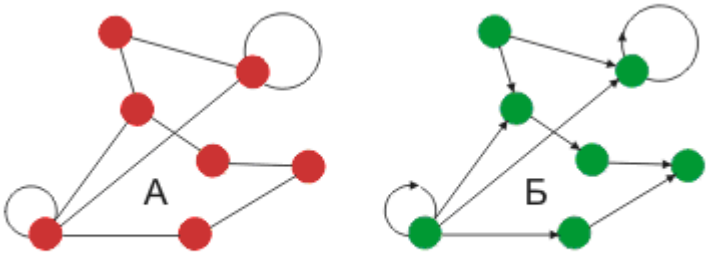


рис.1

называется неориентированным (рис. 1, А); граф, в котором направление линий принципиально (линии являются дугами) называется ориентированным (рис. 1, Б).

Ориентированный граф

Граф называется ориентированным (или орграфом), если некоторые ребра имеют направление. Это означает, что в орграфе некоторая вершина может быть соединена с другой вершиной, а обратного соединения нет. Если ребра ориентированы, что обычно показывают стрелками, то они называются дугами.

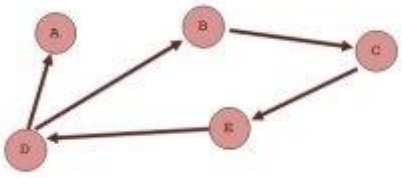


Рис. 4. Ориентированный граф

Ориентированный и неориентированный графы

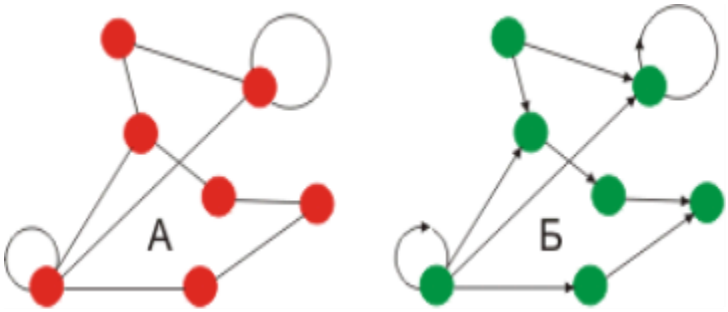


Рис. 5. Примеры неориентированного и ориентированного графов (А и Б)

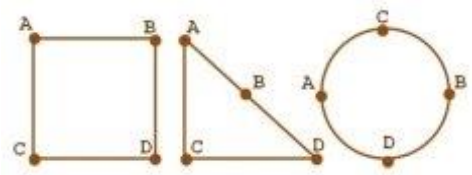
Задание 1. Построить граф по заданному условию:

В соревнованиях по футболу участвуют 6 команд. Каждую из команд обозначили буквами A, B, C, D, E и F. Через несколько недель некоторые из команд уже сыграли друг с другом:

A с C,	D,	F;
B с C,	E,	F;
C с A,		B;
D с A,	E,	F;
E с B,	D,	F;
F с A, B, D.		

Задание 2.

Определить изображают ли фигуры на рисунке один и тот же граф или нет.



1. 2) 3)

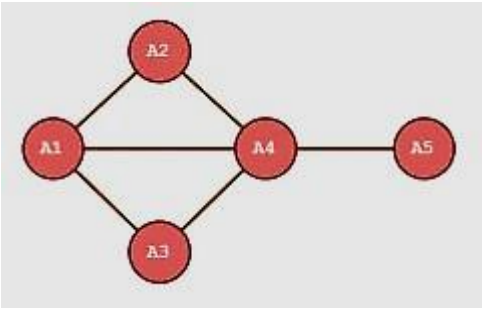
Путь в графе

Путём в графе называется такая последовательность ребер, в которой каждые два соседних ребра имеют общую вершину и никакое ребро не встречается более одного раза.

Задание 3. Определить какая из перечисленных последовательностей путём не является.

1. (A1 A4); (A4 A5).
2. (A1 A2); (A2 A4); (A4 A5).
3. (A1 A4); (A4 A2); (A2 A1); (A1 A4); (A4, A5).

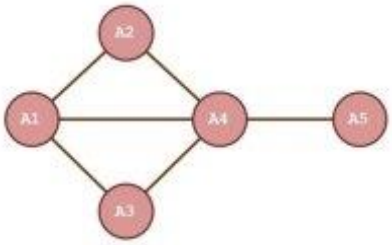
4. (A1 A4); (A4 A2); (A2 A1); (A1 A3); (A3 A4); (A4, A5).



Путь называется простым, если он не проходит ни через одну из вершин графа более одного раза.

Задание 4. Определите, какие последовательности ребер являются путями, и какие из них простые. Если последовательность не является путем укажите почему.

1. (A1 A4); (A4 A5).
2. (A1 A2); (A2 A4); (A4 A5).
3. (A1 A4); (A4 A2); (A2 A1); (A1 A4); (A4, A5).
4. (A1 A4); (A4 A2); (A2 A1); (A1 A3); (A3 A4); (A4, A5).



Понятие цикла в графе

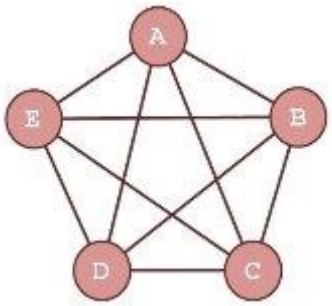
Циклом называется путь, в котором совпадают его начальная и конечная вершины.

Простым циклом в графе называется цикл, не проходящий ни через одну из вершин графа более одного раза.

Задание 5. Назовите в графе циклы, содержащие

a)	4	ребра;
b)	6	ребер;
c)	5	ребер;
d)	10	ребер.

Какие из этих циклов являются простыми?



Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Список используемых источников
4. Выводы и предложения
5. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

Изобразите графически:

1. Неориентированное и ориентированное ребро;

2. Неориентированный граф G(V,E), заданный множеством

V={v0, v1, v2, v3, v4, v5}

E = {e1=(v0, v1), e2=(v0, v2}, e3=(v1, v2), e4=(v1,v4), e5=(v2, v5), e6=(v3, v4)}

3. Плоский граф;

4. Полный неориентированный граф на трех, четырех и пяти вершинах;

5. Неполный ориентированный граф на пяти вершинах;

6. Петлю графа.

7. Цикл графа.

Практическое занятие №2 Решение логических задач с помощью графов. Анализ алгоритмов в профессиональной области

Цель работы: задание графа, вычисление степеней вершин.

Исходные данные: теоретический материал

Содержание и порядок выполнения задания:

Изучите теоретический материал

Выполните задания

Теоретическая часть

1 Граф G - совокупность двух множеств: вершин V и ребер E, между которыми определено отношение инцидентности. Если |V(G)|=n, |E(G)|=m, то граф G есть (n,m) граф, где n - порядок графа, m - размер графа.

2 Каждое ребро e из E инцидентно ровно двум вершинам v', v'', которые оно соединяет. При этом вершина v' и ребро e называются инцидентными друг другу, а вершины v' и v'' называются смежными.

3 Ребро (v',v'') может быть ориентированным и иметь начало (v') и конец (v'') (дуга в орграфе).

4 Ребро (v,v) называется петлей (концевые вершины совпадают).

5 Граф, содержащий ориентированные ребра (дуги), называется орграфом.

6 Граф, не содержащий ориентированные ребра (дуги), называется неографом.

7 Ребра, инцидентные одной паре вершин, называются параллельными или кратными.

8 Конечный граф - число вершин и ребер конечно.

9 Пустой граф - множество ребер пусто (число вершин может быть произвольным).

10 Полный граф - граф без петель и кратных ребер, каждая пара вершин соединена ребром.

11 Локальная степень вершины - число ребер ей инцидентных.

12Внеографе сумма степеней всех вершин равна удвоенному числу ребер (лемма о рукопожатиях). Петля дает вклад, равный 2 в степень вершины.

13В орграфе сумма входящих ребер всех вершин равна сумме исходящих ребер всех вершин и равна числу ребер графа.

14 Графы равны, если множества вершин и инцидентных им ребер совпадают.

15 Графы, отличающиеся только нумерацией вершин и ребер, называются изоморфными.

16 Способы задания графов: − явное задание графа как алгебраической системы; − геометрический; − матрица смежности; − матрица инцидентности

17 Матрица инцидентности: По вертикали указываются вершины, по горизонтали - ребра. aij=1 если вершина i инцидентна ребру j, в противном случае aij=0. Если ребро - петля, то aij=2. Матрицей инцидентности (инциденций) ориентированного графа называется матрица, для которой aij=1, если вершина является началом дуги ,aij= – 1, если является концом дуги, в остальных случаях aij=0.

18 Матрица смежности - квадратная симметричная матрица. По горизонтали и вертикали - все вершины.aij= число ребер, соединяющее вершины i,j. Матрицей смежности ориентированного графа называется матрица, для которой aij=1, если вершина является началом дуги, в остальных случаях aij=0.

19 Маршрут - последовательность ребер, в которых каждые два соседних ребра имеют общую вершину.

20 Маршрут, в котором начало и конец совпадают - циклический.

21 Маршрут в неографе, в котором все ребра разные - цепь.

22 Маршрут в орграфе, в котором все дуги разные - путь.

23 Вершины связанные, если существует маршрут из одной вершины в другую.

24 Связанный граф - если все его вершины связаны.

25 Плоский граф - граф с вершинами, расположенными на плоскости и непересекающимися ребрами.

26 Вершины графа, которые не принадлежат ни одному ребру, называются изолированными.

27 Дерево - связный граф без циклов.

Пример выполнения:

Исходные данные:

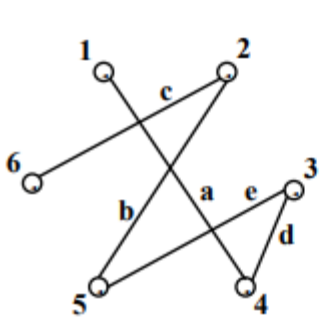
1 Задать неограф, представленный множеством вершин и ребер, графически и матрицами, преобразовать граф в плоский, вычислить степени его вершин.

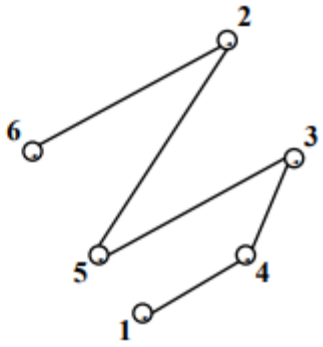
V = {1; 2; 3; 4; 5; 6};E = {a; b; c; d; e}

E = {(1; 4); (2; 5); (2; 6); (3; 4); (3; 5)}

Решение:

1Изобразим граф, соединив вершины: Ребро a соединяет вершины 1 и 4, b соединяет вершины 2 и 5 и т. д. Затем преобразуем этот граф в плоский:





2 Составим матрицу смежности. В первом столбце и первой строке выпишем вершины. Ребру a инцидентны вершины 1 и 4, следовательно в колонке 1 и строке 4 ставим 1, а также колонке 4 и строке 1 ставим 1. Ребру b инцидентны вершины 2 и 5, следовательно в колонке 2 в строке 5 и колонке 5 строке 2 ставим 1 и т.д. Остальные ячейки таблицы содержат нули.

3 Составим матрицу инцидентности. В первом столбце выпишем вершины, первой строке – ребра. Ребру a инцидентны вершины 1 и 4, следовательно в колонке a в строке 1 и строке 4 ставим 1. Ребру b инцидентны вершины 2 и 5, следовательно в колонке b в строке 2 и строке 5 ставим 1и т.д. Остальные ячейки таблицы заполняем нулями

Матрица смежности

	1	2	3	4	5	6
1	0	0	0	1	0	0
2	0	0	0	0	1	1
3	0	0	0	1	1	0
4	1	0	1	0	0	0
5	0	1	1	0	0	0
6	0	1	0	0	0	0

Матрица инцидентности

	a	b	c	d	e
1	1	0	0	0	0
2	0	1	1	0	0
3	0	0	0	1	1
4	1	0	0	1	0
5	0	1	0	0	1
6	0	0	1	0	0

4 Вычислим степени вершин:

ρ (1) = 1ρ (2) = 2ρ (3) = 2ρ (4) = 2ρ (5) = 2ρ (6) = 1

ρ (1) + ρ (2) + ρ (3) + ρ (4) + ρ (5) + ρ (6) = 10 = 2 * q

q = 5 (ребер 5)

Исходные данные:

2Задать граф, представленный матрицей инцидентности, алгебраически, графически и матрицей смежности, преобразовать граф в плоский, вычислить степени его вершин

	a	b	c	d	e	f
1	-1	-1	0	0	0	0
2	1	0	-1	1	0	0
3	0	0	0	-1	0	0
4	0	0	1	0	1	0
5	0	0	0	0	-1	-1
6	0	1	0	0	0	1

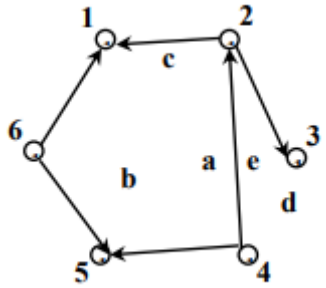
Решение:

1 Количество вершин – 6. V = {1; 2; 3; 4; 5; 6}.

2 Ребро a выходит из вершины 2, т.к. в ячейке (2; 1) стоит 1, а приходит в вершину 1 (в ячейке (1; 1) находится -1) и т.д.

Получим множество E = {(2; 1); (6; 1); (4; 2); (2; 3); (4; 5); (6; 5)}

3Изобразим граф, соединив вершины, этот граф уже плоский, т.к. ребра не пересекаются:



4 Составим матрицу смежности.

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	1	0	0	0
3	0	0	0	0	0	0
4	0	1	0	0	1	0
5	0	0	0	0	0	0
6	1	0	0	0	1	0

5 Вычислим степени вершин:

ρ1 (1) = 0

ρ1 (2) = 2

ρ1 (3) = 0

ρ1 (4) = 2

ρ1 (5) = 0

ρ1 (6) = 2

ρ2 (1) = 2

ρ2 (2) = 1

ρ2 (3) = 1

ρ2 (4) = 0

ρ2 (5) = 2

ρ2 (6) = 0

ρ1 (1) + ρ1 (2) + ρ1 (3) + ρ1 (4) + ρ1 (5) + ρ1 (6) = 6

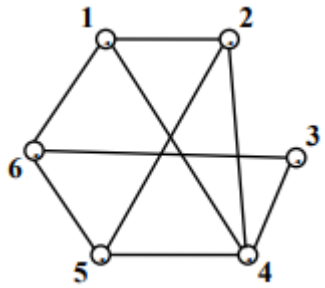
ρ2 (1) + ρ2 (2) + ρ2 (3) + ρ2 (4) + ρ2 (5) + ρ2 (6) = 6

q = 6(ребер 6)

Задания

1. Орграф V = {1; 2; 3; 4; 5; 6}

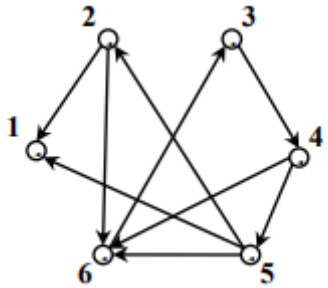
E = {(1; 3); (1; 6); (2; 5); (3; 2); (3; 4); (4; 1); (4; 5); (5; 3); (6; 2)}



2.

3 НеографV = {1; 2; 3; 4; 5; 6}

E = {(1; 2); (1; 5); (2; 3); (3; 1); (3; 4); (4; 2); (4; 5); (4; 6); (5; 3)}



4.

	a	b	c	d	e	f	g	h	i
1	1	1	0	0	0	0	0	0	0
2	1	0	1	1	0	0	0	1	0
3	0	0	0	0	1	1	0	0	0
4	0	1	0	0	0	0	0	1	1
5	0	0	0	1	1	0	1	0	0
6	0	0	1	0	0	1	1	0	1

5.

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Список используемых источников

4. Выводы и предложения
5. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что такое граф?
2. Что такое инцидентное ребро или инцидентная вершина?
3 Что такое петля?
4 Какое ребро называется ориентированным?
5 Что такое кратные ребра?
6 Что такое неограф?
7 Что такое орграф?

Практическое занятие №3 Технология подготовки и решения задач с помощью компьютера

Цель занятия:

- обобщить и систематизировать знания о графах, их видах и свойствах;
- рассмотреть типы задач, которые можно решить с использованием графов;
- отработать навыки преобразования ориентированного графа в дерево;
- сформировать навыки построение путей в графе, поиска кратчайшего пути

Содержание и порядок выполнения задания:

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

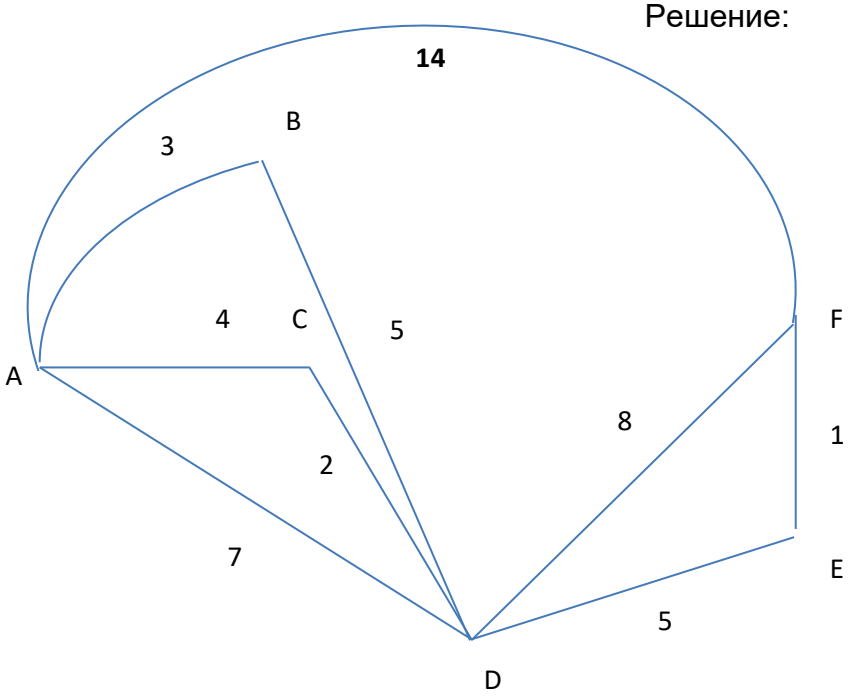
При решении логических задач по информатике часто приходится сталкиваться с заданиями, которые вызывает затруднение при решении. Если же построить граф, то задача существенно упрощается. Теория графов позволяет решать наиболее легким способом, наглядно многие логические задачи, которые способствуют развитию мышления и интеллекта. Слово «граф» означает картинку, где нарисовано несколько точек, некоторые из которых соединены линиями. Графами являются блок – схемы программ для ЭВМ, сетевые графики строительства, где вершины – события, означающие окончания работ на некотором участке, а ребра, связывающие эти вершины, - работы, которые возможно начать по совершении одного события и необходимо выполнить для совершения следующего.

Задача 1. Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице. Отсутствие числа в таблице означает, что прямой дороги между пунктами нет.

	A	B	C	D	E	F
A		3	4	7		14
B	3			5		
C	4			2		
D	7	5	2		5	8
E				5		1
F	14			8	1	

Определите длину кратчайшего пути между пунктами A и F при условии, что передвигаться можно только по указанным в таблице дорогам.

Решение:



AF=14

AF=AB+BD+DF=3+5+8=16

AF=AC+CD+DF=4+2+8=14

AF=AD+DE+EF=7+5+1=13

AF= AC+ CD+ DE+ EF=4+2+5+1=12

Ответ: 12

Задача 2

Сколькими способами можно рассадить в ряд на три стула трёх студентов? Выписать все возможные случаи.

Решение этой задачи удобнее всего представить в виде дерева. За его корневую вершину возьмём произвольную точку плоскости O.

На первый стул можно посадить любого из трёх студентов — обозначим их A, B и C. На схеме это соответствует трём ветвям, исходящим из точки O (рис. 1).

Рис. 1

Посадив на первый стул студента A, на второй стул можно посадить студента B или C. Если же на первый стул сядет B, то на второй можно посадить A или C. Если на

первый стул сядет С, то на второй можно будет посадить А или В. Это соответствует на схеме двум ветвям, исходящим из каждой вершины первого уровня (рис. 2).

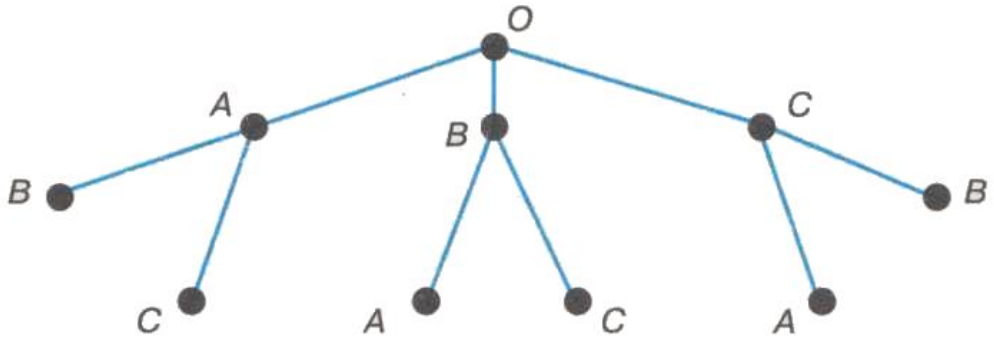


Рис. 2

Очевидно, что третий стул в каждом случае займёт оставшийся студент. Это соответствует одной ветви дерева, которая «вырастает» на каждой из предыдущих ветвей (рис. 3).

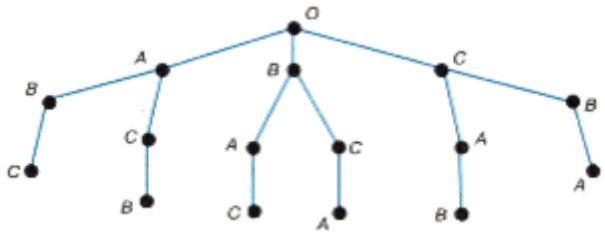


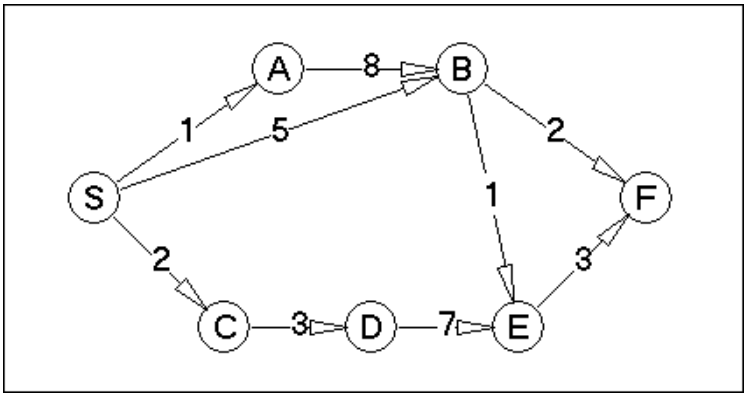
Рис. 3

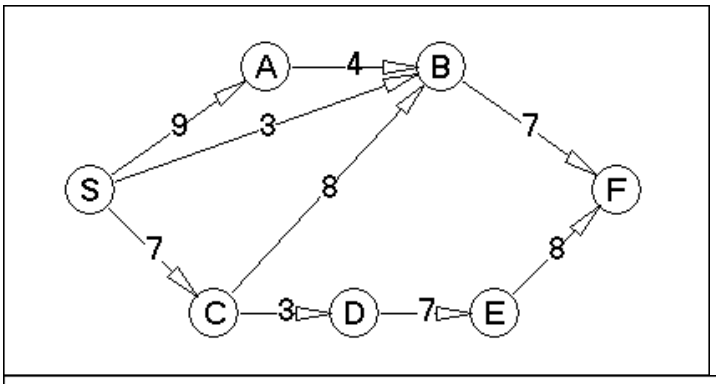
Выпишем все пути от вершин первого уровня к вершинам третьего уровня: А-В-Су А-С-В, В-А-С, В-С-А, С-А-Б, С-В-А. Каждый из выписанных путей определяет один из вариантов рассаживания учеников на стулья. Так как других путей нет, то искомое число способов — 6.

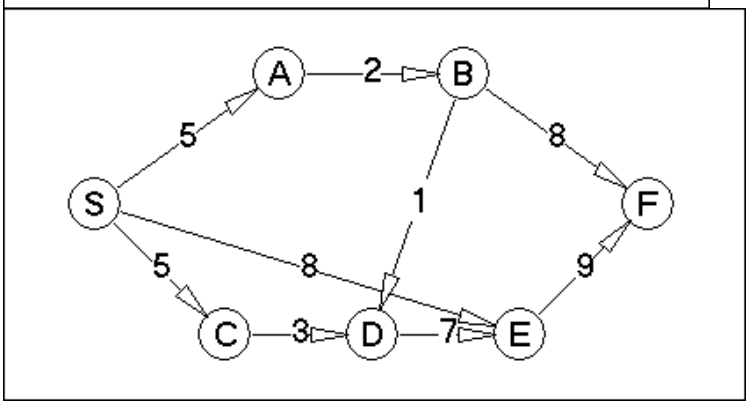
Дерево можно не строить, если не требуется выписывать все возможные варианты, а нужно просто указать их число. В этом случае рассуждать нужно так: на первый стул можно усадить одного из трёх человек, на второй — одного из двух оставшихся, на третий — одного оставшегося: 3-2-1 = 6.

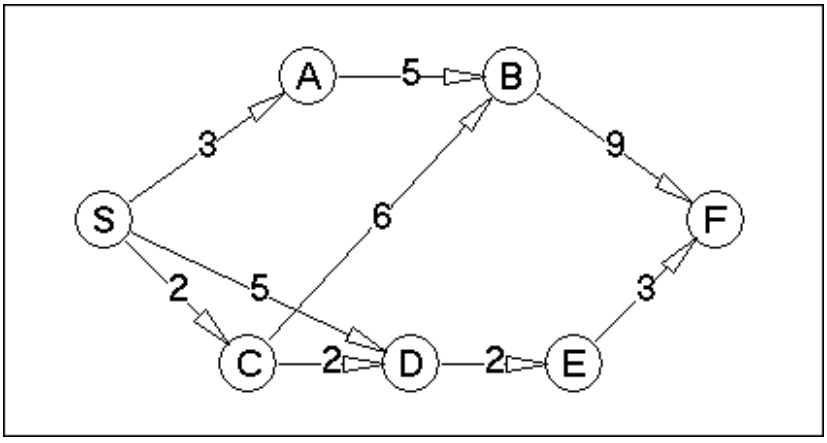
Задание1

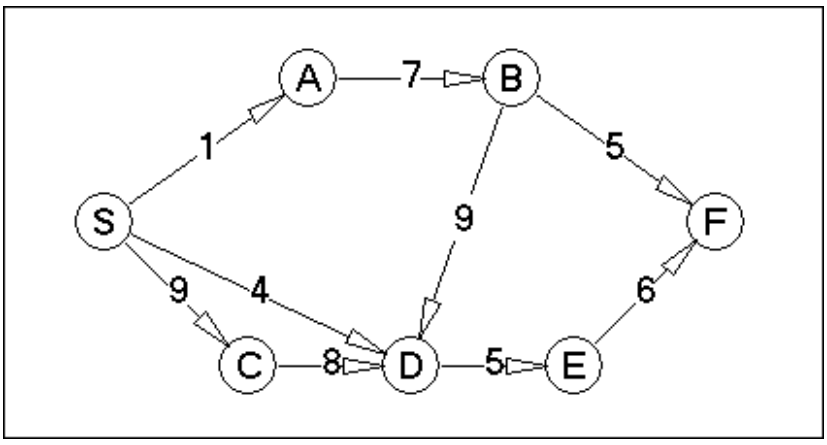
Для данного графа найти наикратчайшее расстояние











Задание 2

Между населёнными пунктами A, B, C, D, E, F построены дороги, протяжённость которых приведена в таблице:

	A	B	C	D	E	F
A		6	4	2	1	
B	6		1			
C	4	1		3		2
D	2		3		2	
E	1			2		6
F			2		6	

Определите длину кратчайшего пути между пунктами A и F. Передвигаться можно только по дорогам, протяжённость которых указана в таблице.

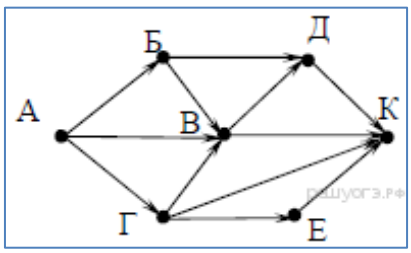
Задание 3

В таблице приведена стоимость перевозок между пятью железнодорожными станциями, обозначенными буквами A, B, C, D и E. Построить схему, соответствующую таблице.

	A	B	C	D	E
A		1	4		1
B	1			3	
C	4				2
D		3			
E	1		2		

Задание 4

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, К. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных путей из города А в город К?



Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения

6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Скажите, что такое граф?
2. Что является вершинами и ребрами графа?
3. Какой граф можно назвать взвешенным?
4. Какой граф можно назвать сематической сетью?

Практическое занятие № 4 Введение в язык программирования Python. Ввод и вывод данных. Типы данных

Цель занятия:

1. Познакомиться со средой разработки Python.

2. Изучить основные типы данных, команды ввода и вывода данных.

Исходные материалы: теоретический материал, компьютер, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

Python– это объектно-ориентированный, интерпретируемый, переносимый язык сверхвысокого уровня. Программирование на Python позволяет получать быстро и качественно необходимые программные модули.

Python (в русском языке распространено название питóн) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем.

Python — активно развивающийся язык программирования, новые версии с добавлением и изменением языковых свойств выходят примерно раз в два с половиной года. Он находит применение во множестве сфер человеческой деятельности.

Python – не самый «молодой» язык программирования, но и не слишком старый. К моменту его создания уже существовали такие языки как «Паскаль» или «Си». А потому при создании «питона» авторы старались взять лучшее из различных платформ для разработчиков. Фактически Python представляет собой своеобразный «джем» удачных решений более чем из 8 различных языков.

В комплекте вместе с интерпретатором Python идет IDLE (интегрированная среда разработки). По своей сути она подобна интерпретатору, запущенному в интерактивном режиме с расширенным набором возможностей (подсветка синтаксиса, просмотр объектов, отладка и т.п.).

Для запуска IDLE в Windows необходимо перейти в папку Python в меню “Пуск” и найти там ярлык с именем “IDLE (Python 3.X XX-bit)”.

После установки программы запустите интерактивную графическую среду IDLE и дождитесь появления приглашения для ввода команд:

Type "copyright", "credits" or "license ()" for more information.

>>>

Три знака «больше» (>>>) называются *приглашением*. После приглашения можно вводить различные команды.

В самом начале обучения Python можно представить как обычный интерактивный калькулятор. В интерактивном режиме IDLE найдем значения следующих математических выражений. После завершения набора выражения нажмите клавишуEnter для завершения ввода и вывода результата на экран.

```
>>>3.0+6
9.0
>>>4+9
13
>>>1-5
- 4
>>> _+6
2
>>>
```

Приглашение возникнет снова. Это значит, что оболочка Python готова к выполнению дальнейших команд.

Нижним подчеркиванием в предыдущем примере обозначается последний полученный результат.

Любая Python-программа состоит из последовательности допустимых символов, записанных в определенном порядке и по определенным правилам.

Программа включает в себя:

комментарии;

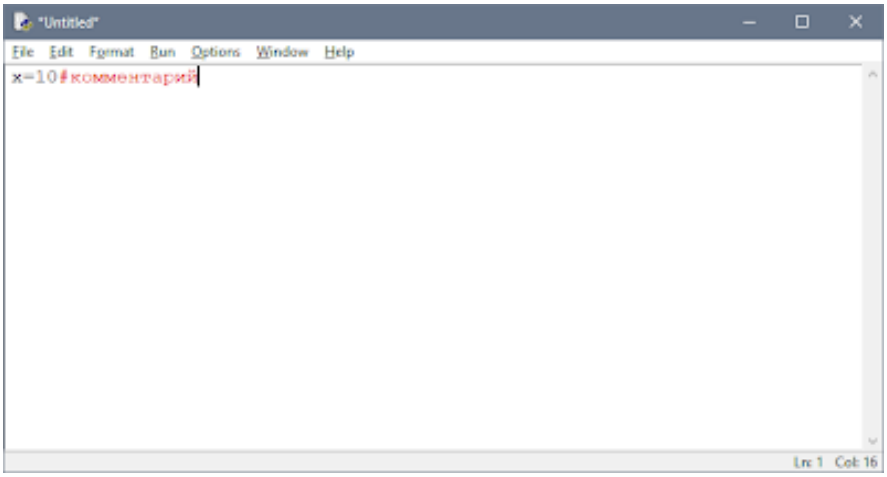
команды;

знаки пунктуации;

идентификаторы;

ключевые слова.

Комментарии в Python обозначаются предваряющим их символом # и продолжаются до конца строки (т.е. в Python все комментарии являются однострочными), при этом не допускается использование перед символом # кавычек:



Знаки пунктуации

В алфавит Python входит достаточное количество знаков пунктуации, которые используются для различных целей. Например, знаки "+" или "*" могут использоваться для сложения и умножения, а знак запятой "," - для разделения параметров функций.

Идентификаторы

Идентификаторы в Python - это имена используемые для обозначения переменной, функции, класса, модуля или другого объекта.

Ключевые слова

Некоторые слова имеют в Python специальное назначение и представляют собой управляющие конструкции языка.

Ключевые слова в Python:

['False', 'None', 'True', 'and', 'as', 'assert', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

Типы данных

None (неопределенное значение переменной)

Логические переменные (Boolean Type)

Числа (Numeric Type)

int – целое число

float – число с плавающей точкой

complex – комплексное число

Списки (Sequence Type)

list – список

tuple – кортеж

range – диапазон

Строки (TextSequence *Type*)

str

Ввод и вывод данных

Ввод данных осуществляется при помощи команды input(список ввода):

```
a = input()
print(a)
```

В скобках функции можно указать сообщение - комментарий к вводимым данным:

```
a = input ("Введите количество: ")
```

Команда input() по умолчанию воспринимает входные данные как строку символов. Поэтому, чтобы ввести целочисленное значение, следует указать тип данных int():

```
a = int (input())
```

Для ввода вещественных чисел применяется команда

```
a=float(input())
```

Вывод данных осуществляется при помощи команды print(список вывода):

```
a = 1
b = 2
print(a)
print(a + b)
print("сумма = ", a + b)
```

Существует возможность записи команд в одну строку, разделяя их через ;. Однако не следует часто использовать такой способ, это снижает удобочитаемость:

```
a = 1; b = 2; print(a)
print (a + b)
print ("сумма = ", a + b)
```

Для команды print может задаваться так называемый сепаратор — разделитель между элементами вывода:

```
x=2
y=5
print ( x, "+", y, "=", x+y, sep = " " )
```

Результат отобразится с пробелами между элементами: 2 + 5 = 7

Простые арифметические операции над числами

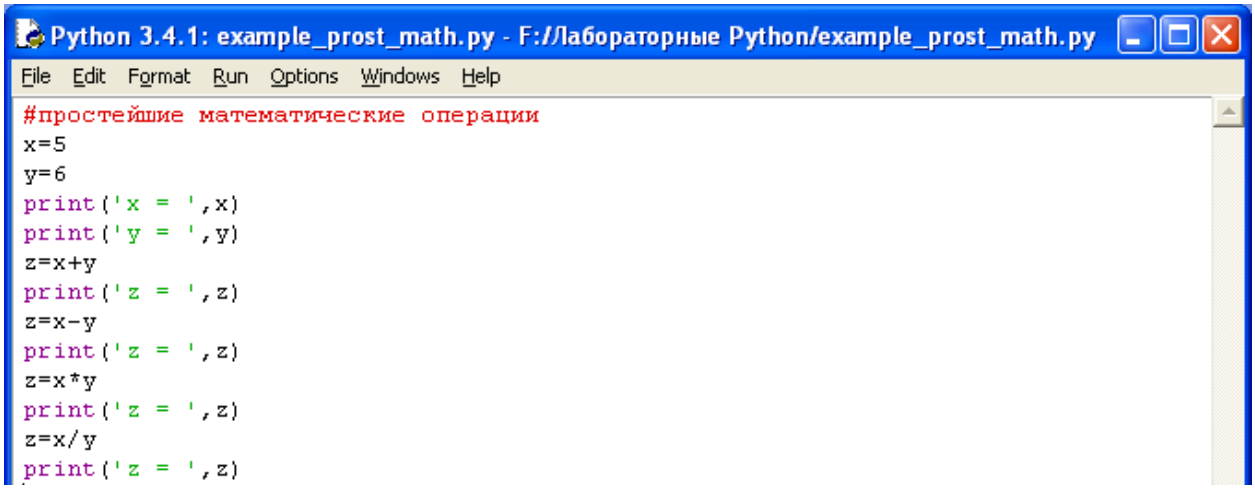
x + y	Сложение
x - y	Вычитание
x * y	Умножение

x / y	Деление
//	Деление с округлением вниз
**	Возведение в степень
%	Остаток от деления
abs(x)	Модуль числа

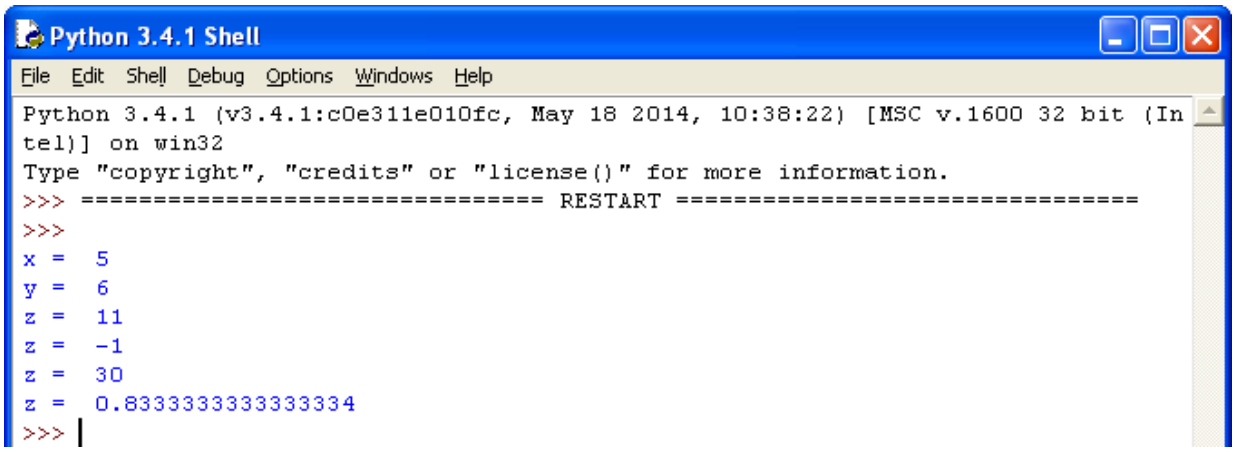
```
>>>5/3
1.6666666666666667
>>> 5//3
1
>>>5%3
2
>>>5**2
25
>>>
```

Обратите внимание, что при записи числа 1.66666666666666667 используется не запятая, а точка.

Если один из операторов является вещественным числом, то в результате получится вещественное число



Пример программы на Python



Результат выполнения программы с применением простых арифметических операций

Порядок выполнения операций

Операции — это любые действия, которые совершаются с помощью операторов. Математические операции выполняются по очереди в зависимости от их приоритета (если не задать другую очередность с помощью скобок).

Умножение и деление имеют более высокий приоритет, чем сложение и вычитание, и это значит, что они будут выполняться первыми. Иначе говоря, при вычислении математического выражения Python сначала умножит и разделит числа, а затем перейдет к сложению и вычитанию.

Например, 5 + 30 * 20

в этом выражении сперва будут перемножены числа 30 и 20, а затем к их произведению будет прибавлено число 5.

```
>>> 5 + 30 * 20
605
```

По сути, это выражение означает «умножить 30 на 20 и прибавить к результату 5». Получается 605. Однако мы можем изменить порядок операций, заключив первые два числа в скобки. Вот так:

```
>>> (5 + 30) * 20
700
```

В результате получилось 700, а не 605, поскольку Python выполняет операции в скобках прежде, чем операции вне скобок. Другими словами, это выражение означает «прибавить 5 к 30 и умножить результат на 20».

Скобки могут быть *вложенными*, то есть внутри скобок могут стоять еще одни скобки:

```
>>> ((5 + 30) * 20) / 10
70.0
```

В этом примере Python сперва вычислит выражение во внутренних скобках, затем во внешних и в самом конце выполнит стоящую за скобками операцию деления.

Иначе говоря, это выражение означает «прибавить 5 к 30, затем умножить результат на 20, потом разделить результат на 10». Вот что при этом происходит:

сложение 5 и 30 дает 35;

умножение 35 на 20 дает 700;

деление 700 на 10 дает окончательный результат — 70.

Если бы мы не использовали скобки, результат вышел бы другим:

```
>>> 5 + 30 * 20 / 10
65.0
```

В этом случае сперва 30 умножается на 20 (получается 600), затем 600 делится на 10 (выходит 60) и, наконец, к 60 прибавляется 5, что дает в итоге 65.

! *Запомните, что умножение и деление всегда выполняются прежде, чем сложение и вычитание, если не менять порядок вычислений с помощью скобок.*

Для форматированного вывода используется format:

Строковый метод format() возвращает отформатированную версию строки, заменяя идентификаторы в фигурных скобках {}. Идентификаторы могут быть позиционными, числовыми индексами, ключами словарей, именами переменных.

Синтаксис команды format:

поле замены := "{" [имя поля] ["!" преобразование] [":" спецификация] "}"

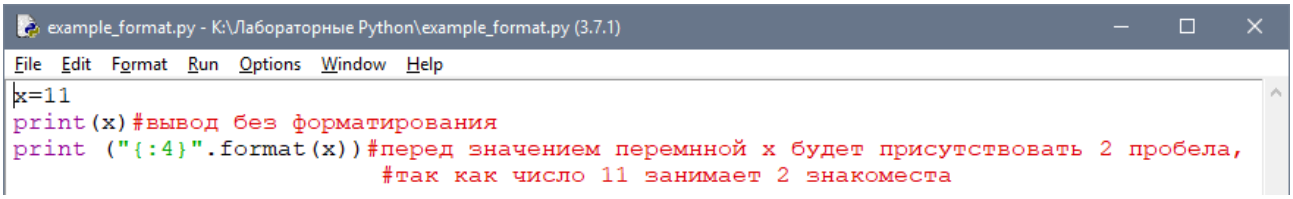
имя поля := arg_name ("." имя атрибута | "[" индекс "]")*

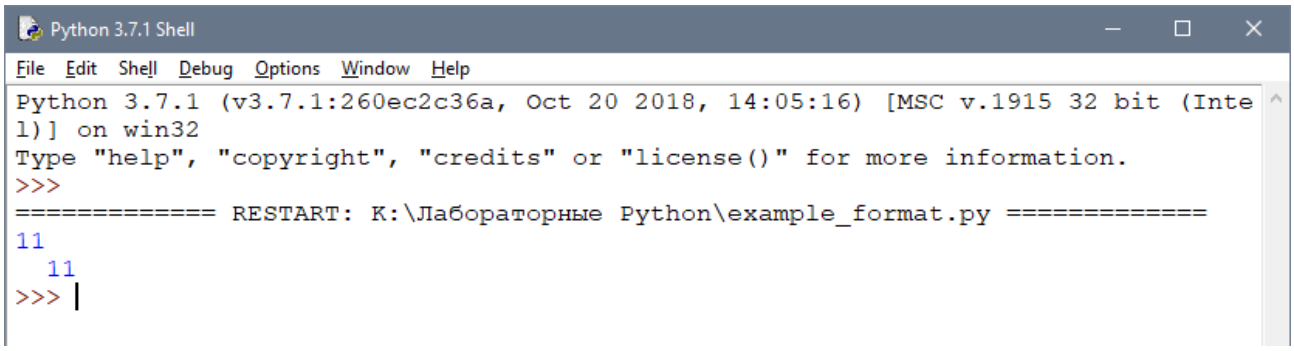
преобразование := "r" (внутреннее представление) | "s" (человеческое представление)

спецификация := см. ниже

Аргументов в format() может быть больше, чем идентификаторов в строке. В таком случае оставшиеся игнорируются.

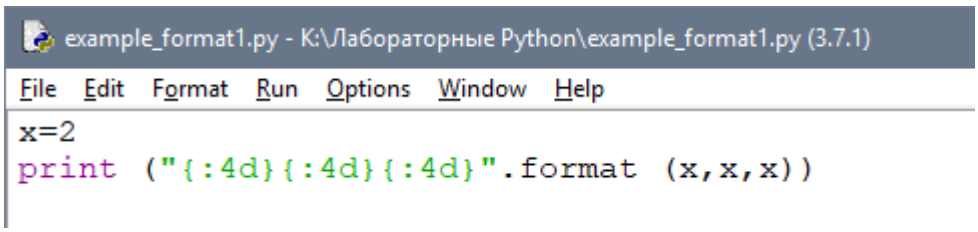
Идентификаторы могут быть либо индексами аргументов, либо ключами:

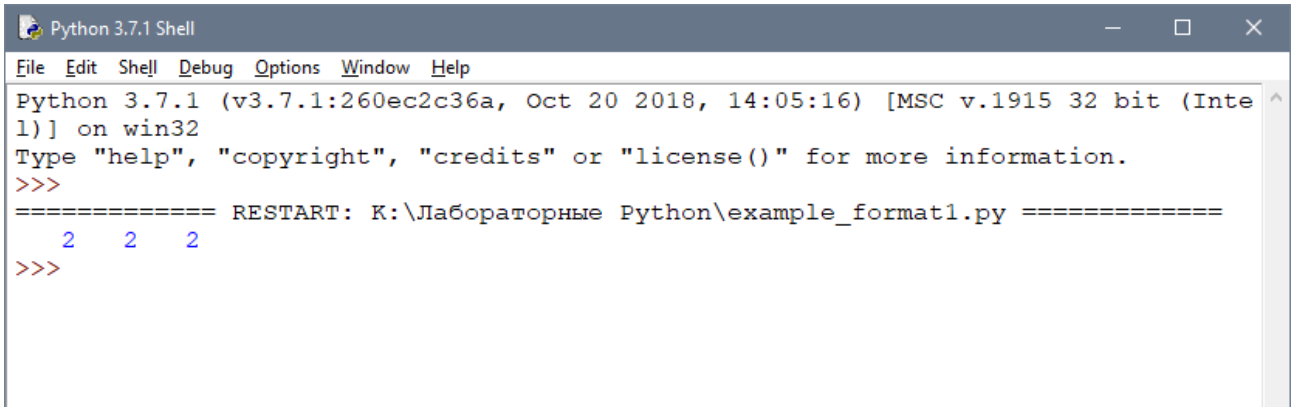




В результате выведется число 11, а перед ним два пробела, так как указано использовать для вывода четыре знакоместа.

Или с несколькими аргументами:





В итоге каждое из значений выводится из расчета 4 знакоместа.

Спецификация формата:

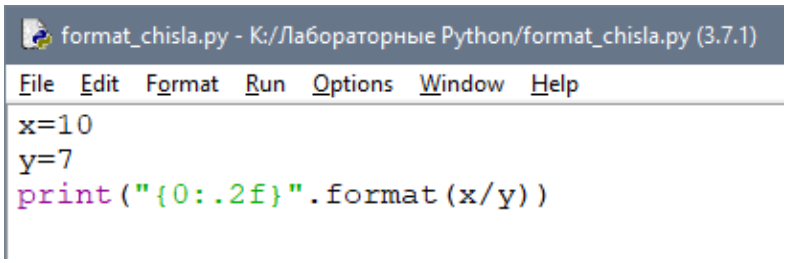
спецификация	:= [[fill]align][sign][#][0][width][,][.precision][type]
заполнитель	:= символ кроме '{' или '}'
выравнивание	:= "<" | ">" | "=" | "^"
знак	:= "+" | "-" | " "
ширина	:= integer
точность	:= integer
тип	:= "b" | "c" | "d" | "e" | "E" | "f" | "F" | "g" | "G" | "n" | "o" | "s" | "x" | "X" | "%"

Тип	Значение
'd', 'i', 'u'	Десятичное число.

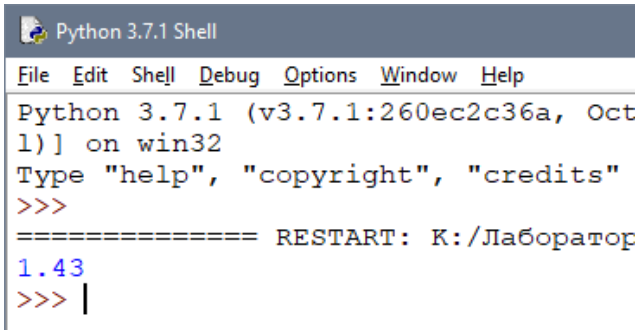
'o'	Число в восьмеричной системе счисления.
'x'	Число в шестнадцатеричной системе счисления (буквы в нижнем регистре).
'X'	Число в шестнадцатеричной системе счисления (буквы в верхнем регистре).
'e'	Число с плавающей точкой с экспонентой (экспонента в нижнем регистре).
'E'	Число с плавающей точкой с экспонентой (экспонента в верхнем регистре).
'f', 'F'	Число с плавающей точкой (обычный формат).
'g'	Число с плавающей точкой. с экспонентой (экспонента в нижнем регистре), если она меньше, чем -4 или точности, иначе обычный формат.
'G'	Число с плавающей точкой. с экспонентой (экспонента в верхнем регистре), если она меньше, чем -4 или точности, иначе обычный формат.
'c'	Символ (строка из одного символа или число - код символа).
's'	Строка.
'%'	Число умножается на 100, отображается число с плавающей точкой, а за ним знак %.

Для форматирования вещественных чисел с плавающей точкой используется следующая команда:

print('{0:.2f}'.format(вещественное число))



В результате выведется число с двумя знаками после запятой.



Пример

Напишите программу, которая запрашивала бы у пользователя:

Вариант 0

- ФИО ("Ваши фамилия, имя, отчество?")

- возраст ("Сколько Вам лет?")

- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш возраст"

"Вы живете в"

Решение

```
a=input('Введите ваши фамилию, имя, отчество ')
b=input('Сколько вам лет? ')
c=input('Где вы живёте? ')
print('Ваше имя ',a)
print('Ваш возраст ',b)
print('Вы живете в ',c)
```

```
Введите ваши фамилию, имя, отчество Иванов Иван Иванович
Сколько вам лет? 15
Где вы живёте? Уссурийск
Ваше имя  Иванов Иван Иванович
Ваш возраст  15
Вы живете в  Уссурийск
```

Задания для самостоятельной работы (по вариантам)

Напишите программу, которая запрашивала бы у пользователя:

Вариант 1

Имя, Фамилия, Возраст, Место жительства

- фамилия, имя ("Ваши фамилия, имя?")

- возраст ("Сколько Вам лет?")

- место жительства ("Где вы живете?")

После этого выводила бы три строки:

"Ваши фамилия, имя"

"Ваш возраст"

"Вы живете в"

Вариант 2

Имя, Дата рождения, Образование

- имя ("Ваше, имя?")

- дата рождения ("Ваша дата рождения?")

- образование ("Где Вы учитесь?")

После этого выводила бы три строки:

"Ваше имя"

"Дата рождения"

"Вы учитесь в "

Вариант 3

Фамилия, Место жительства

- Фамилия("Ваша фамилия?")

- место жительства ("Где Вы живете?")

После этого выводила бы две строки:

"Ваша фамилия"

"Вы живете в"

Вариант 4

Фамилия, Место рождения, любимая музыка

- Фамилия, ("Ваша фамилия?")

- место рождения ("Где Вы родились?")

- музыка("Какая музыка нравится? ")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы родились в"

"Ваша любимая музыка "

Вариант 5

Имя, Фамилия, ФИО мамы, ФИО отца

- ФИО (например, "Ваши фамилия, имя, отчество?")

- возраст ("Сколько Вам лет?")

- место жительства ("Где Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия, отчество"

"Ваш возраст"

"Вы живете в"

Вариант 6

Имя, Любимый предмет в школе, Номер класса

- имя ("Ваше имя?")

- любимый предмет ("Какой Ваш любимый предмет в школе?")

- номер класса ("В каком классе Вы учитесь?")

После этого выводила бы три строки:

"Ваше имя"

"Ваш любимый предмет в школе"

"Вы учитесь в классе номер"

Вариант 8

Имя, Фамилия, Отчество, Хобби

- ФИО (например, "Ваши фамилия, имя, отчество?")

- хобби ("Чем Вы увлекаетесь?")

После этого выводила бы две строки:

"Ваши имя, фамилия, отчество"

"Ваше хобби"

Вариант 9

Имя, Фамилия, любимый спорт

- Фамилия, имя ("Ваши фамилия, имя?")

- образование ("В какой школе Вы учитесь?")

- ФИО Вашего руководителя по информатики ("ФИО Вашего руководителя по информатики?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы учитесь в школе номер: "

"ФИО Вашего руководителя по информатике "

Вариант 10

Имя, Фамилия, Любимый предмет в школе (в институте), ФИО классного руководителя (куратора)

- Фамилия, имя ("Ваши фамилия, имя?")

- любимый предмет в школе ("Какой Ваш любимый предмет в школе?")

- ФИО классного руководителя ("ФИО Вашего классного руководителя?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш любимый предмет в школе "

"ФИО Вашего классного руководителя"

Вариант 11

Имя, Фамилия, Возраст, Дата рождения

- Фамилия, имя ("Ваши фамилия, имя?")

- возраст ("Сколько Вам лет?")

- дата рождения ("Когда Вы родились?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш возраст"

"Дата Вашего рождения"

Вариант 12

Имя, Фамилия, Место жительства, Месторождения

- Фамилия, имя ("Ваши фамилия, имя?")

- место рождения ("Где Вы родились?")

- место жительства ("Где Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы родились в"

"Вы живете в"

Вариант 13

Имя, Фамилия, Возраст, Номер телефона

- Фамилия, имя ("Ваши фамилия, имя?")

- возраст ("Сколько тебе лет?")

- номер телефона ("Номер Вашего телефона?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Ваш возраст"

"Ваш номер телефона"

Вариант 14

Имя, Фамилия, Страна, Край, Город

- Фамилия, имя ("Ваши фамилия, имя?")

- страна ("В какой стране Вы живете?")

- город ("В каком городе Вы живете?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"Вы живете в стране"

"Вы живете в крае"

"Вы живете в городе"

Вариант 15

Имя, Фамилия, ФИО Вашего классного руководителя

- Фамилия, имя ("Ваши фамилия, имя?")

- ФИО Вашего классного руководителя ("ФИО Вашего классного руководителя?")

После этого выводила бы три строки:

"Ваши имя, фамилия"

"ФИО Вашего руководителя по информатике"

"ФИО Вашего классного руководителя"

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;

2. Цель занятия;

3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4. Список используемых источников;

5. Выводы и предложения;

6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1.Как в языке Python называются указания компьютеру, определяющие, какие операции выполнит компьютер над данными?

2.Определите порядок выполнения операций в указанной инструкции?

```
a = 3 - 5 * 4 ** (-3 + 2)
```

3. Символ # в Python обозначает …

4.Операция 3**4 - это

5.345 - … тип данных.

6.Операция 46%10 – это …

7.Функция input() – предназначена для …

Практическое занятие № 5 Оператор присваивания. Математические операции с целыми и вещественными числами.

Цель занятия:

1. Познакомиться с основными математическими операциями в Python

Исходные материалы: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

Язык Python, благодаря наличию огромного количества библиотек для решения разного рода вычислительных задач, сегодня является конкурентом таким пакетам как Matlab и Octave. Запущенный в интерактивном режиме, он, фактически, превращается в мощный калькулятор. В этом практическом занятии речь пойдет об арифметических операциях, доступных в данном языке Арифметические операции изучим применительно к числам.

Если в качестве операндов некоторого арифметического выражения используются только целые числа, то результат тоже будет целое число. Исключением является операция деления, результатом которой является вещественное число. При совместном использовании целочисленных и вещественных переменных, результат будет вещественным.

Целые числа (int)

Числа в Python 3 поддерживают набор самых обычных математических операций:

x + y	Сложение
x - y	Вычитание
x * y	Умножение
x / y	Деление
x // y	Получение целой части от деления
x % y	Остаток от деления
-x	Смена знака числа
abs(x)	Модуль числа
divmod(x, y)	Пара (x // y, x % y)
x ** y	Возведение в степень
pow(x, y[, z])	x : Число, которое требуется возвести в степень. y : Число, являющееся степенью, в которую нужно возвести первый аргумент. Если число отрицательное или одно из чисел "x" или "y" не целые, то аргумент "z" не принимается. z : Число, на которое требуется произвести деление по модулю. Если число указано, ожидается, что "x" и "y" положительны и имеют тип int.

Пример применения выше описанных операций над целыми числами

```
x = 5
y = 2
z = 3
x+y = 7
x-y = 3
x*y = 10
x/y = 2.5
x//y = 2
x%y = 1
-x= -5
abs(-x) = 5
divmod(x,y) = (2, 1)
x**y = 25
pow(x,y,z) = 1
```

Вещественные числа (float)

Вещественные числа поддерживают те же операции, что и целые. Однако (из-за представления чисел в компьютере) вещественные числа неточны, и это может привести к ошибкам.

Пример применения вышеописанных операций над вещественными числами

```
x = 5.5
y = 2.3
x+y = 7.8
x-y = 3.2
x*y = 12.649999999999999
x/y = 2.3913043478260874
x//y = 2.0
x%y = 0.9000000000000004
-x= -5.5
abs(-x) = 5.5
divmod(x,y) = (2.0, 0.9000000000000004)
x**y = 50.44686540422945
```

Библиотека (модуль) math

В стандартную поставку Python входит библиотека math, в которой содержится большое количество часто используемых математических функций.

Для работы с данным модулем его предварительно нужно импортировать.

Рассмотрим наиболее часто используемые функции модуля math

math.ceil(x)	Возвращает ближайшее целое число большее, чем x
math.fabs(x)	Возвращает абсолютное значение числа x
math.factorial(x)	Вычисляет факториал x
math.floor(x)	Возвращает ближайшее целое число меньшее, чем x
math.exp(x)	Вычисляет e**x
math.log2(x)	Логарифм по основанию 2
math.log10(x)	Логарифм по основанию 10
math.log(x[, base])	По умолчанию вычисляет логарифм по основанию e, дополнительно можно указать основание логарифма
math.pow(x, y)	Вычисляет значение x в степени y
math.sqrt(x)	Корень квадратный от x

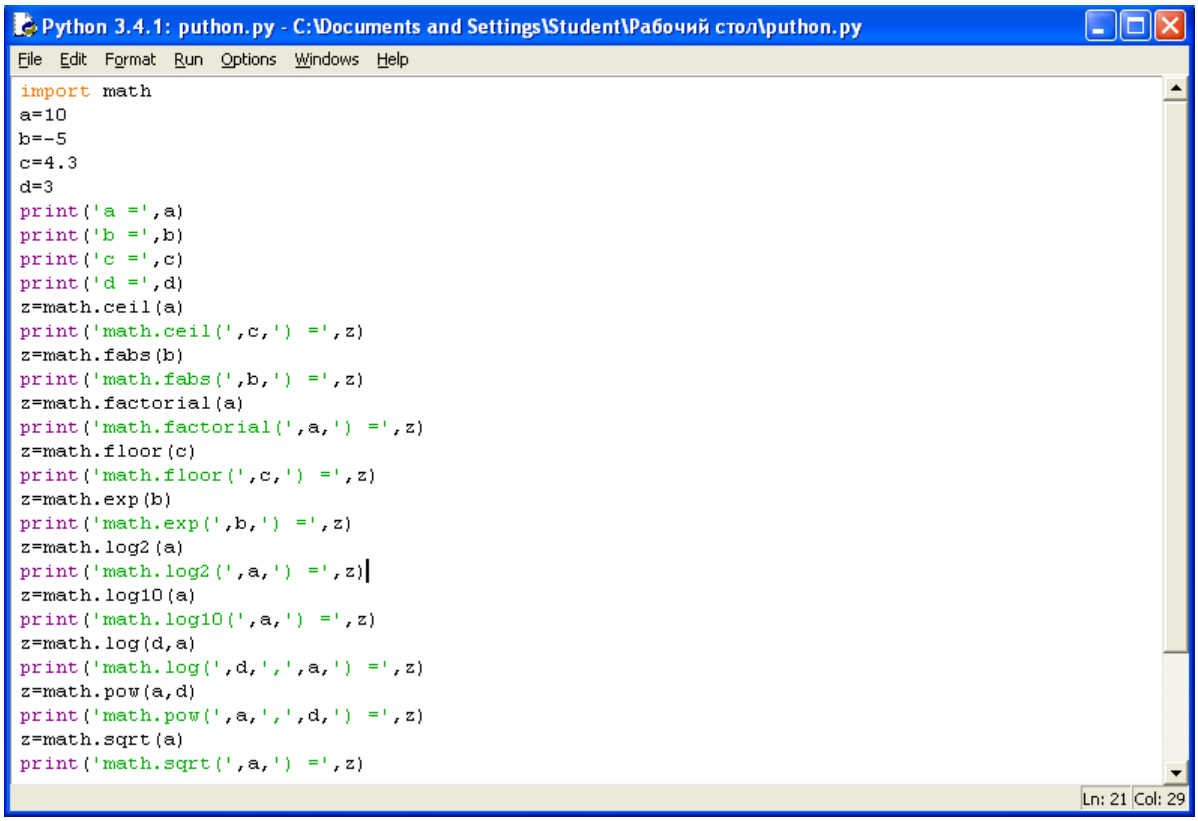
Пример применения вышеописанных функций над числами

В программе определены 4 переменные - a, b, c, d, каждая из которых является либо целым числом, либо вещественным, либо отрицательным.

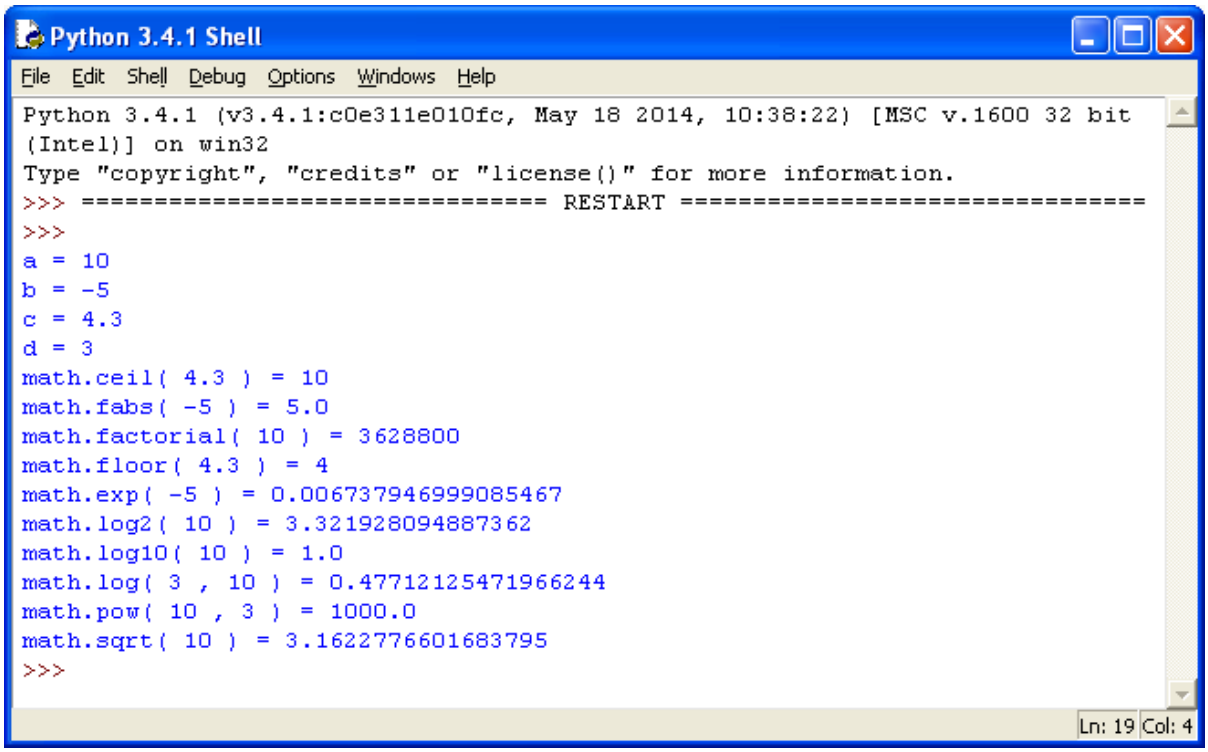
Командой print() выводится значение каждой переменной на экран при выполнении программы.

В переменную z помещается результат выполнения функции модуля math.

Затем командой print() выводится сообщение в виде используемой функции и её аргумента и результат её выполнения.



Пример программы на Python



Результат выполнения программы с применением функций модуля math

Тригонометрические функции модуля math

math.cos(x)	Возвращает cos числа X
math.sin(x)	Возвращает sin числа X
math.tan(x)	Возвращает tan числа X

math.acos(x)	Возвращает acos числа X
math.asin(x)	Возвращает asin числа X
math.atan(x)	Возвращает atan числа X

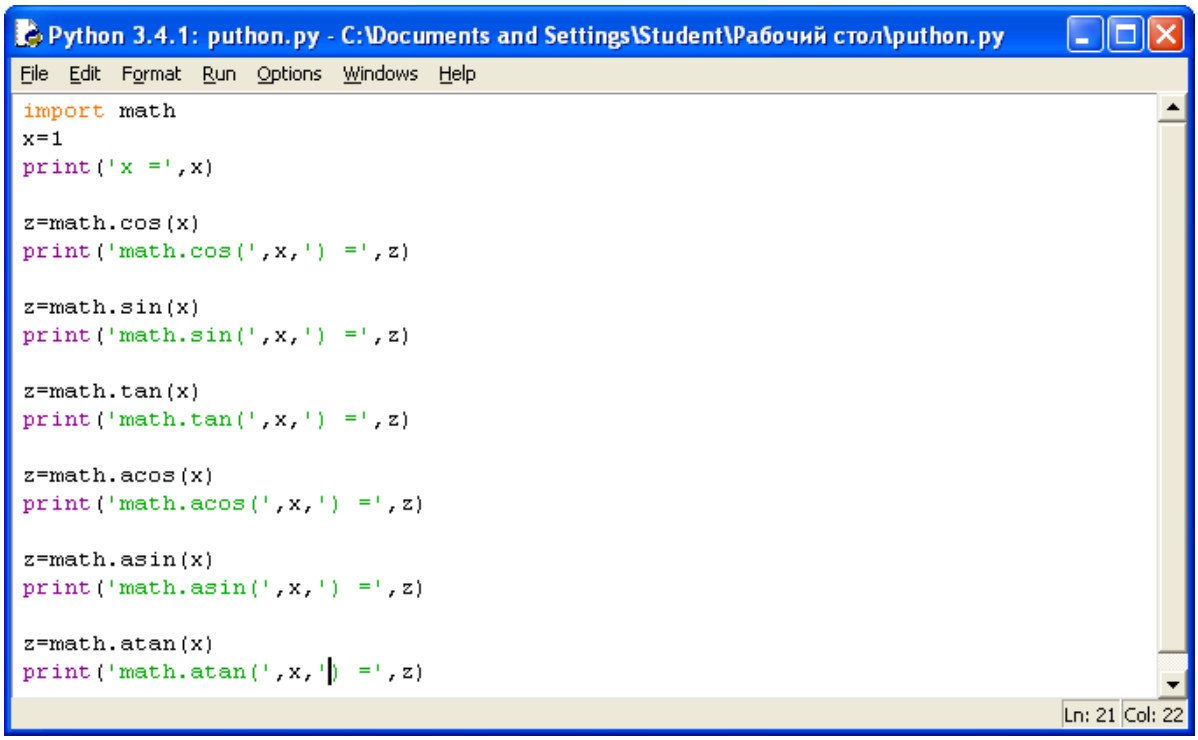
Пример применения вышеописанных функций над числами

В программе определена переменная x, содержащая целое число.

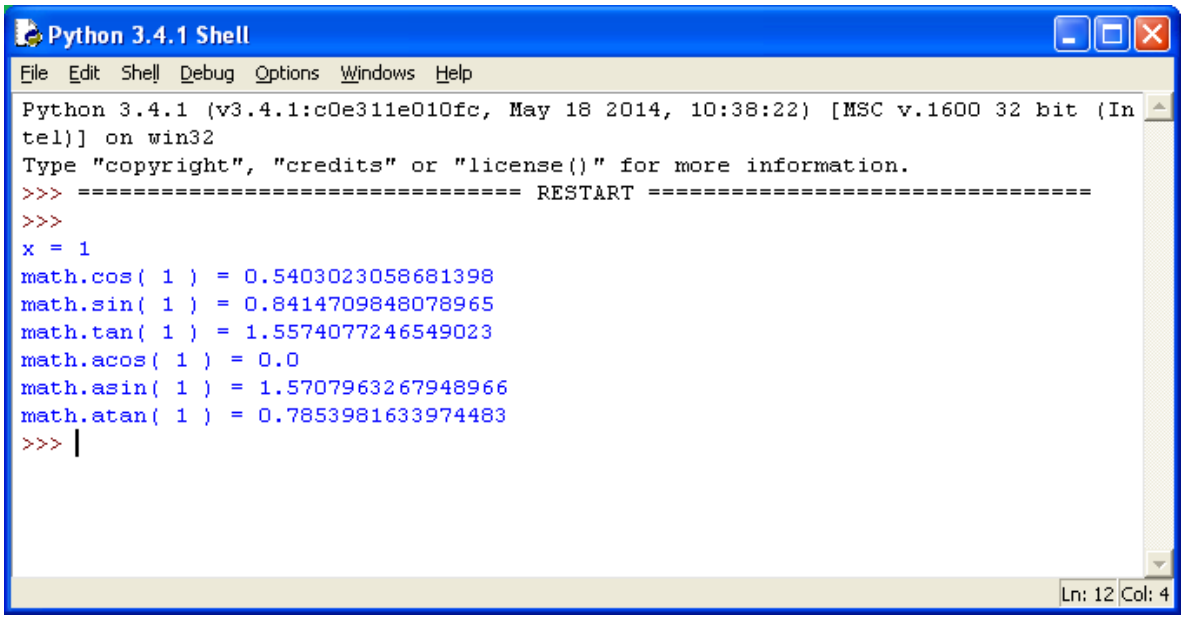
Значение переменной выводится командой print() на экран.

В переменную z помещается результат выполнения тригонометрической функции модуля math.

Затем командой print() выводится сообщение в виде используемой функции и её аргумента и результат её выполнения.



Пример программы с использованием тригонометрических функций модуля math



Результат выполнения программы с применениемтригонометрических функций модуля math

Константы:

- math.pi - число Pi.
- math.e - число e (экспонента).

Пример

Напишите программу, которая бы вычисляла заданное арифметическое выражение при заданных переменных. Ввод переменных осуществляется с клавиатуры. Вывести результат с 2-мя знаками после запятой.

Вариант 0

$$Z = \frac{9\pi t + 10\cos(x)}{\sqrt{t} - |\sin(t)|} * e^{x}$$

x=10; t=1

Решение

Сначала импортируем модуль math. Для этого воспользуемся командой importmath.

Затем следует ввести значения двух переменных целого типа x и t.

Для ввода данных используется команда input, но так как в условии даны целые числа, то нужно сначала определить тип переменных: x=int(), t=int().

Определив тип переменных, следует их ввести, для этого в скобках команды int() нужно написать команду input().

Для переменной x это выглядит так: x=int(input("сообщение при вводе значения")).

Для переменной t аналогично: t=int(input("сообщение при вводе значения")).

Следующий шаг - это составление арифметического выражения, результат которого поместим в переменную z.

Сначала составим числитель. Выглядеть он будет так: 9*math.pi*t+10*math.cos(x).

Затем нужно составить знаменатель, при этом обратим внимание на то, что числитель делится на знаменатель, поэтому и числитель, и знаменатель нужно поместить в скобки (), а между ними написать знак деления /.

Выглядеть это будет так: (9*math.pi*t+10*math.cos(x))/(math.sqrt(t)-math.fabs(math.sin(t))).

Последним шагом является умножение дроби на экспоненту в степени x.

Так как умножается вся дробь, то следует составленное выражение поместить в скобки (), а уже потом написать функцию math.pow(math.e,x).

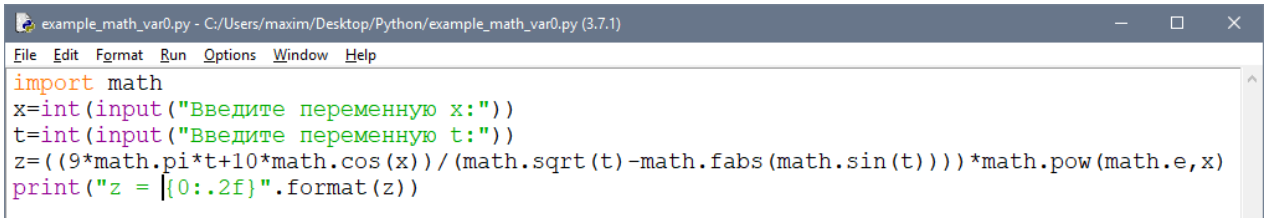
В результате выражение будет иметь вид:
z=((9*math.pi*t+10*math.cos(x))/(math.sqrt(t)-math.fabs(math.sin(t))))*math.pow(math.e,x).

При составлении данного выражения следует обратить внимание на количество открывающихся и закрывающихся скобок.

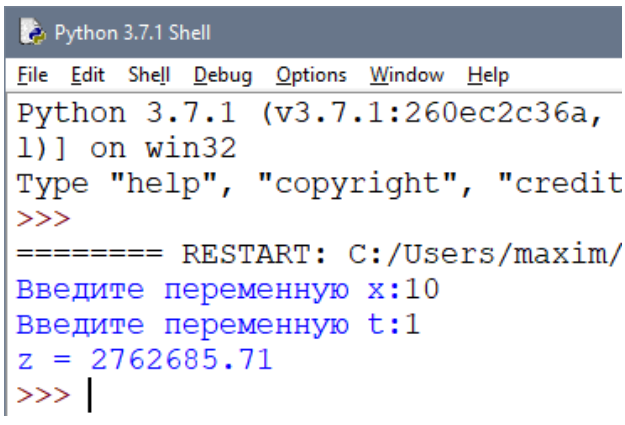
Командой print() выведем значение переменной, отформатировав его командой format.

Сам формат записывается в апострофах в фигурных скобках {}.

В задаче требуется вывести число с двумя знаками после запятой, значит вид формата будет выглядеть следующим образом: {0:.2f}, где 2 - это количество знаков после запятой, а f указывает на то, что форматируется вещественное число. При этом перед 2 нужно поставить точку, указав тем самым на то, что форматируем именно дробную часть числа.



Результат



ВАРИАНТЫ ЗАДАНИЙ

Вариант 1.

Вычислить. $a = \dfrac{\sqrt{|x-1|} - \sqrt[3]{|y|}}{1 + \dfrac{x^2}{2} + \dfrac{z^2}{4}}$, и $b = x(arctgZ + e)^{-(y+3)}$; если *x = 2; y = 6; z = 7.*

Вариант 2.

Вычислить $a = \dfrac{3 + e^{y-1}}{1 + x(y - tgZ)}$, и $b = 1 + |y - x| + \cdot\dfrac{(y-x)^2}{2} + \dfrac{|y-z|^3}{3}$;

если *x = 5; y = 1; z = 4.*

Вариант 3.

Вычислить $a = (1+y)\dfrac{x + y/(x^2+4)}{e^{-x-2} + 1/(z^2+4)}$, и $b = \dfrac{1 + \cos(y-2)}{x^{4/2} + \sin^2 z}$; если *x =1; y=12; z = 6.*

Вариант 4.

Вычислить $a = y + \dfrac{x}{z^2 + \left|\dfrac{x^2}{y + x^3/3}\right|}$, и $b = (1 + \arcsin\ x^2 + tg^2\dfrac{z}{2})$;

если *x = 11; y = 5; z = 10.*

Вариант 5.

Вычислить $a = \dfrac{2\cos(x - n/6)}{1/2 + \sin^2 y}$, и $b = 1 + \dfrac{z^2}{3 + z^2/5}$; если *x = 15; y = 7; z = 3.*

Вариант 6.

Вычислить $a = \dfrac{1 + \sin^2(x-y)}{2 + \left|x - 2x/(1 + x^2y^2)\right|} + z$, и $b = \cos^2(arctg\dfrac{1}{z})$; если

x = 3; y = 4; z = 5.

Вариант 7.

Вычислить $a = \ln\left|(y - \sqrt{|x|})(x - \dfrac{y}{z + x^2/4})\right|$, и $b = x - \dfrac{x^2}{3!} + \dfrac{x^5}{5}$;

если *x =5; y =3; z =17.*

Вариант 8.

Вычислить $a = \left|\dfrac{\sin^3\left|3x^3 + 5y^2 + 15z\right|}{\sqrt{(12x^3 + 6y^2 - z)^2 + 3.14}}\right|$, и $b = tg(7x^2 + e^{3y^2 - z})$; если

x = 7; y = 4; z = 8.

Вариант 9.

Вычислить $a = \sqrt{\frac{|\sin 8y| + 17x}{(1 - \sin 4 \cdot y \cdot \cos(z^2 + 18))^2}}$; и $b = y - \sqrt{\frac{3|z|^2}{3 + |tg3x^2 - \sin 5y|}}$; если

x = 17; y = 16; z = 15.

Вариант 10.

Вычислить $a = l^{(2x^2 - y)} + \frac{3\arccos y^2}{(z^2 + xy)^2}$, и $b = tg\left(\frac{|x - y|^2}{3Z + \cos x^2}\right)$; если

x = 6; y = 7; z = 8.

Вариант 11.

Вычислить $a = 10 \cdot \ln\left|(x - \sqrt{\cos y})(z - \frac{y^2}{x - x^2/2})\right|$; и $b = 2x^3 + \frac{|y^2 - 3z^3|}{e^{3x + y^2}}$; если *x = 3; y = 4; z = 5.*

Вариант 12.

Вычислить $a = \frac{\cos^2 x + 5\sin^3 y}{\ln|2z + z^3|}$; и $b = \arcsin\left(\frac{4x^2 + 5y^3}{\sqrt{2z - y^2}}\right) / \ln(\sqrt{7x - 7x/5y^2})$; если *x = 4; y = 5; z = 6.*

Вариант 13.

Вычислить $a = tg\left((x + y)^2 - \sqrt{\frac{\cos^2(z)}{tgz^2}}\right)$, и $b = \ln(x^2 + \frac{e^{-3y^2}}{\sin^2 y})$; если

x = 5; y = 6; z = 7.

Вариант 14.

Вычислить $a = \left(\frac{y}{\sin x} + \frac{y}{(\sin^2 x - 3\cos z)}\right) \cdot e^{5x^2}$, и $b = \frac{5 - e^{z-2}}{y + x^2|z^2 - tgz|}$;

если *x = 8; y = 9; z = 10.*

Вариант 15.

Вычислить $a = z + |y^2 - x^2| + y(\arcsin(z - e^{(z+5)}))$; и $b = \sqrt{\frac{x - \cos^2(y + z)}{5 + \ln(y - 5x/|xy - \sqrt{7}|)}}$

; если *x = 9; y = 10; z = 11*

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;
2. Цель занятия;

3. Вариант задания;

4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

5. Список используемых источников;

6. Выводы и предложения;

7. Дата и подпись курсанта и преподавателя;

Вопросы для самопроверки

1 Перечислите математические операции для целых чисел.

2 Перечислите часто используемые функции модуля math.

3 Перечислите тригонометрические функции модуля math.

Практическое занятие №6 Стандартные функции. Математический модуль math.

Цель занятия:

1. Познакомиться со структурой ветвление (if, if-else, if-elif-else);

2. Научиться работать с числами и строками используя структуру ветвление.

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

Условный оператор ветвления if, if-else, if-elif-else

Оператор ветвления if позволяет выполнить определенный набор инструкций в зависимости от некоторого условия. Возможны следующие варианты использования.

1. Конструкция if

Синтаксис оператора if выглядит так:

```
if логическое выражение:
    команда_1
    команда_2
    ...
    команда_n
```

После оператора if записывается логическое выражение.

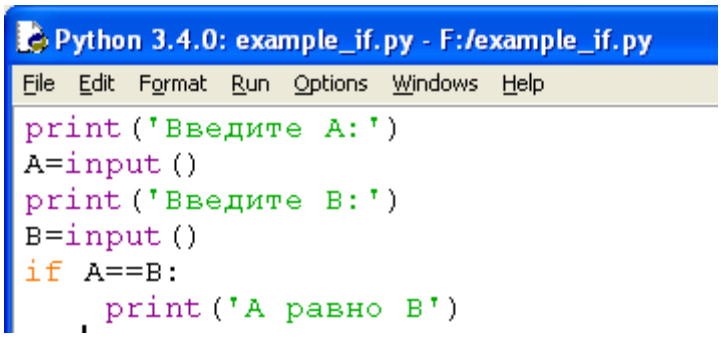
Логическое выражение — конструкция языка программирования, результатом вычисления которой является «истина» или «ложь».

Если это выражение истинно, то выполняются инструкции, определяемые данным оператором. Выражение является истинным, если его результатом является число не

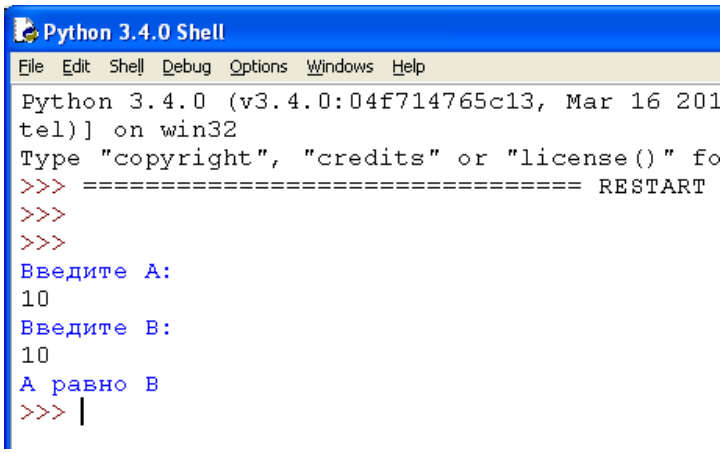
равное нулю, непустой объект, либо логическое True. После выражения нужно поставить двоеточие “:”.

ВАЖНО: блок кода, который необходимо выполнить, в случае истинности выражения, отделяется четырьмя пробелами слева!

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое выражение A==B истинно, то выводится соответствующее сообщение.



Пример программы на Python



Результат выполнения программы с использованием условного оператора if

2. Конструкция if – else

Бывают случаи, когда необходимо предусмотреть альтернативный вариант выполнения программы. Т.е. при истинном условии нужно выполнить один набор инструкций, при ложном – другой. Для этого используется конструкция if – else.

Синтаксис оператора if – else выглядит так:

if логическое выражение:

команда_1

команда_2

...

команда_n

else:

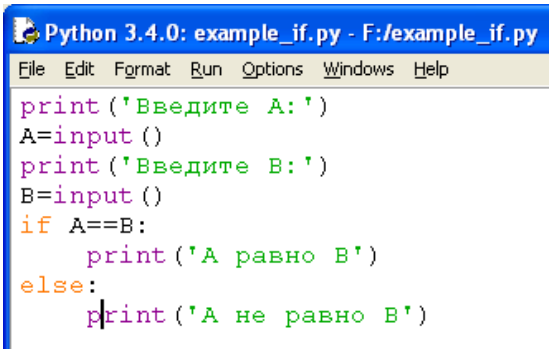
команда_1

команда_2

...

команда_n

Программа запрашивает у пользователя два числа, затем сравнивает их и если числа равны, то есть логическое выражение A==B истинно, то выводится соответствующее сообщение. В противном случае выводится сообщение, что числа не равны.



Пример программы на Python

```
>>> ================================ RESTART
>>>
Введите A:
10
Введите B:
5
A не равно B
>>>
```

Результат выполнения программы с использованием условного оператора if-else

3. Конструкция if – elif – else

Для реализации выбора из нескольких альтернатив можно использовать конструкцию if – elif – else.

Синтаксис оператора if – elif – else выглядит так:

if логическое выражение_1:

команда_1

команда_2

...

команда_n

elif логическое выражение_2:

команда_1

команда_2

...

команда_n

elif логическое выражение_3:

команда_1

команда_2

...

команда_n

else:

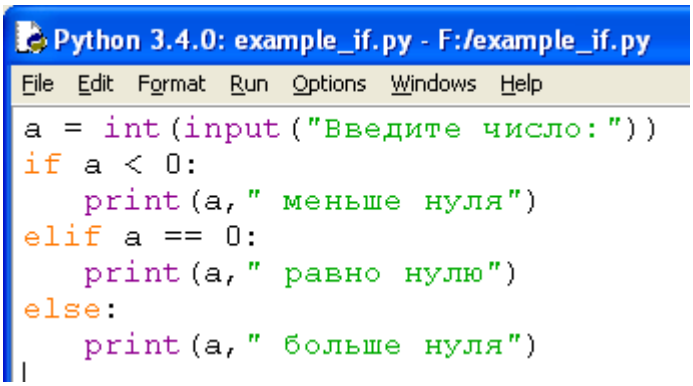
команда_1

команда_2

...

команда_n

Программа запрашивает число у пользователя и сравнивает его с нулём a<0. Если оно меньше нуля, то выводится сообщение об этом. Если первое логическое выражение не истинно, то программа переходит ко второму - a==0. Если оно истинно, то программа выведет сообщение, что число равно нулю, в противном случае, если оба вышеуказанных логических выражения оказались ложными, то программа выведет сообщение, что введённое число больше нуля.



Пример программы на Python

```
Введите число:41
41  больше нуля
>>> ================================ RESTART =
>>>
Введите число:-5
-5  меньше нуля
>>> ================================ RESTART =
>>>
Введите число:0
0  равно нулю
>>>
```

Результат выполнения программы с использованием условного оператора if-elif-else

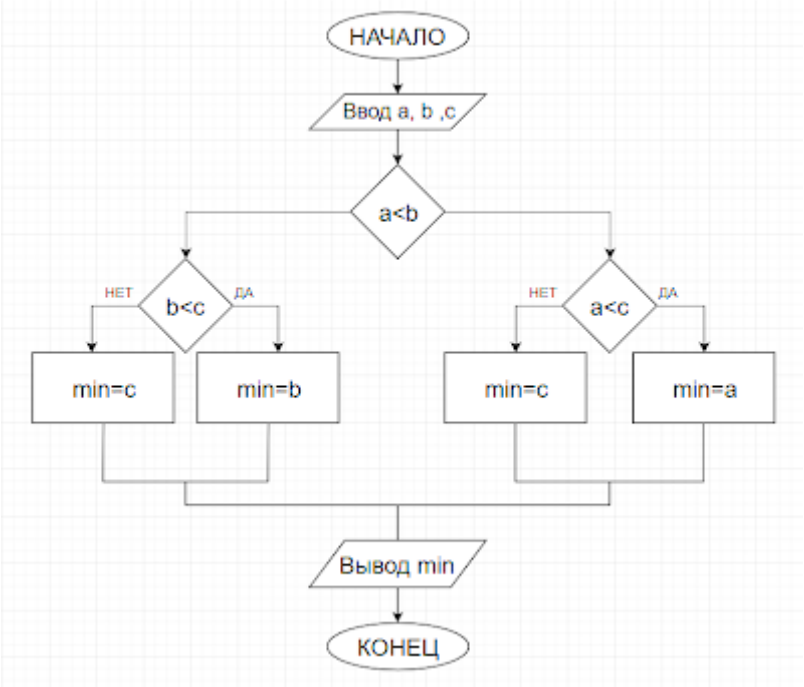
Пример

Вариант 0

Дано 3 числа. Найти минимальное среди них и вывести на экран.

Решение

Для простоты построим блок-схему задачи.



Командами

a=input('')

b=input('')

c=input('')

введём три числа, присвоив значения переменным a, b, c.

Условной конструкцией if-else проверим на истинность логическое выражение a<b. Если оно истинно, то переходим на проверку логического выражения a<c. Если оно истинно, то переменной "y" присвоим значение переменной "a", т.е. "a" будет минимальным, а иначе "y" присвоится значение переменной "c".

Если в начале логическое выражение a<b оказалось ложным, то переходим на проверку другого логического выражения b<c.

Если оно истинно, то "y" присвоится значение переменной "b", иначе "c".

Командой print() выводим минимальное значение.

Пример программы:

```
#нахождение минимального из 3-х чисел
a=input('Введите целое число \n')
b=input('Введите целое число \n')
c=input('Введите целое число \n')
if a<b:
    if a<c:
        y=a
    else:
        y=c
else:
    if b<c:
        y=b
    else:
        y=c
print('Минимальное:',y)
```

Результат выполнения программы:

```
Введите целое число
2
Введите целое число
5
Введите целое число
1
Минимальное: 1
```

Задания для самостоятельной работы (по вариантам)

Вариант 1

Даны три целых числа. Выбрать из них те, которые принадлежат интервалу [1,3].

Вариант 2

Дан номер года (положительное целое число). Определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

Вариант 3

Написать программу вычисления стоимости покупки с учетом скидки. Скидка в 3% предоставляется в том случае, если сумма покупки больше 500 руб., в 5% - если сумма больше 1000 руб.

Вариант 4

Написать программу, которая бы по введенному номеру единицы измерения (1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер) и массе M выдавала соответствующее значение массы в килограммах.

Вариант 5

Найти косинус минимального из 4 заданных чисел.

Вариант 6

Вывести на экран синус максимального из 3 заданных чисел.

Вариант 7

Даны три стороны одного треугольника и три стороны другого треугольника. Определить, будут ли эти треугольники равновеликими, т. е. имеют ли они равные площади. Если это не так, то вывести «Foul!!!»

Вариант 8

Составьте программу подсчёта площади равнобедренного треугольника. Если площадь треугольника чётная, разделить её на 2, в противном случае вывести сообщение «Не могу делить на 2!»

Вариант 9

Составить программу, которая по данному числу (1-12) выводит название соответствующего ему месяца на английском языке.

Вариант 10

Составить программу, осуществляющую перевод величин из радианной меры в градусную или наоборот. Программа должна запрашивать, какой перевод нужно осуществить, и выполнять указанное действие.

Вариант 11

Дано три числа. Найти количество положительных чисел среди них;

Вариант 12

Если действительные числа x и y – одного знака, найти их среднее геометрическое, в противном случае найти их среднее арифметическое.

Вариант 13

Определить, существует ли прямоугольный треугольник со сторонами x,y,z. Если – да, вычислить его площадь.

Вариант 14

Определить, существует ли треугольник с длинами сторон a, b, c. Если – да, вычислить его площадь по формуле Герона.

Формула Герона имеет вид:

S=p(p-a)(p-b)(p-c), где p=12(a+b+c)

Вариант 15

Вычислить значение функции f(x), если

$$f(x)=\begin{cases} 0{,}5-\sqrt[4]{|x|}, & x\geq 0 \\ \dfrac{\sin^2 x^2}{|x+1|}, & x<0 \end{cases}$$

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;
2. Цель занятия;

3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4. Список используемых источников;

5. Выводы и предложения;

6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Какой вычислительный процесс называется разветвляющимся?

2. Опишите синтаксис оператора if.

3. Дайте определение понятию «логическое выражение»

4. Опишите конструкцию if-elif-else.

5. Что позволяет выполнить оператор ветвления if.

Практическое занятие № 7 Линейные и условные алгоритмы. (составление трассировочных таблиц) Описание алгоритмов с помощью блок-схем Проверка условий в Python. Синтаксис If,If-else,if-elif-else. Составление программ с проверкой условий.

Цель занятия:

1. Познакомиться с циклическими конструкциями

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

В Python существуют два типа цикличных выражений:

- Цикл while
- Цикл for

1. Цикл while в Python

Инструкция while в Python повторяет указанный блок кода до тех пор, пока указанное в цикле логическое выражение будет оставаться истинным.

Синтаксис цикла while:

while логическое выражение:

команда 1

команда 2

...

команда n

После ключевого слова while указывается условное выражение, и пока это выражение возвращает значение True, будет выполняться блок инструкций, который идет далее.

Все инструкции, которые относятся к циклу while, располагаются на последующих строках и должны иметь отступ от начала строки (4 пробела).

```
#! Программа по вычислению факториала
number = int(input("Введите число: "))
i = 1
factorial = 1
while i <= number:
    factorial *= i
    i += 1
print("Факториал числа", number, "равен", factorial)
```

Пример программы на Python

```
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (In
tel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Введите число: 5
Факториал числа 5 равен 120
>>>
```

Результат выполнения программы с использованием циклического оператора while

2.Цикл for в Python:

Цикл for в Python обладает способностью перебирать элементы любого комплексного типа данных (например, строки или списка).

Синтаксис цикла for:

for int in range():

команда 1

команда 2

...

команда

Переменной int присваивается значение первого элемента функции range(), после чего выполняются команды. Затем переменной int присваивается следующее по порядку значение и так далее до тех пор, пока не будут перебраны все элементы функции range().

Функция range() является универсальной функцией Python для создания списков (list) содержащих арифметическую прогрессию. Чаще всего она используется в циклах for.

range(старт, стоп, шаг) - так выглядит стандартный вызов функции range() в Python. По умолчанию старт равняется нулю, шаг единице.

Вариант 0

1. Найти сумму n элементов следующего ряда чисел: 1 -0.5 0.25 -0.125 ... n. Количество элементов (n) вводится с клавиатуры. Вывести на экран каждый член ряда и его сумму. Решить задачу используя циклическую конструкцию for.

Решение:

В данном случае ряд чисел состоит из элементов, где каждый следующий меньше предыдущего в два раза по модулю и имеет обратный знак. Значит, чтобы получить следующий элемент, надо предыдущий разделить на -2.

Какой-либо переменной надо присвоить значение первого элемента ряда (в данном случае это 1). Далее в цикле добавлять ее значение к переменной, в которой накапливается сумма, после чего присваивать ей значение следующего элемента ряда, разделив текущее значение на -2. Цикл должен выполняться n раз.

```
n=int(input('Введите количество элементов последоваетльности: '))
x=1
s=0
print(x)
for i in range(n):
    s+=x
    x/=-2
    print(x)
print('Сумма ряда:',s)
```

Пример программы с циклом for

```
Введите количество элементов последоваетльности: 5
1
-0.5
0.25
-0.125
0.0625
-0.03125
Сумма ряда: 0.6875
```

Результат выполнения программы

2. Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

Решение:

Для начала введём целое число командой int(input(текст сообщения)).

Затем зададим переменной i значение 2. Переменная i выполняет роль счётчика. Если задать ей значение 1, то условие задачи не будет выполнено, а результатом всегда будет 1.

В цикле while в качестве логического выражения используется команда n%i сравниваемая с нулём. Таким образом, если остаток от деления введённого числа на текущее значение i не равно нулю, то счётчик увеличивается на 1, а если равно нулю цикл заканчивается и командой print() выводится сообщение и значение i.

```
n = int(input('Введите целое число не меньшее 2\n'))
i = 2
while n%i != 0:
    i+=1
print('наименьший натуральный делитель:',i)
```

Пример программы с циклом while

```
Введите целое число не меньшее 2
49
наименьший натуральный делитель: 7
```

Результат выполнения программы

Вариант 1

1. Дано вещественное число – цена 1 кг конфет. Вывести стоимость 1, 2, … 10 кг конфет. Решить задачу используя циклическую конструкцию for.

2. Дана непустая последовательность целых чисел, оканчивающаяся нулем. Найти: а) сумму всех чисел последовательности; б) количество всех чисел последовательности

Решить задачу используя циклическую конструкцию while.

Вариант2

1. Даны два числа A и B (A<B). Найти сумму всех целых чисел от A до B включительно. Решить задачу используя циклическую конструкцию for.

2. Дана последовательность отрицательных целых чисел, оканчивающаяся положительным числом. Найти среднее арифметическое всех чисел последовательности (без учета положительным числа).

Решить задачу используя циклическую конструкцию while.

Вариант 3

1. Даны два числа A и B (A<B). Найти сумму квадратов всех целых чисел от A до B включительно. Решить задачу используя циклическую конструкцию for.

2. Дана последовательность из n целых чисел. Первое число в последовательности чётное. Найти сумму всех идущих подряд в начале последовательности чётных чисел. Условный оператор не использовать

Решить задачу используя циклическую конструкцию while.

Вариант 4

1. Найти среднее арифметическое всех целых чисел от a до 200 (значения a и b вводятся с клавиатуры; a≤ 200). Решить задачу используя циклическую конструкцию for.

2. Дана последовательность из n вещественных чисел, начинающаяся с положительного числа. Определить, какое количество положительных чисел записано в начале последовательности. Условный оператор не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант 5

1. Найти сумму всех целых чисел от a до b (значения a и b вводятся с клавиатуры; b ≥a). Решить задачу используя циклическую конструкцию for.

2.Дано целое число N (> 0), являющееся некоторой степенью числа 2: N = 2^K. Найти целое число K — показатель этой степени.

Решить задачу используя циклическую конструкцию while.

Вариант 6

1. Найти сумму квадратов всех целых чисел от a до 50 (значение a вводится с клавиатуры; 0 ≤a≤50). Решить задачу используя циклическую конструкцию for.

2.Дано целое число N (> 1). Найти наименьшее целое число K, при котором выполняется неравенство 5^K > N.

Решить задачу используя циклическую конструкцию while.

Вариант 7

1. Дана непустая последовательность целых чисел, оканчивающаяся нулем.

Найти:

а) сумму всех чисел последовательности;

б) количество всех чисел последовательности.

Решить задачу используя циклическую конструкцию for.

2.Дано целое число N (> 1). Найти наибольшее целое число K, при котором выполняется неравенство 2^K > N.

Решить задачу используя циклическую конструкцию while.

Вариант 8

1. Дана последовательность из n вещественных чисел. Первое число в последовательности нечетное. Найти сумму всех идущих подряд в начале последовательности нечетных чисел. Условный оператор не использовать. Решить задачу используя циклическую конструкцию for.

2.Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

Решить задачу используя циклическую конструкцию while.

Вариант 9

1. Среди чисел 1, 4, 9, 16, 25, ... найти первое число, большее n. Решить задачу используя циклическую конструкцию for.

2.Среди чисел 1, 5 10, 16, 23, ... найти первое число, большее n. Условный оператор не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант 10

1. Известны оценки по физике каждого из 20 учеников класса. Определить среднюю оценку. Решить задачу используя циклическую конструкцию for.

2.Дано число A (> 1). Вывести наибольшее из целых чисел K, для которых сумма 1 + 1/2 + … + 1/K будет меньше A, и саму эту сумму.

Решить задачу используя циклическую конструкцию while.

Вариант 11

1. Известно сопротивление каждого из элементов электрической цепи. Все элементы соединены последовательно. Определить общее сопротивление цепи. Решить задачу используя циклическую конструкцию for.

2.Дано целое число N (> 0). Найти наибольшее целое число K, квадрат которого не превосходит N: K^2 ≤ N. Функцию извлечения квадратного корня не использовать.

Решить задачу используя циклическую конструкцию while.

Вариант 12

1. Известны оценки по физике каждого ученика двух классов. Определить среднюю оценку в каждом классе. Количество учащихся в каждом классе одинаковое. Решить задачу используя циклическую конструкцию for.

2. Выведите на экран для числа 2 его степени от 0 до 20

Решить задачу используя циклическую конструкцию while.

Вариант 13

1. В области 12 районов. Известны количество жителей (в тысячах человек) и площадь (в км2) каждого района. Определить среднюю плотность населения по области в целом. Решить задачу используя циклическую конструкцию for.

2.Мой богатый дядюшка подарил мне один доллар в мой первый день рождения. В каждый день рождения он удваивал свой подарок и прибавлял к нему столько долларов, сколько лет мне исполнилось. Написать программу, указывающую, к какому дню рождения подарок превысит 100$.

Решить задачу используя циклическую конструкцию while.

Вариант 14

1. Одноклеточная амеба каждые 3 часа делится на 2 клетки. Определить, сколько клеток будет через 3, 6, 9, ..., 24 часа, если первоначально была одна амеба. Решить задачу используя циклическую конструкцию for.

2. Вывести таблицу значений функции y = -0.5x + x. Значения аргумента (x) задаются минимумом, максимумом и шагом. Например, если минимум задан как 1, максимум равен 3, а шаг 0.5. То надо вывести на экран изменение x от 1 до 3 с шагом 0.5 (1, 1.5, 2, 2.5, 3) и значения функции (y) при каждом значении x.

Решить задачу используя циклическую конструкцию while.

Вариант 15

1. Начав тренировки, лыжник в первый день пробежал 10 км. Каждый следующий день он увеличивал пробег на 10% от пробега предыдущего дня. Определить:

а) пробег лыжника за второй, третий, ..., десятый день тренировок;

б) какой суммарный путь он пробежал за первые 7 дней тренировок.

Решить задачу используя циклическую конструкцию for.

2. Найти сумму и произведение цифр, введенного целого числа. Например, если введено число 325, то сумма его цифр равна 10 (3+2+5), а произведение 30 (3*2*5).

Решить задачу используя циклическую конструкцию while.

Выводы и предложения о проделанной работе

Содержание отчета:

1.Наименование практического занятия;

2.Цель занятия;

3.Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4.Список используемых источников;

5.Выводы и предложения;

6.Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Какие типы цикличных выражений существуют в Python?

2. Какой способностью обладает цикл for в Python?

3. Напишите синтаксис цикла while

Практическое занятие № 8 Циклические алгоритмы (составление трассировочных таблиц). Описание алгоритма с помощью блок-схем Реализация циклических алгоритмов в Python. Синтаксис цикла с предусловием и постусловием. Синтаксис цикла с параметром.

Цель занятия:

1. Познакомиться с методами работы со строками

Исходные данные: теоретический материал, программа Python

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

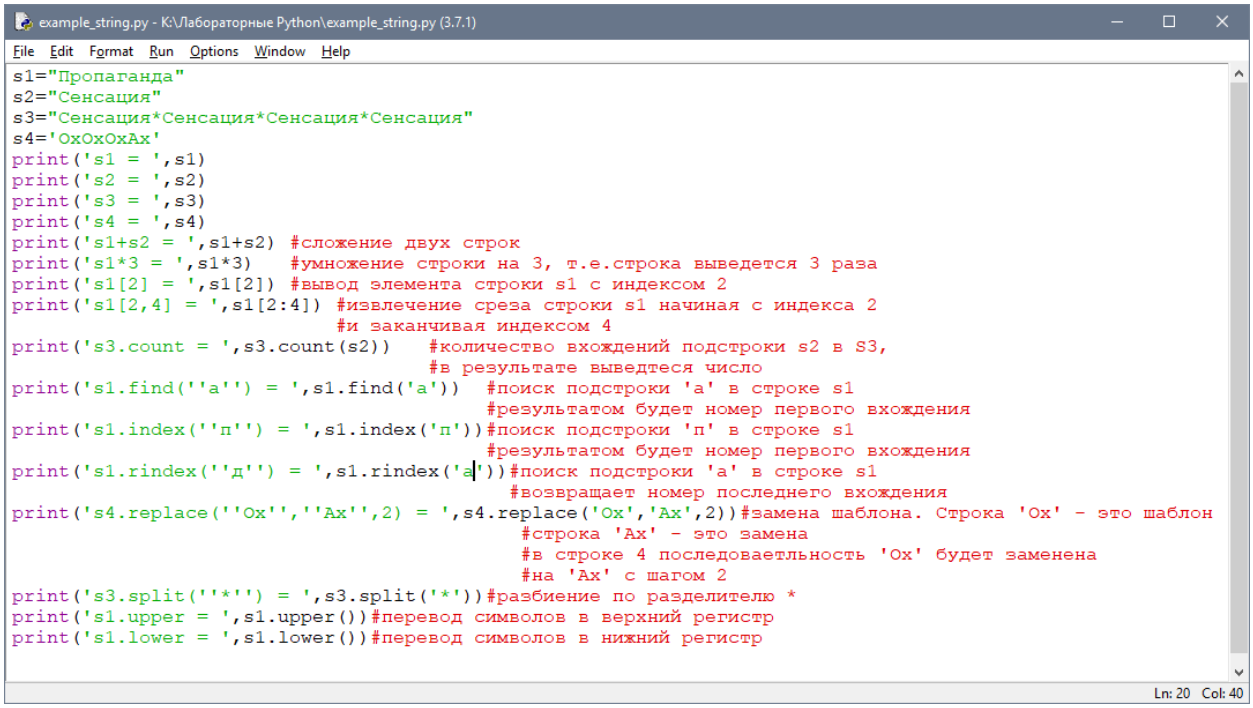
Теоретическая часть

Строка — базовый тип, представляющий из себя неизменяемую последовательность символов; str от «string» — «строка».

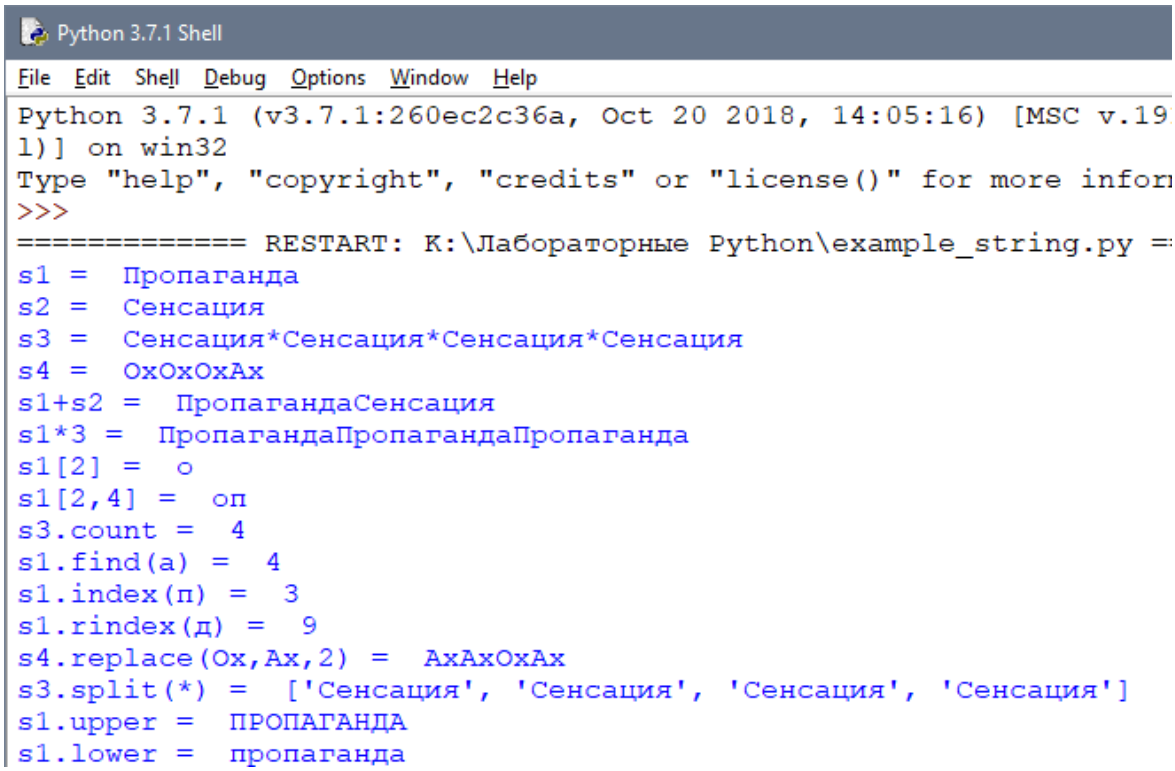
Функции и методы работы со строками

Функция или метод	Назначение
S1 + S2	Конкатенация (сложение строк)
S1 * 3	Повторение строки
S[i]	Обращение по индексу
S[i:j:step]	Извлечение среза
len(S)	Длина строки
S.join(список)	Соединение строк из последовательности str через разделитель, заданный строкой
S1.count(S[, i, j])	количество вхождений подстроки s в строку s1. Результатом является число. Можно указать позицию начала поиска i и окончания поиска *j*
S.find(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или -1
S.index(str, [start],[end])	Поиск подстроки в строке. Возвращает номер первого вхождения или вызывает ValueError
S.rindex(str, [start],[end])	Поиск подстроки в строке. Возвращает номер последнего вхождения или вызывает ValueError
S.replace(шаблон, замена)	Замена шаблона
S.split(символ)	Разбиение строки по разделителю
S.upper()	Преобразование строки к верхнему регистру
S.lower()	Преобразование строки к нижнему регистру

Ниже приведена программа, демонстрирующая использование функций и методов работы со строками.



Пример программы на Python



Результат выполнения программы с использованием функций и методов работы со строками

Пример

Вариант 0

Проверить, будет ли строка читаться одинаково справа налево и слева направо (т. е. является ли она палиндромом).

Решение

Сначала введём строку командой: s=input('Введите строку ').

Затем определим логическую переменную flag и присвоим ей значение 1: flag=1.

Для начала в введённой строке нужно удалить пробелы. Для этого воспользуемся циклической конструкцией for, которая выполнится столько раз, какую имеет длину строка. Длину строки определим функцией len(s).

В теле цикла будем проверять следующее условие: s[i]!=' '. Данное логическое выражение будет истинно в том случае, если i-ый элемент строки не будет равен пробелу, тогда выполнится команда следующая после двоеточия: string+=s[i].

К строке string, которая была объявлена в начале программы, будет добавляться посимвольно строка s, но уже без пробелов.

Для проверки строки на "палиндром" воспользуемся циклической конструкцией for.

Длина половины строки находится делением нацело на 2.Если количество символов нечетно, то стоящий в середине не учитывается, т.к. его сравниваемая пара - он сам.

Количество повторов цикла равно длине половины строки. Длину строки определим функцией len(s), где аргумент введённая нами строка s. Зная длину строки, можно вычислить количество повторов цикла. Для этого целочисленно разделим длину строки на 2: len(s)//2.

Для задания диапазона для цикла используем функцию range(), в которой аргументом будет являться половина длины строки: range(len(s//2)).

for i in range(len(s//2)).

Если символ с индексом i не равен "симметричному" символу с конца строки (который находится путем индексации с конца)

if s[i] != s[-1-i],

то переменной flag присваивается значение 0 и происходит выход из цикла командой break.

Далее, при помощи условной конструкции if-else в зависимости от значения flag либо - 0, либо -1 выводится сообщение, что строка палиндром, либо нет.

```
s=input('Введите строку \n')
flag=1
string=''
for i in range(len(s)):
    if s[i]!=' ':
        string+=s[i]
print(string)
for i in range(len(s)//2):
    if string[i]!=string[-i-1]:
        flag=0
        break
if flag: print('Палиндром')
else: print('не палиндром')
```

Пример программы на Python

```
Введите строку
а роза упала на лапу азора
арозаупаланалапуазора
Палиндром
```

Результат выполнения программы

Задания для самостоятельной работы (по вариантам)

Вариант 1

Дана строка, содержащая русскоязычный текст. Найти количество слов, начинающихся с буквы "е".

Вариант 2

В строке заменить все двоеточия (:) знаком процента (%). Подсчитать количество замен.

Вариант 3

В строке удалить символ точку (.) и подсчитать количество удаленных символов.

Вариант 4

В строке заменить букву(а) буквой (о). Подсчитать количество замен. Подсчитать, сколько символов в строке.

Вариант 5

В строке заменить все заглавные буквы строчными.

Вариант 6

В строке удалить все буквы "а" и подсчитать количество удаленных символов.

Вариант 7

Дана строка. Преобразовать ее, заменив звездочками все буквы "п", встречающиеся среди первых n/2 символов. Здесь n - длина строки.

Вариант 8

Дана строка, заканчивающаяся точкой. Подсчитать, сколько слов в строке.

Вариант 9

Определить, сколько раз в тексте встречается заданное слово.

Вариант 10

Дана строка-предложение на английском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.

Вариант 11

Дана строка. Подсчитать самую длинную последовательность подряд идущих букв «н». Преобразовать ее, заменив точками все восклицательные знаки.

Вариант 12

Дана строка. Вывести все слова, оканчивающиеся на букву "я".

Вариант 13

Дана строка символов, среди которых есть одна открывающаяся и одна закрывающаяся скобки. Вывести на экран все символы, расположенные внутри этих скобок.

Вариант 14

Дана строка. Вывести все слова, начинающиеся на букву "а" и слова оканчивающиеся на букву "я".

Вариант 15

Дана строка текста. Подсчитать количество букв «т» в строке

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;
2. Цель занятия;

3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;

4. Список используемых источников;

5. Выводы и предложения;

6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Что называется строкой в Python

2. Перечислите функции и методы работы со строками.

3. Для чего служит функция S.split?

4. Для чего используется функция range()?

Практическое занятие № 9 Моделирование в векторном редакторе. Знакомство с интерфейсом. Создание изображений из графических примитивов. Работа с объектами векторного редактора

Цель занятия:

1. Познакомиться с понятиями графики

Исходные данные: теоретический материал, программа

Содержание и порядок выполнения:

1. Изучить теоретическую часть;

2. Выполнить задания.

Теоретическая часть

С помощью графики в Python можно рисовать фигуры и изображения, создавать анимацию, визуализировать математические вычисления в Python. В программах python можно использовать элементы графики в компьютерных играх.

Для работы с графикой в Python нужно импортировать модуль graphics.py.

Чтобы начать работу с графикой в Python, нужно создать окно для графики

Графический объект = GraphWin("Название окна для графики", ширина окна для графики в пикселях, высота окна для графики в пикселях)

GraphWin это ключевое слово, которое задаёт окно графической области, в котором будут отображаться графические объекты.

В качестве параметров этой функции указывается название окон для графики, ширина и высота окон в пикселях.

После запуска программы откроется окно для графики, где будут отображаться графические объекты.

Вся работа с графикой будет осуществляться нами через графические объекты.

Общая структура работы с графическими объектами в Python

Графический_объект.Вызов_команды()

Общая структура графической программы в Python.

```
# импортируем библиотеку graphics
fromgraphicsimport *
# создаём окно для графики
win = GraphWin("Окно для графики", 400, 400)
# ...рисуем все объекты...
win.getMouse() # ждём нажатия кнопки мыши
win.close() # закрываем окно для графики
```

В этой программе мы определили объект графическое окно win и открыли его с размерами 400 на 400 пикселей.

Команда win.getMouse() ожидает нажатие на любую кнопку мыши, наведённую на область окна win.

win.close() закрывает окно для графических объектов win.

С помощью модуля graphics.py в программах на Python можно отобразить точку, линию, окружность, прямоугольник, эллипс и многоугольник, вывести текст на экран.

Чтобы задать расположение объекта в графическом окне Python, необходимо указать его координаты в системе координат Python. Начало координат находится в левом верхнем углу окна для графики.

Положительное направление оси X определяется слева направо, оси Y определяется сверху вниз. Чем больше значение координаты X, тем правее точка, чем больше значение координаты Y, тем точка ниже. Чтобы нарисовать заданный объект obj в окне для графики win, нужно использовать процедуру obj.draw(win)

Перед тем, как рисовать графические объекты в заданном графическом окне, нужно их задать.

Для задания точки в Python используется функция Point(x, y)

obj = Point(x, y)

x, y – координаты точки.

Пример программы на Python, которая задаёт и отображает точку в графическом окне.

```
fromgraphicsimport * # импортируем библиотеку graphics
win = GraphWin("Окно для графики", 400, 400) # создаём окно для графики размером 400 на 400 пикселей
obj = Point(50, 50) # создаём точку в координатах (50, 50)
obj.draw(win) # отображаем точку в окне для графики
```

```
win.getMouse() # ждём нажатия кнопки мыши
win.close() # закрываем окно для графики
```

Для задания отрезка в Python используется функция Line(объект точка первого конца, объект точка второго конца)

```
obj = Line(Point(x1, y1), Point(x2, y2))
```

x1, y1 – координаты начала отрезка линии, x2, y2 – координаты конца отрезка линии.

Чтобы задать цвет рисования линий в Python используется команда obj.setOutline("цвет")

Пример программы на Python, которая отображает линию в графическом окне.

```
from graphics import *
win = GraphWin("Окнодляграфики", 400, 400)
obj = Line(Point(50, 50), Point(350, 350))
obj.setOutline("blue")
obj.draw(win)
win.getMouse()
win.close()
```

Для отображения окружности в Python используется

```
obj = Circle(Point(x, y), R)
```

x, y – координаты центра окружности,

R – радиус окружности.

Пример программы на Python, которая отображает окружность в графическом окне.

```
from graphics import *
win = GraphWin("Окнодляграфики", 400, 400)
obj = Circle(Point(200, 200), 50)
obj.draw(win)
win.getMouse()
win.close()
```

Дляотображенияпрямоугольника вPython используетсяпроцедура

```
obj = Rectangle(Point(x1, y1), Point(x2, y2))
```

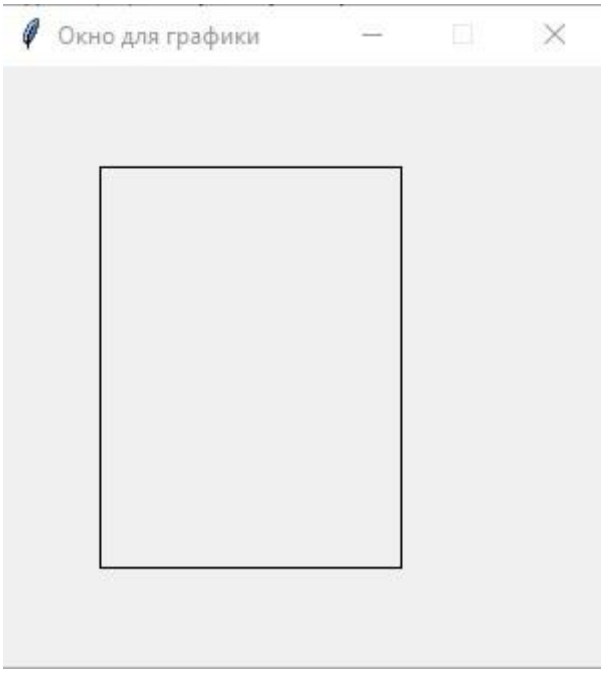
x1, y1 – координатылевоговерхнегоуглапрямоугольника,

x2, y2 – координатыправогонижнегоуглапрямоугольника

ПримерпрограммынаPython, котораяотображаетпрямоугольниквграфическомокне.

```
fromgraphicsimport *
win = GraphWin("Окнодляграфики", 300, 300)
obj = Rectangle(Point(50, 50), Point(200, 250))
```

```
obj.draw(win)
win.getMouse()
win.close()
```



Для отображения эллипса в Python используется процедура

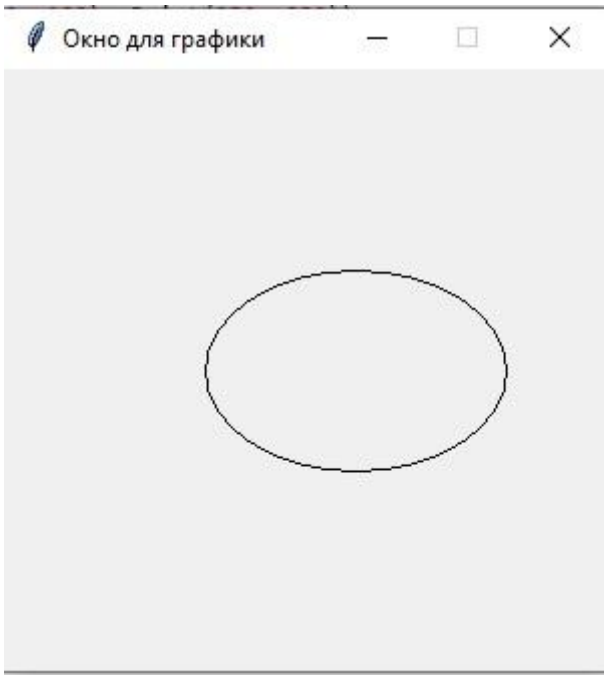
```
obj = Oval(Point(x1, y1), Point(x2, y2))
```

x1, y1 – координаты первого фокуса эллипса,

x2, y2 – координаты второго фокуса эллипса.

Пример программы на Python, которая отображает эллипс в графическом окне.

```
from graphics import *
win = GraphWin("Окнодлярафики", 300, 300)
obj = Oval(Point(100, 100), Point(250, 200))
obj.draw(win)
win.getMouse()
win.close()
```



Для отображения многоугольника в Python используется процедура

obj = Polygon(Point(x1, y1), Point(x2, y2),…, Point(xn, yn))

x1, y1, x2, y2,…, xn, yn – координаты вершин многоугольника.

Пример программы на Python, которая отображает пятиугольник в графическом окне.

```
from graphics import *
win = GraphWin("Окнодляграфики", 400, 400)
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150), Point(70, 70))
obj.draw(win)
win.getMouse()
win.close()
```

Определение цвета закрашивания графического объекта в Python

Чтобы задать цвет закрашивания графического объекта в python используется команда obj.setFill("цвет")

Пример программы на Python, которая рисует закрашенную синюю окружность

```
from graphics import *
win = GraphWin("Окнодляграфики", 400, 400)
obj = Circle(Point(200, 200), 50)
obj.setFill("blue")
obj.draw(win)
win.getMouse()
win.close()
```

Для редактирования границ объектов в Python используются процедуры setOutline(“цвет границы”) иsetWidth(ширина границы).

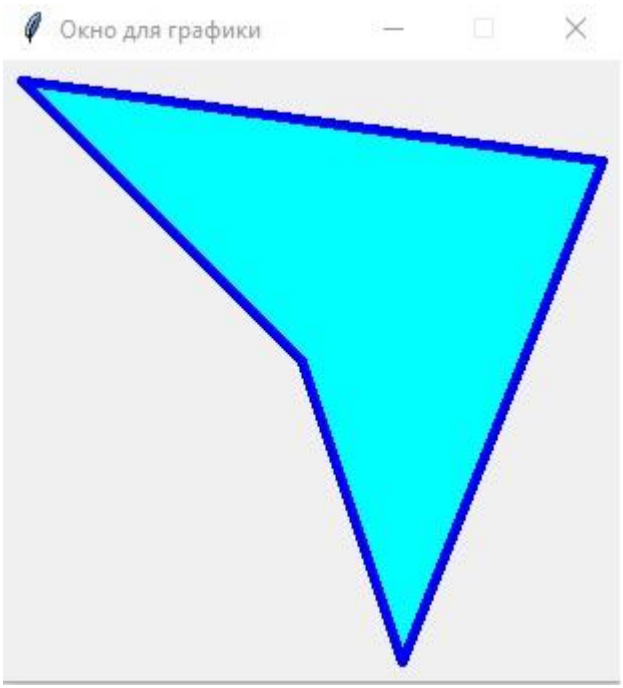
obj.setOutline("blue") – объект obj отображается с границей синего цвета.

obj.setWidth(5) – объект obj отображается с шириной границы 5 пикселей.

По умолчанию графический объект в Python будет отображаться с границами чёрного цвета шириной 1 пиксель.

Пример программы на Python, которая отображает фигуру с синей границей и заливкой в графическом окне.

```
from graphics import *
win = GraphWin("Окнодляграфики", 310, 310)
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150), Point(70, 70))
obj.setOutline("blue")
obj.setWidth(5)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
win.close()
```



Чтобы переместить графический объект в Python, используется процедура move(dx, dy), которая перемещает объект на dx пикселей вправо и dy пикселей вниз.

obj.move(50, 50) смещает объект obj на 50 пикселей вправо и 50 пикселей вниз.

Для клонирования объектов используется процедура clone()

newObj = obj.clone()

С помощью этой команды создаётся новый графический объект newObj, который идентичен объекту obj.

Для удаления фигур с экрана используется процедура undraw() Объект удаляется с графического окна, но не удаляется из памяти.

obj.undraw()

Пример программы на Python, которая удаляет, перемещает и копирует объект в графическом окне.

```
fromgraphicsimport *
win = GraphWin("Окно для графики", 400, 400)
obj = Polygon(Point(30, 10), Point(30, 50), Point(20, 30), Point(15, 30), Point(7, 7))
obj.setOutline("blue")
obj.setWidth(2)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
obj.undraw()
win.getMouse()
obj.draw(win)
obj.move(100, 100)
win.getMouse()
shape = obj.clone()
shape.move(-100, -100)
shape.draw(win)
win.getMouse()
win.close()
```

Для создания текста в графическом окне в Python используется команда

текстовый объект=Text(координаты точки размещения текста, “Текст”)

msg = Text(Point(50, 100), “Hello World!”)

На экран в точке с координатами (50, 100) выведется текст со строкой Hello World!

Для изменения размера текста используется команда текстовый объект.setSize(размер текста)

msg.setSize(12)

Цвет текста изменяется с помощью метода setTextColor(цвет)

msg.setTextColor(“black”)

Текст в графическом объекте можно заменить с помощью метода setText(“Текст”)

msg.setText(“Другой текст”)

Стиль текста изменяется с помощью процедуры setStyle(стиль)

msg.setStyle(“bold”)

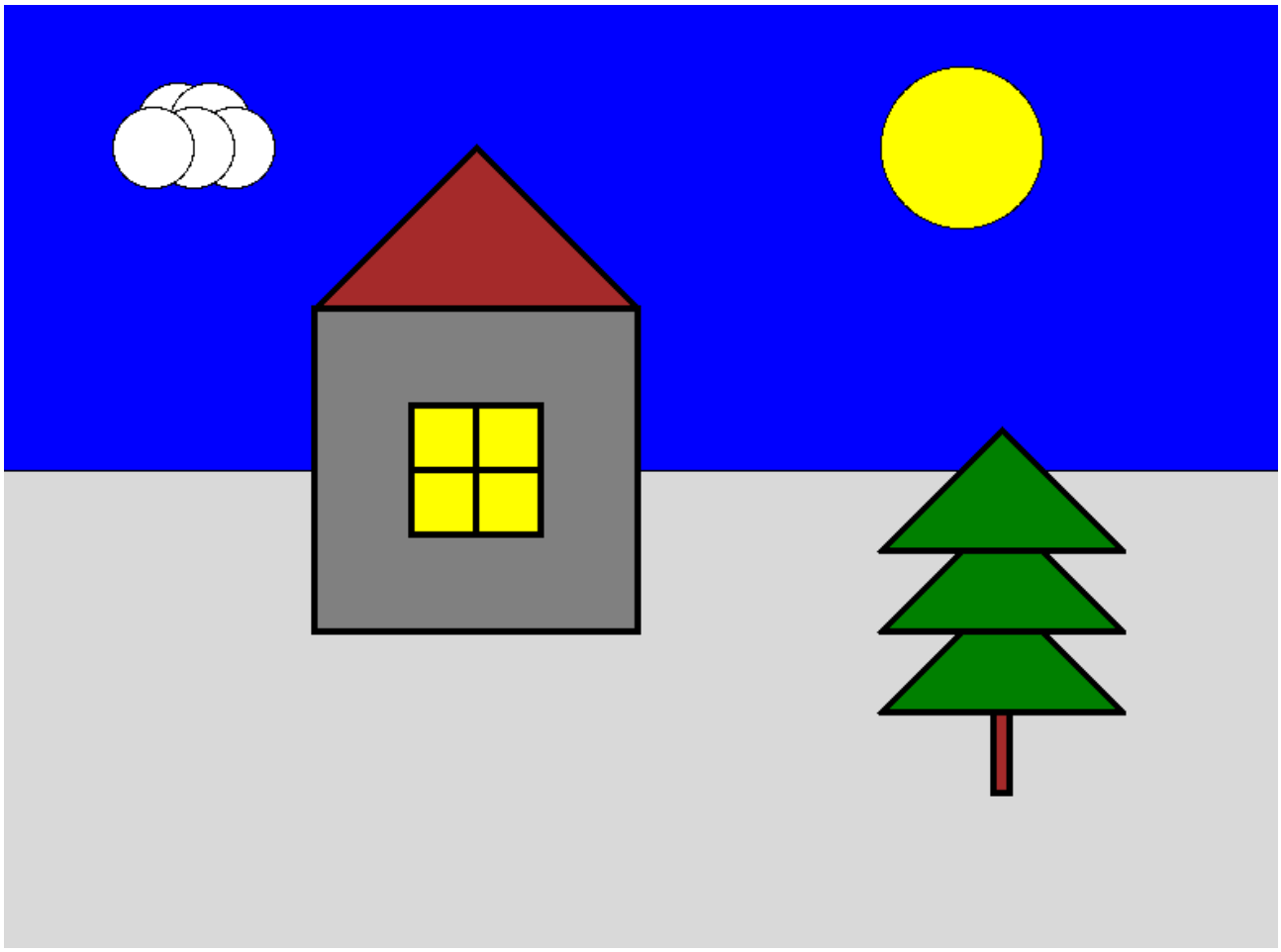
Стиль normal изменяет стиль текста на обычный, bold меняет стиль на полужирный, italic меняет стиль на курсив, bolditalic меняет стиль текста на полужирный курсив.

Пример программы на Python, которая отображает текст в графическом окне.

```
fromgraphicsimport *
win = GraphWin("Окнодляграфики", 400, 400)
obj = Polygon(Point(10, 10), Point(300, 50), Point(200, 300), Point(150, 150), Point(70, 70))
obj.setOutline("blue")
obj.setWidth(5)
obj.setFill("cyan")
obj.draw(win)
win.getMouse()
obj.undraw()
msg = Text(Point(200, 200), "Фигураудалиласьсэкрана")
msg.setSize(12)
msg.setTextColor("black")
msg.setStyle("bold italic")
msg.draw(win)
win.getMouse()
win.close()
```

Задание №1. Пейзаж

Используя полученные знания, нарисуйте любую статическую сцену, которая содержит не менее 5 различных объектов, состоящих из пяти и более примитивов. Проявите свою творческую натуру. Примером сцены может являться следующая картинка:



Задание №2 Измените вашу сцену так, чтобы объекты были нарисованы на пейзаже в других местах. Добавьте ещё два таких же облака, но так, чтобы все три облака выглядели естественно, не выстроившись в линейку.

Функции с параметрами

А теперь представьте, что в предыдущем задании вас попросили сделать не две копии, а сто?

Наивным решением будет написать сто почти одинаковых функций с измененными цифрами, но если мы вдруг внезапно захотим во всех этих объектах убрать какой-либо примитив — нам придется залезть в каждую такую функцию и изменить соответствующие строчки. Такой подход нежизнеспособен.

Рациональным выходом из подобной ситуации будет являться использование функций с параметрами. В физике положение объекта мы задавали с помощью координат, почему бы такой подход не распространить и на графические объекты?

Продемонстрируем, как этот код можно оптимизировать.

```
defdraw_eye(x,y,size):
eye=gr.Circle(gr.Point(x,y),size)
pupil=gr.Circle(gr.Point(x,y),size/2)

eye.setFill('red')
pupil.setFill('black')
```

```
eye.draw(window)
pupil.draw(window)

defdraw_angry_lecturer():
draw_face()
draw_eye(150,180,20)
draw_eye(250,180,14)
draw_eyebrows()
draw_mouth()
```

Как видите теперь, если мы хотим изменить конструкцию обоих глаз одновременно, нам достаточно изменить код в одном месте, и это изменение распространиться на все объекты, которые отрисовываются данной функцией.

Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия;
2. Цель занятия;
3. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»;
4. Список используемых источников;
5. Выводы и предложения;
6. Дата и подпись курсанта и преподавателя.

Вопросы для самопроверки

1. Что делает графический объект GraphWin?
2. Какая процедура используется для рисования прямоугольника?
3. Какая процедура используется для рисования эллипса?
4. Какая команда используется для отображения текста?

Практическое занятие № 10 Закраска рисунков и контуров

Цель занятия: Освоить способы создания блок-схем различных типов алгоритма в программе Dia.

*Исходные данные:*теоретический материал

Содержание и порядок выполнения задания

Изучить теоретическую часть

Выполнить задания

Теоретическая часть

Правила изображения блок-схем

1.При составлении блок-схем используются только строго определенные типы блоков:

a)*начало* и *конец* алгоритма;

b)*обработки* (внутри блока записываются формулы, обозначения операций и функций);

c)*условия* (внутри блока записываются условия выбора направления действия алгоритма);

d)*ввода/вывода* информации;

e)*вывода* информации на экран дисплея;

f)*вывода* информации на печатающее устройство;

g)*вычисления по подпрограмме* или стандартнойподпрограмме;

h)*начало цикла*;

i)*соединительный блок*;

j)*линии потока* (если поток направлен вниз или вправо стрелку не ставят).

2.Все блоки нумеруются. Номера ставятся вверху слева от блока (блоки "начало", "останов" и соединительные блоки не нумеруются).

3.Стрелки не ставят, если поток направлен сверху вниз или слева направо.

4.Каждый блок имеет единственную точку входа, кроме блока "начало", который не имеет входа.

5.Каждый безусловный блок имеет единственную точку выхода, кроме блока "останов", который не имеет выхода.

6.Условный блок имеет два, в отдельных случаях три выхода.

7.Выход условного блока можно пометить "да", "нет", "+", "–", 0, 1.

8.Линии, идущие на вход некоторого блока, могут соединяться, что соответствует переходу на конкретный этап вычислений после нескольких других этапов.

9.Линия, исходящая из входной точки блока, не может разветвляться на несколько направлений.

Правила построения алгоритмов на языке блок-схем

1.Блок-схема строится сверху вниз.

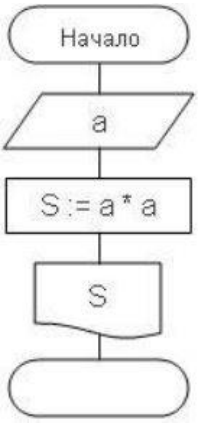
2.В любой блок-схеме имеется только один элемент, соответствующий началу алгоритма, и один элемент, соответствующий концу алгоритма.

3.Должен быть хотя бы один путь из начала блок-схемы к любому элементу.

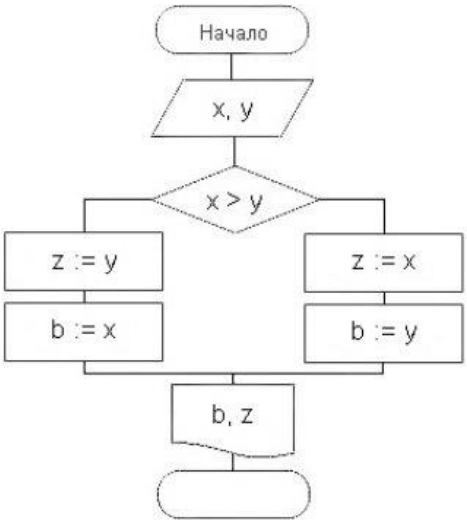
4.Должен быть хотя бы один путь от каждого элемента блок-схемы в конец блок-схемы.

Задание 1. Создание линейного алгоритма.

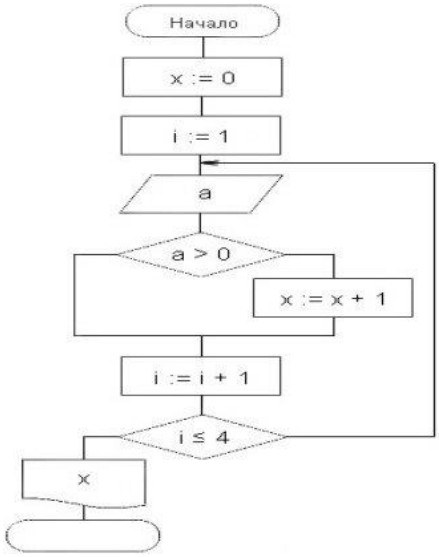
Запустить DIA. Создать алгоритм программы, с помощью фигур. Для этого в пункте меню Вставка выбираем Фигуры и соответствующий элемент библиотеки - блок-схемы.



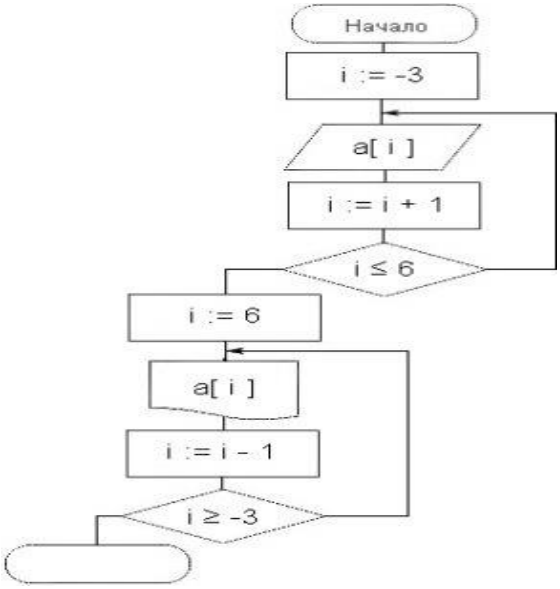
Задание 2. Создание алгоритма ветвления



Задание 3. Создание алгоритма цикла



Задание 4. Создание алгоритма массива



Выводы и предложения о проделанной работе

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Список используемых источников
5. Выводы и предложения
6. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Что называется алгоритмом?
2. Для чего предназначена программа Dia?
3. Какие инструменты есть в программе Dia для создания блок-схем различных алгоритмических конструкций?
4. Какие еще существуют библиотеки элементов в программе Dia?

Практическое занятие № 11Создание изображений с использованием переходов

Цель занятия. Получить практические навыки работы с графическим редактором GIMP. Создание геометрических фигур. Освоение инструментов рисования.

Оборудование: ПК, графический редактор Gimp.

Содержание и порядок выполнения задания

Изучите теоретический материал

Выполните задания.

Теоретический материал

Растровая графика

Растровое изображение — изображение, представляющее собой сетку пикселей или точек цветов (обычно прямоугольную) на компьютерном мониторе, бумаге и других отображающих устройствах и материалах. Важными характеристиками изображения являются:

количество пикселов — разрешение. Может указываться отдельно количество пикселов по ширине и высоте (1024*768, 640*480,…) или же, редко, общее количество пикселей (часто измеряется в мегапикселах);

количество используемых цветов или «глубина цвета» (эти характеристики имеют следующую зависимость: $N = 2^I$, где N - количество цветов, а I - глубина цвета);

цветовое пространство (цветовая модель) RGB, CMYK, XYZ, YCbCr и др.

Создается растровая графика фотоаппаратами, сканерами, непосредственно в растровом редакторе, также путем экспорта из векторного редактора или в виде скриншотов.

Достоинства

Растровая графика позволяет создать (воспроизвести) практически любой рисунок, вне зависимости от сложности, в отличие, например, от векторной, где невозможно точно передать эффект перехода от одного цвета к другому без потерь в размере файла.

Распространённость — растровая графика используется сейчас практически везде: от маленьких значков до плакатов.

Высокая скорость обработки сложных изображений, если не нужно масштабирование.

Растровое представление изображения естественно для большинства устройств ввода-вывода графической информации, таких как мониторы (за исключением векторных), матричные и струйные принтеры, цифровые фотоаппараты, сканеры.

Недостатки

Большой размер файлов с простыми изображениями.

Невозможность идеального масштабирования.

Невозможность вывода на печать на плоттер.

Из-за этих недостатков для хранения простых рисунков рекомендуют вместо даже сжатой растровой графики использовать векторную графику.

Форматы

Растровые изображения обычно хранятся в сжатом виде. В зависимости от типа сжатия может быть возможно или невозможно восстановить изображение в точности

таким, каким оно было до сжатия (сжатие без потерь или сжатие с потерями соответственно). Так же в графическом файле может храниться дополнительная информация: об авторе файла, фотокамере и её настройках, количестве точек на дюйм при печати и др.

BMP или WindowsBitmap — обычно используется без сжатия, хотя возможно использование алгоритма RLE.

GIF (GraphicsInterchangeFormat) — устаревающий формат, поддерживающий не более 256 цветов одновременно. Всё ещё популярен из за поддержки анимации, которая отсутствует в чистом PNG, хотя ПО начинает поддерживать APNG.

PCX устаревший формат, позволявший хорошо сжимать простые рисованые изображения (при сжатии группы подряд идущих пикселов одинакового цвета заменяются на запись о количестве таких пикселов и их цвете).

PNG (PortableNetworkGraphics) .

JPEG очень широко используемый формат изображений. Сжатие основано на усреднении цвета соседних пикселей(информация о яркости при этом не усредняется) и отбрасывании высокочастотных составляющих в пространственном спектре фрагмента изображения. При детальном рассмотрении сильно сжатого изображения заметно размытие резких границ и характерный муар вблизи них.

TIFF поддерживает большой диапазон изменения глубины цвета, разные цветовые пространства, разные настройки сжатия (как с потерями, так и без) и др.

RAW хранит информацию, непосредственно получаемую с матрицы цифрового фотоаппарата или аналогичного устройства без применения к ней каких-либо преобразований, а также хранит настройки фотокамеры. Позволяет избежать потери информации при применении к изображению различных преобразований (потеря информации происходит в результате округления и выхода цвета пиксела за пределы допустимых значений). Используется при съёмке в сложных условиях (недостаточная освещённость, невозможность выставить баланс белого и т.п.) для последующей обработки на компьютере (обычно в ручном режиме). Практически все полупрофессиональные и профессиональные цифровые фотоаппараты позволяют сохранять RAW изображения. Формат файла зависит от модели фотоаппарата, единого стандарта не существует

К программным средствам обработки растровой графики относятся растровые графические редакторы: GIMP, Paint.NET, TuxPaint, AdobePhotoshop, AdobeFireworks, CorelPhoto-Paint, CorelPaintShopPro, CorelPainter, MicrosoftPaint.

GIMP — многоплатформенное программное обеспечение для работы над изображениями. GIMP является акронимом, означающим GNU ImageManipulationProgram.

Редактор GIMP пригоден для решения множества задач по изменению изображений, включая ретушь фотографий, объединение и созданий изображений. Программа GIMP многофункциональна. Еѐ можно использовать как простой графический редактор, как профессиональное приложение по ретуши фотографий, как сетевую систему пакетной обработки изображений, как программу для рендеринга изображений, как преобразователь форматов изображения и т.д. GIMP спроектирован расширяемым при помощи дополнений, реализующих любые возможные функции. Передовой интерфейс для разработки сценариев позволяет легко автоматизировать выполнение любых задач любого уровня.

Одной из сильных сторон GIMP является его доступность из многих источников для многих операционных систем. GIMP входит в состав большинства дистрибутивов GNU/Linux. GIMP также доступен и для других операционных систем вроде MicrosoftWindows™ или Mac OS X™ от Apple (Darwin). GIMP — свободное программное обеспечение, выпускаемое под

лицензией GPL(GeneralPublicLicense). GPL предоставляет пользователям право доступа к исходному коду программ и право изменять его. Краткий обзор возможностей и функций GIMP

Полный набор инструментов для обработки растровой графики

Возможность работы с векторной графикой

Создание анимации

Работа с принтером и сканером

Захват изображений

Множество подключаемых модулей (plug-in)

Быстрое создание различных логотипов для web-дизайна

и многое другое...

Основное диалоговое окно GIMP

Основное окно состоит из нескольких основных элементов: инструментов и диалога цвета. Инструменты позволяют производить определенные действия над уже открытым изображением. Свойства любого инструмента можно вызвать двойным щелчком на его иконке.

Диалог цвета позволяет выбрать типы воздействия инструментов. Так, диалог цвета позволяет выбрать цвет пера и фона, а так же переключать их, нажав на стрелочки.

Для рисования в нашем распоряжении есть *Карандаш*, *Кисть*, *Ластик*, *Аэрограф*, *Штамп*, *Размыватель*, *Чернила*, *Осветление* и *Палец*. Инструменты Карандаш, Кисть, Ластик, Аэрограф чувствительны к размеру и виду кисти. Выбрать их можно в диалоге Кисти

Кисть также может работать и в других режимах: Добавление (обратный режиму вычитания), Осветление (операция деления) или Затемнение (операция умножения). С ее помощью вы также сможете изменять тон и яркость изображения. Есть возможность изменять размер кисти, ее жесткость, непрозрачность и цвет в зависимости от скорости

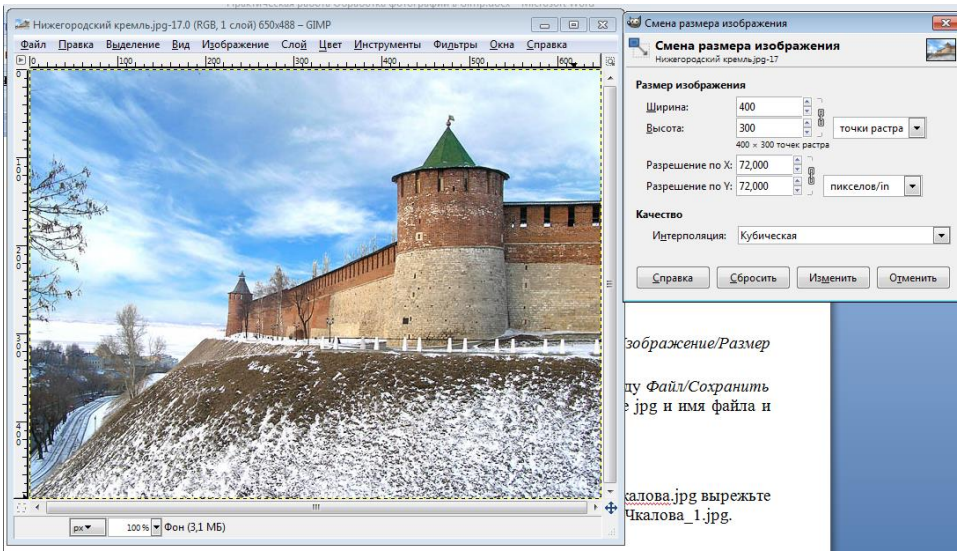
движения по холсту, в зависимости от силы нажима, да и просто кисть может изменять свои параметры случайно. В этих же режимах работают Карандаш, Аэрограф, Чернила.

Задание 1. Изменение размеров изображения. У изображения Нижегородский кремль.jpg изменить размеры, установив размер 400x300 и сохранив результат под именем Нижегородский кремль _1.jpg.

Алгоритм

Запустить программу Gimp.

Для изменения размеров изображения выполните команду *Изображение/Размер изображения*, интерполяция – *Кубическая*, нажмите *Изменить.*



Сохраните рисунок как Город_1.jpg. Для этого выполните команду *Файл/Сохранить как ….* В появившемся диалоговом окне выберите расширение jpg и имя файла и нажмите кнопку *Сохрани,* качество - 85.

Закройте рисунок.

Задание 2. Кадрирование изображения. Из изображения Памятник_Чкалова.jpg вырежьте памятник и сохраните результат под именем Памятник_Чкалова_1.jpg.

Алгоритм

Загрузить файл Памятник_Чкалова.jpg.jpg.

Для выполнения кадрирования выберите инструмент *Кадрирование* и выделите прямоугольную область памятника.

Сохраните рисунок.

Задание 3. Поворот изображения. Фотографию Пизанская башня.jpg приведите в порядок – выпрямите башню и сохраните под именем Пизанская башня_1.jpg.

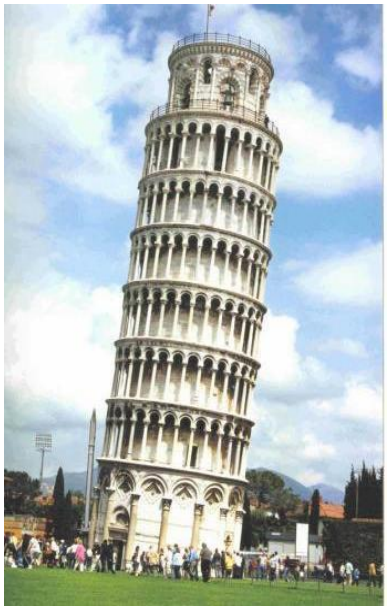
Алгоритм

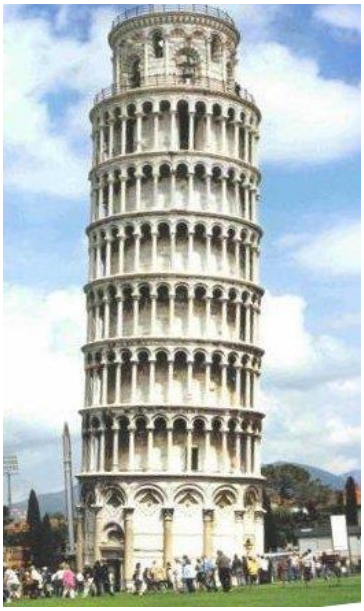
Загрузить файл Пизанская башня.jpg.

Для выполнения поворота выполните команду *Инструмент/Инструменты преобразования/Вращение,* угол вращения -7 градусов.

Кадрируйте полученное изображение.

Сохраните рисунок как Пизанская башня_1.jpg.





Задание 4. Коррекция изображения. Из изображения Медведь.jpg удалите медведя и сохраните рисунок под именем Медведь_0.jpg.

Алгоритм

Загрузить файл Медведь.jpg.

Выберите *инструмент Штамп* .

Выберите размер штампа, но не меняйте других свойств.

Назначьте образец (ее начальную точку). Для этого прижмите клавишу *Ctrl* и щелкнув левой кнопкой мышки по части изображения, которое Вы возьмете за образец.

Прижмите левую кнопку мышки и водите по закрываемому образцом фрагменту (мотоциклисту). Меняйте образец почаще, добиваясь нужного результата.

Сохранить рисунок как Медведь_0.jpg.

7. Аналогично на изображения 4 Медведя.jpg вставьте еще одного медвежонка и сохраните рисунок под именем 5 Медведей.jpg.

Задание 5. Художественная обработка. Из изображений Лес_летом.jpg и Лес_осенью.jpg создайте изображение Лес_Лето_Осень.jpg

Алгоритм

Загрузить файлы Лес_летом.jpg и Лес_осенью.jpg и расположите их так, чтобы удобно было работать с обоими.

Размер изображения Лес_летом.jpg сделайте таким же, как у Лес_осенью.jpg

На изображении Лес_летом.jpg выполните команду *Правка* пункт *Копировать,* в результат которой выделенное скопируется в буфер обмена.

Перейдите на рисунок Лес_осенью.jpg и выполните команду *Правка* пункт *Вставить*. В результате будет создан плавающий слой.

В палитре *Слои* нажмите правой кнопку мыши на плавающем выделении и выберите команду *Создать слой.*

Выберите инструмент *Овальное выделение* и установите для него параметры: *Растушевать края, радиус – 50.* Выделите центр рисунка или правую половину. При необходимости измените размеры и/или место область выделения.

Выполните очистку выделения, нажав клавишу *Delete*.

При желании выполните регулировки каждого из слоев.

Сохраните рисунок как Лес.jpg

Задание 6. Художественная обработка. Из изображений Лес_осенью.jpg создайте рисунок Лес_Осень_1.jpg, Лес_Осень_2.jpg, Лес_Осень_3.jpg применив фильтры: *Фильтры/Имитация/Масляная краска*, *Фильтры/Имитация /Холст, Фильтры/Имитация /Рассеянный свет*. Можно поэкспериментировать разными фильтрами.

Задание 7. Фотоколлаж. Создать фотоколлаж из файлов: Лес летом, Лес осенью, Лес зимой, Лес весной, Времена года.

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1.Расшифруйте аббревиатуру GIMP.

2.Перечислите возможности редактора GIMP.

3.Перечислите основные компоненты диалогового окна GIMP.

4.Что из себя представляет окно изображения?

5.Перечислите основные компоненты панели инструментов.

Практическое занятие № 12 Работа с текстом

Цель занятия: познакомиться с векторным редактором Inkscape, научиться работать с основными инструментами и операциями над контурами.

Оборудование: ПК, векторный редактор Inkscape

Содержание и порядок выполнения задания

Изучите теоретический материал

Выполните задания.

Теоретический материал

Векторная графика

Векторное изображение - это графический объект, построенный из геометрических примитивов, таких как точки, линии, сплайны и многоугольники.

Рассмотрим, к примеру, такой графический примитив, как окружность радиуса r. Для её построения необходимо и достаточно следующих исходных данных:

координаты центра окружности;

значение радиуса r;

цвет заполнения (если окружность не прозрачная);

цвет и толщина контура (в случае наличия контура).

Преимущества

Размер, занимаемой описательной частью, не зависит от реальной величины объекта, что позволяет, используя минимальное количество информации, описать сколько угодно раз большой объект файлом минимального размера.

В связи с тем, что информация об объекте хранится в описательной форме, можно бесконечно увеличить графический примитив, например, дугу окружности, и она останется гладкой. С другой стороны, если кривая представлена в виде ломаной линии, увеличение покажет, что она на самом деле не кривая.

Параметры объектов хранятся и могут быть легко изменены. Также это означает что перемещение, масштабирование, вращение, заполнение и т. д. не ухудшат качества рисунка. Более того, обычно указывают размеры в аппаратно-независимых единицах (англ. device-independentunit), которые ведут к наилучшей возможной растеризации на растровых устройствах.

При увеличении или уменьшении объектов толщина линий может быть задана постоянной величиной, независимо от реального контура.

Недостатки

Не каждый объект может быть легко изображен в векторном виде — для подобного оригинальному изображению может потребоваться очень большое количество объектов и их сложности, что негативно влияет на количество памяти, занимаемой изображением, и на время для его отображения (отрисовки).

Перевод векторной графики в растр достаточно прост. Но обратного пути, как правило, нет — трассировка растра, при том что требует значительных вычислительных мощностей и времени, не всегда обеспечивает высокое качества векторного рисунка.

Форматы векторной графики: .cdr, .ai, .cmx, .eps, .fla, .svg, .swf, .wmf.

К программным средствам создания и обработки векторной графики относятся следующие ГР: CorelDraw, AdobeIllustrator, а также векторизаторы (трассировщики) - специализированные пакеты преобразования растровых изображений в векторные.

Как в растровой, так и в векторной графике необходим способ кодирования цвета.

Графический редактор Inkscape обладает достаточным набором инструментов для создания иллюстраций и довольно удобным интерфейсом. Он представляет собой программу ориентированую на решение конкретной задачи – создание иллюстративной графики.

Иллюстративная графика – это прикладная ветвь машинной график, сравнительно недавно выделившаяся в отдельное направление наряду с графикой деловой, инженерной и научной. К области иллюстративной графики относятся в первую очередь рисунки, коллажи, рекламные объявления, заставки, постеры – всё, что принято называть художественной продукцией. Объекты иллюстративной графики отличаются от объектов других прикладных областей своей первичностью – они не могут быть построены автоматически по некоторым исходным данным, без участия художника или дизайнера.

Основные принципы работы

Перед началом работы с Inkscape нужны общие представления о возможностях этой программы. Основным понятием в любом редакторе векторной графики, является понятие объекта. Работа над любой иллюстрацией заключается в создании объектов, их редактировании и расположении в нужных местах. При этом сначала создается приблизительная форма объектов, после чего форма уточняется путем добавления, удаления и перемещения узлов контура. После создания необходимой формы задается цвет контура и выбирается заливка. В этом редакторе можно создавать как стандартные фигуры (прямоугольники, эллипсы, многоугольники, и т.д.), так и произвольные, состоящие из прямых и кривых линий.

Средства работы с текстом позволяют создавать небольшие текстовые документы, оформленные рисунками.

Применение различных эффектов помогает создать красивое изображение из простых объектов. Каждый рисунок состоит из одной или нескольких фигур, которые могут накладываться и полностью или частично закрывать друг друга.

Итак, основные приемы работы с Inkscape:

Создание простых геометрических фигур или произвольных кривых и ломаных, замкнутых или разомкнутых;

Редактирование любого объекта: изменение цвета контура и заливки, изменение формы объекта;

Размещение всех объектов в нужных местах, определение порядка взаимного перекрытия объектов;

Вставка и форматирование текста;

Вставка готовых картинок или ранее созданных иллюстраций в документ.

Понятие объекта в Inkscape

Любое изображение в векторном формате состоит из множества составляющих частей, которые редактируются независимо друг от друга. Главными «кирпичиками», из которых составляется изображение, являются объекты. Объектом называется элемент изображения: прямая, круг, прямоугольник, кривая, замкнутая кривая, многоугольник и другие.

Любой векторный объект Inkscape имеет ряд общих характеристик. Каждый объект состоит из некоторого количества точек или узлов, соединенных прямыми или кривыми линиями — сегментами. Координаты узлов и параметры сегментов определяют внешний вид объекта. Сегменты объекта образуют контур, который имеет свой цвет и толщину. Область внутри объекта можно закрасить или залить одним цветом, смесью цветов или узором. Эту область принято называть заливкой. У одного объекта не может быть различных заливок или соединительных линий разной толщины и цвета. Для создания сложных изображений требуется использовать множество объектов.

Важными объектами Inkscape являются плавно изогнутые кривые, с помощью которых можно построить любой произвольный контур. Они называются кривыми Безье. Математик Пьер Безье (Pierre Bezier) открыл, что произвольную кривую можно задать с помощью двух векторов, находящихся в начале и конце кривой. Это положение легло в основу описания кривых Безье в Inkscape. Кроме положения начальной и конечной точки (то есть узлов кривой), внешний вид кривой определяется кривизной, то есть ее изогнутостью между двумя узлами. Кривизна определяется двумя параметрами кривой в каждом узле, которые графически представлены с помощью отрезков, выходящих из узлов. Эти отрезки называются манипуляторами кривизны.

Задание 1: создать изображение солнца

Ход работы:

1) Выбираем на панели инструментов элемент «Рисовать круги, эллипсы, дуги» или F5. Нажимаем клавишу Ctrl и рисуем окружность.

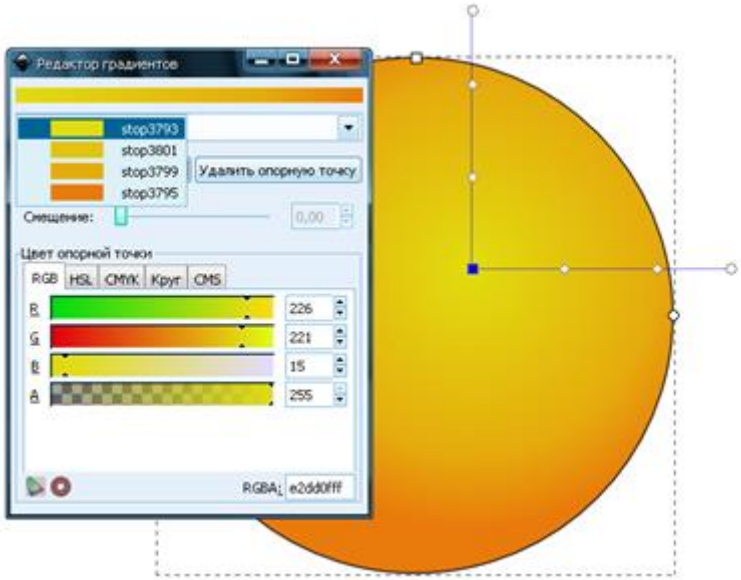
2) Заливаем окружность. Открываем «Заливка и обводка» *(Объект – Заливка и обводка)*. Выбираем способ заливки Радиальный градиент. Теперь там же нажимаем кнопку «Изменить» и переходим к редактированию градиента.

Установим значения (R, G, B, A): 236, 221, 15, 255

Выберите 2 точку градиента и установите значения: (R, G, B, A): 233, 123, 12, 255

Снова выберите 1 точку градиента и добавьте еще две опорных точки

Передвинем точки равномерно и сместим центр градиента



3) Добавляем солнцу глаза. Нарисуйте окружность и выберите градиентную радиальную заливку

Установим значения (R, G, B, A): 255, 255, 255, 255

Выберите 2 точку градиента и установите значения: (R, G, B, A): 225, 225, 225, 255

Снова выберите 1 точку градиента и добавьте опорную точку и сдвиньте ее вправо

4) Нарисуйте еще одну окружность меньшего диаметра.

Установите параметры: (R, G, B, A): 170, 223, 255, 255

Самостоятельно подберите значения для второй части градиента.

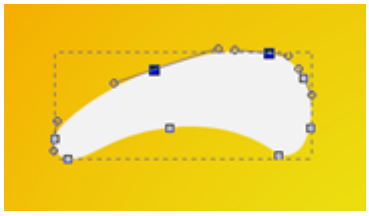
Добавьте опорную точку и сместите ее вправо.

5) Теперь убираем обводку и дорисовываем зрачок и блики.

6) Сделаем наш глаз более объемным. Продублируем *(Правка – Продублировать объект)* нижнюю белую окружность два раза. Делаем сплошную заливку светлее для одной окружности и темнее для другой, перемещаем дубликаты под окружность (PgDown) и создаем тень над и под глазом .

7) После того как мы нарисовали глаз, сгруппируем объекты. Выбираем элемент «Выделять и трансформировать объекты», выделяем все элементы глаза и группируем их *(Объект - Сгруппировать)*, затем дублируем группу и перемещаем дубликат на место второго глаза, затем чуть-чуть уменьшаем второй глаз.

8) Теперь изобразим бровь. Активизируем инструмент «Рисовать кривые Безье и прямые линии» и рисуем кривую из отрезков. Выберем инструмент «Редактировать элементы узлов и рычаги» и задаем для каждого узла автоматическое сглаживание.



9) Теперь зальем бровь выбрав «Линейный градиент» и изменив направление инструментом «Редактировать элементы узлов и рычаги»

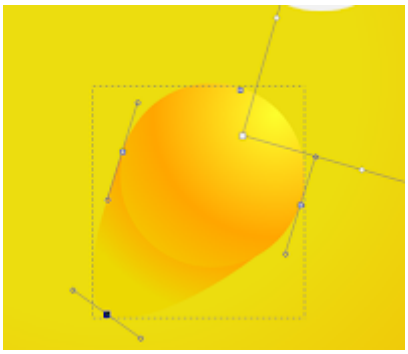
10) Теперь создадим объем для брови точно так же как для глаза, но сделаем только один дубликат и опустим чуть ниже брови.

11) Группируем объекты брови, дублируем группу, выбираем *(Объект – Отразить горизонтально)* чтобы развернуть дубликат и помещаем вторую бровь на место.

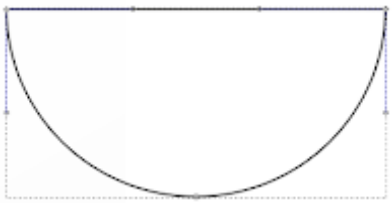
12) Теперь дорисуем солнцу нос. Рисуем окружность, затем делаем контур из окружности *(Контур - Оконтурить объект)*.

13) Выбираем градиентную радиальную заливку и выставляем параметры. Добавляем опорные точки и смещаем центр градиента

14) Дублируем нос, снимаем обводку и помещаем дубликат под оригинал, активизируем инструмент «Редактировать элементы узлов и рычаги» и вытягиваем один узел, чтобы создать тень под носом.

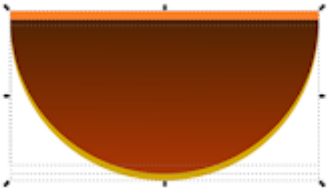


15) Теперь изобразим рот. Создаем окружность, активизируем инструмент «Редактировать элементы узлов и рычаги» и тянем узел пока не сделаем из круга полукруг.

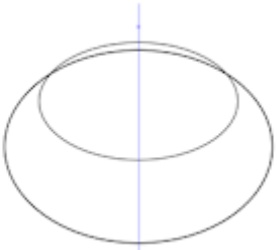


16) Переводим получившийся полукруг в контур, и заливаем линейным градиентом.

17) Дублируем объект, помещаем дубликаты ниже оригинала, заливаем соответствующими цветами и смещаем, чтобы получить подобную картину.

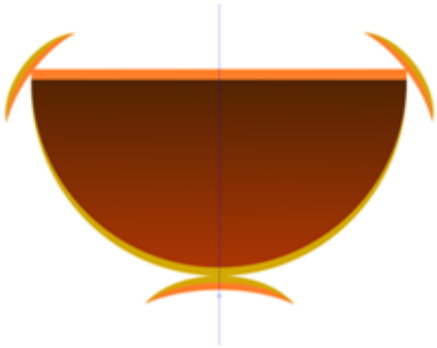


18) Теперь для выразительности персонажа дорисуем элементы придающие визуальный объем на подбородке и в уголках рта. Рисуем два эллипса, один поверх другого.



19) Выбираем эллипсы и оконтуриваем объект, затем делаем *(Контур – Разность)*, чтобы вырезать сегмент верхним объектом из нижнего.

20) Дублируем полученный объект, уменьшаем по вертикали, заливаем объекты цветом и группируем, чтобы получить вот такую картину.

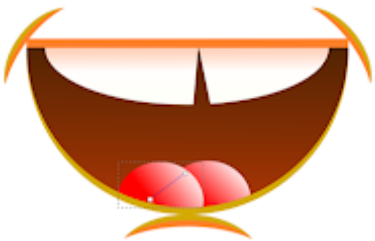


21) Теперь изобразим зубы. Создаем полукруглый сегмент, переводим его в контуры и заливаем градиентом.

22) Добавляем прикольную щербинку к зубам, инструментом «Рисовать звезды и многоугольники» рисуем поверх зубов треугольный сегмент, выделяем зубы и этот сегмент и делаем *(Контур - Разность)*.

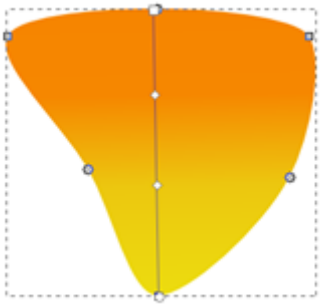
23) Теперь нарисуем язык из двух эллипсов. Дублируем контур рта и поочередно каждого из двух эллипсов, выделяем и делаем *(Контур – Пересечение)*

24) Ставим полученные сегменты языка на место и заливаем их линейным градиентом как на картинке.



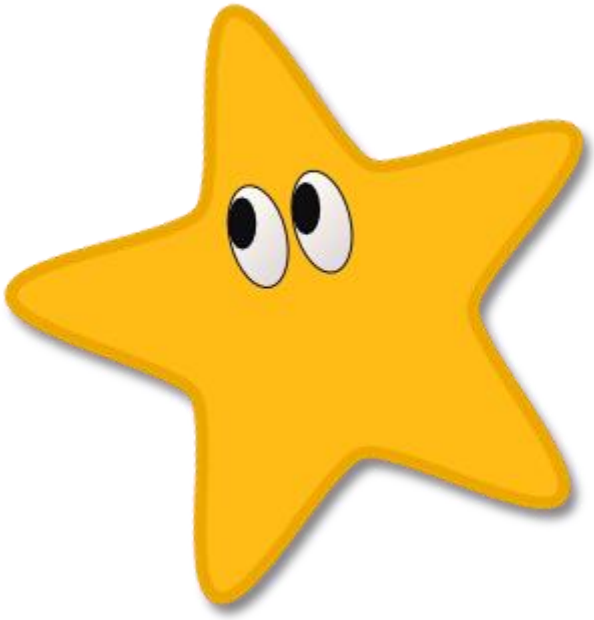
25) Группируем все объекты рта и помещаем рот на место

26) Теперь нарисуем декоративную солнечную корону. Рисуем один скругленный треугольный луч, заливаем его градиентом как на картинке, затем дублируем этот луч и немножко изгибаем его.

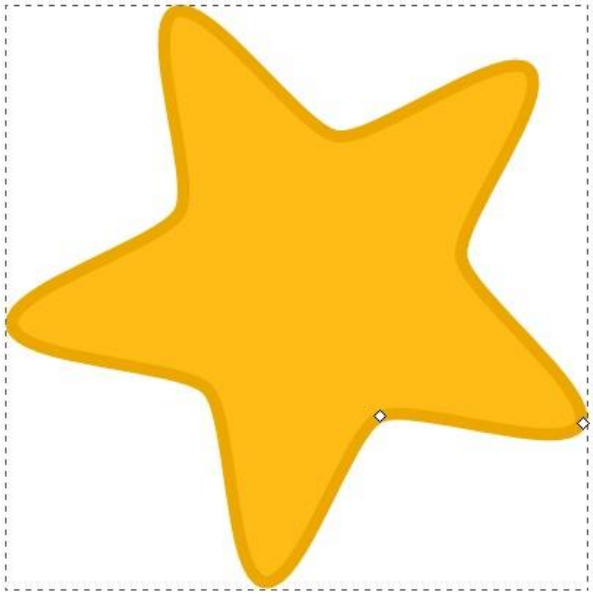


27) Дублируя лучи, располагаем их вокруг солнца.

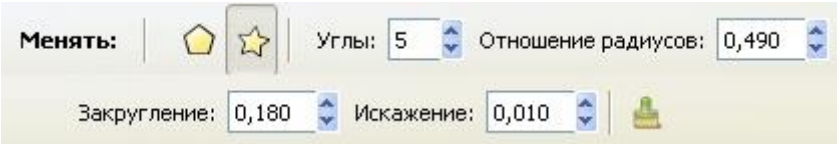
Задание 2 ЗВЕЗДОЧКА С ГЛАЗАМИ



1 Возьмите инструмент для рисования звезд и просто создадим с его помощью вот такую пятиугольную звезду.

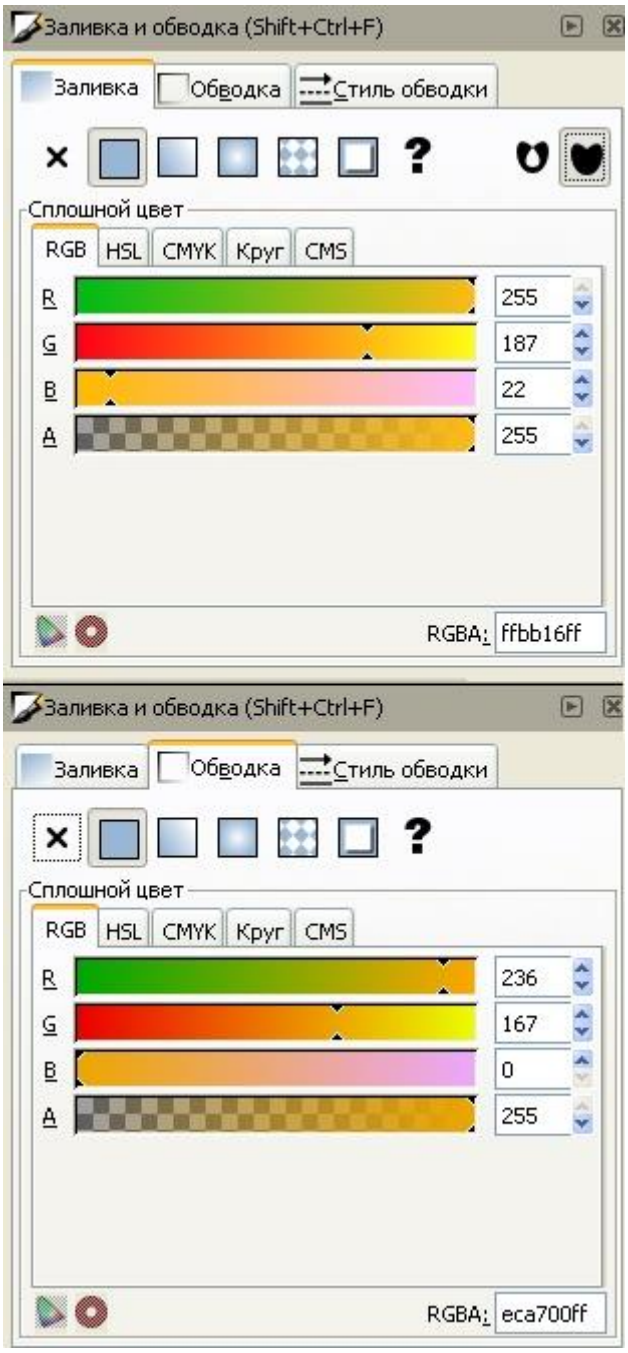


Ниже приведен скриншот контекстной панели инструмента звезды. Параметры, указанные на этом скриншоте, как раз отвечают за скругление углов, количество лучей и т.д. Подробнее можно прочитать в разделе инструкция с описанием этого инструмента.

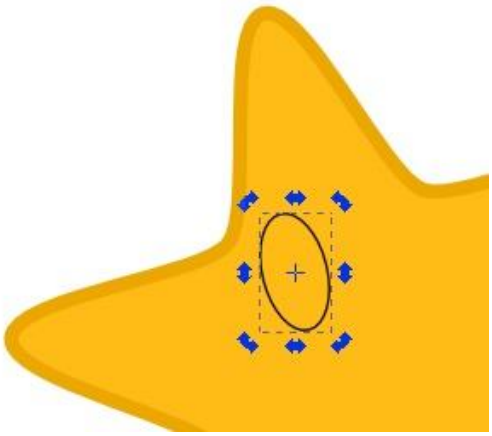


На двух следующих рисунках приведены параметры для цвета заливки звезды и параметры для цвета обводки. Это окошко открывается по комбинации клавиш Ctrl+Shift+F и будет активным для редактирования, если нарисованная вами

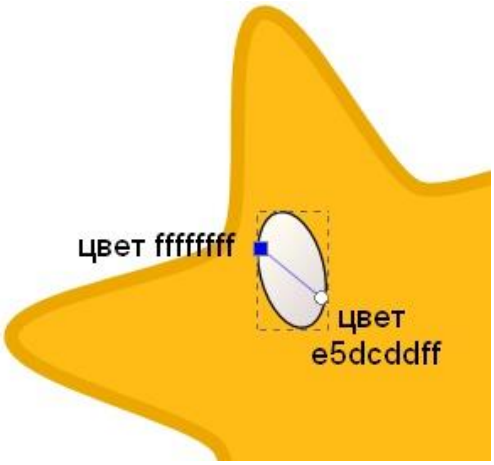
звездочка выделена, т.е. сама активна. Толщину обводки можно поменять на третьей закладке.



2 Теперь нарисуем глазик. Для этого нам понадобится инструмент эллипс. На скриншоте ниже как раз показан нарисованный нами эллипс для будущего глаза звезды. Этот эллипс пока без заливки, но уже с нужной толщиной обводки в 1 пиксель черного цвета. Поворачивать, перемещать и изменять размер эллипса можно с помощью инструмента инструмента выделения и трансформации. Подробнее можно прочитать в разделе инструкция с описанием этого инструмента.



Сделаем заливку для эллипса глаза в виде линейного градиента. Расположение направляющей градиента показано на рисунке. Если направляющая у вас сразу не видна, после того как на закладке заливки вы выбрали тип линейный градиент, то активируйте инструмент градиент в боковом окне инструментов и все появится. Если щелкнуть инструментом градиент на крайние точки направляющей, то можно задать их цвета. Цвета точек направляющей градиента показаны на рисунке ниже.



Теперь, когда белок глаза готов, нарисуем еще один овал, который будет абсолютно черным. Сделать черную заливку и обводку не должно составить у вас труда. Расположите второй овал-зрачок так, как вам нравится, и смотря какую эмоцию звездочке вы хотите придать. Как сделали мы, видно на рисунке ниже. Теперь сгруппируйте оба овала, что бы они стали одним целым. Для этого выделите их оба и нажмите комбинацию клавиш Ctrl+G.

3 Теперь можно легко сделать второй глазик. Для этого надо сделать копию первого. Т.е. продублировать его. Продублировать объект можно по комбинации клавиш Ctrl+D. При этом вы не заметите визуальной разницы, т.к. копия объекта располагается прямо поверх копируемого. Но теперь вы можете сдвинуть ее мышью или стрелочками клавиатуры и увидите, как под ней будет появляться точно такой же объект. Расположите правильно второй глазик звезды. Рисунок готов.

Можно добавить звезде тень. Сделать это можно в меню "Фильтры" - "Свет и тень" - "Отбрасывать тень".

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Вариант задания
4. Отчет о выполнении на каждый этап раздела «Содержание и порядок выполнения задания»
5. Список используемых источников
6. Выводы и предложения
7. Дата и подпись курсанта и преподавателя

Вопросы для самопроверки:

1. Для чего предназначены векторные графические редакторы?

2.В чем заключаются основные отличия векторных изображений от растровых?

3.В каких сферах деятельности векторные изображения нашли наиболее широкое применение?

4.Что является элементарным объектом векторной графики?

Практическое занятие №13 Представление о моделировании в среде графических редакторов Моделирование геометрических фигур растровой графики

Цель занятия:

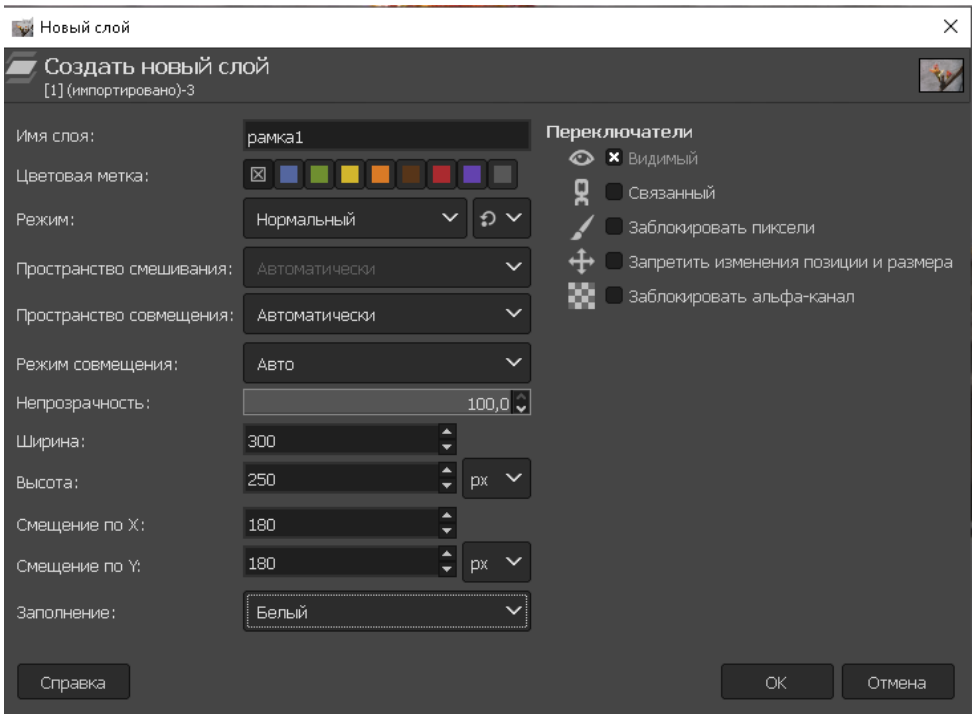
- Научиться основам работы со слоями

Исходные материалы и данные: ПК, Программа GIMP.

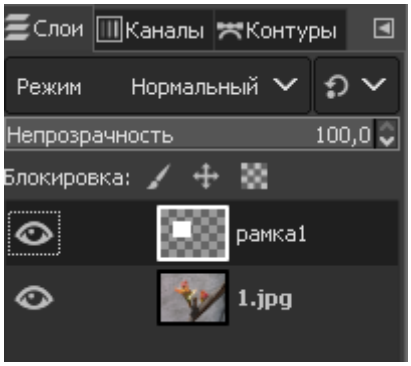
Содержание и порядок выполнения задания:

Открываем изображение, над которым будем работать. Лучше чтобы его размер был побольше, мы взяли фотографию виноградного ростка из своего фотоальбома предварительно доведя ее до разрешения 1024x768 пикселей.

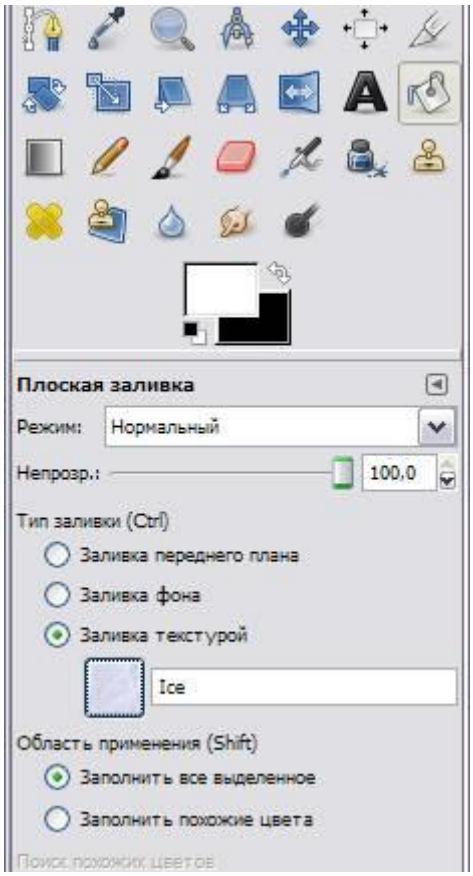
Теперь нужно создать еще один слой размером 300x250 пикселей с белым или еще лучше - прозрачным фоном. Меню Слой/Создать слой. Мы назовем его Рамка1.



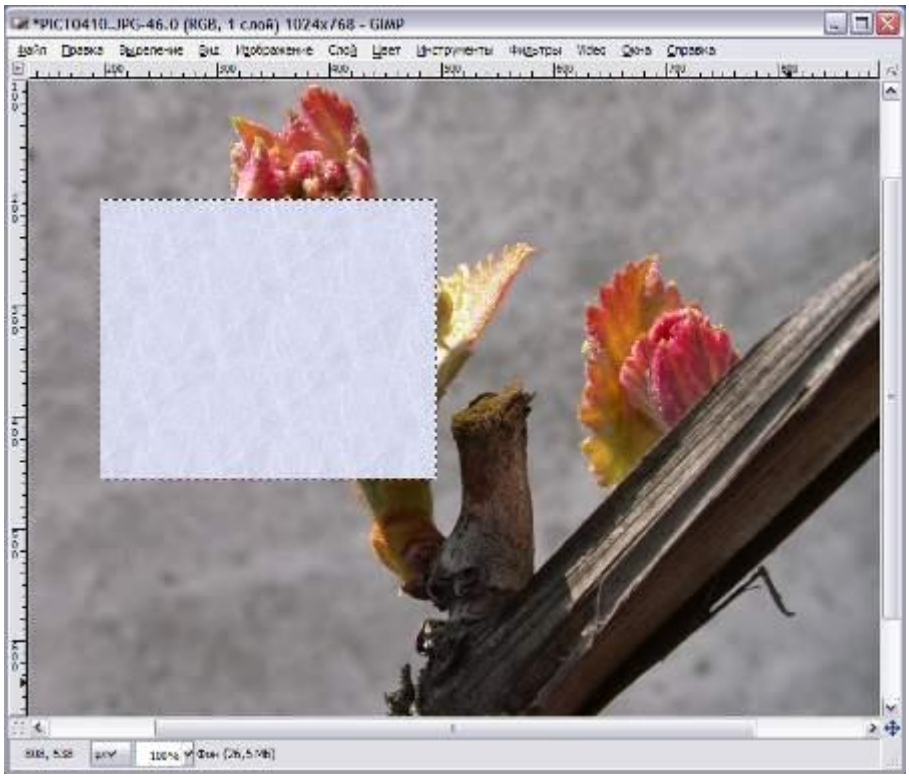
Откройте меню Окно/Прикрепляющие диалоги/Слои.



Выберем Плоскую заливку  (Shift+B) и зальем этот слой каким-нибудь цветом или текстурой. Тип заливки будет определять, как будут выглядеть рамки фотографий, поэтому цвет выбирайте на свое усмотрение. Мы же выбрали Заливку текстурой и текстуру Ice из стандартного набора.

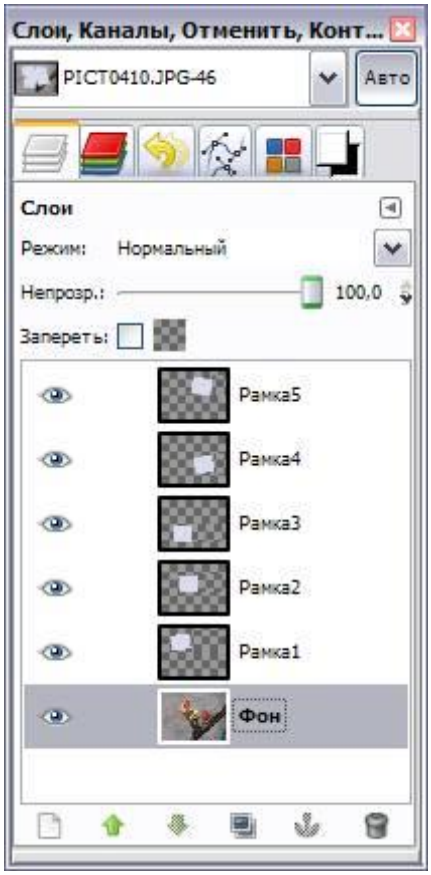


Переместим слой с верхнего левого угла, где он оказался По-умолчанию при создании, в какое-нибудь более достойное место и займемся клонированием фоторамок.

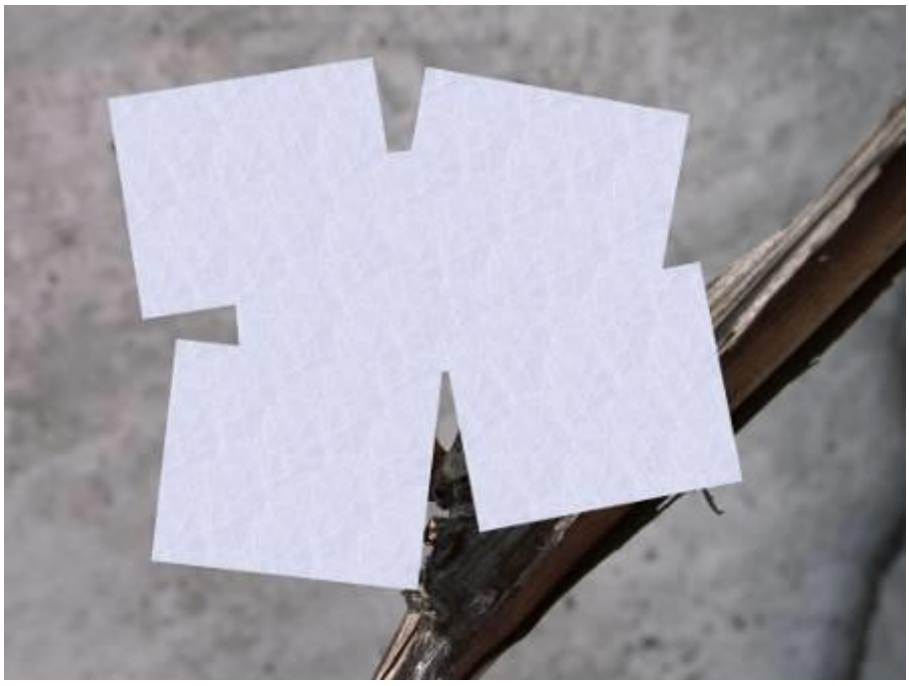


Создадим несколько дубликатов слоя Рамка1 и назовем их соответственно Рамка2 и т.д. Порядок слоев пока значения не имеет. Мы создали всего 5 слоев с рамками. (Чтобы скопировать слой достаточно нажать кнопочку под списком слоев или клавиши Shift+Ctrl+D). Все слои с рамками у нас оказались один под другим, поэтому мы выбираем

инструмент Перемещение (M) и разместим наши рамки по всему изображению. Сразу бросается в глаза, что рамки лежат слишком уж ровно, поэтому мы приведем их в творческий беспорядок инструментом Вращение (Shift+R).

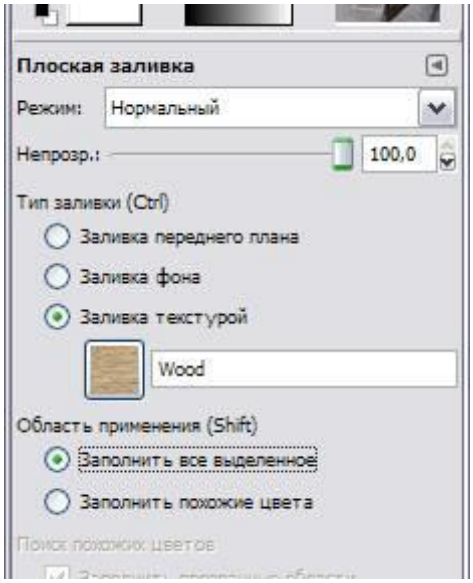


Если не хотите каждый раз при вращении нажимать кнопку Повернуть в выскакивающем диалоге вращения, то при вращении удерживайте клавишу Shift. Когда вы отпустите кнопку мыши, изображение повернется сразу же. При этом слой будет вращаться относительно своего центра на тот угол, который вы задали мышкой. В итоге должно получиться примерно вот так:



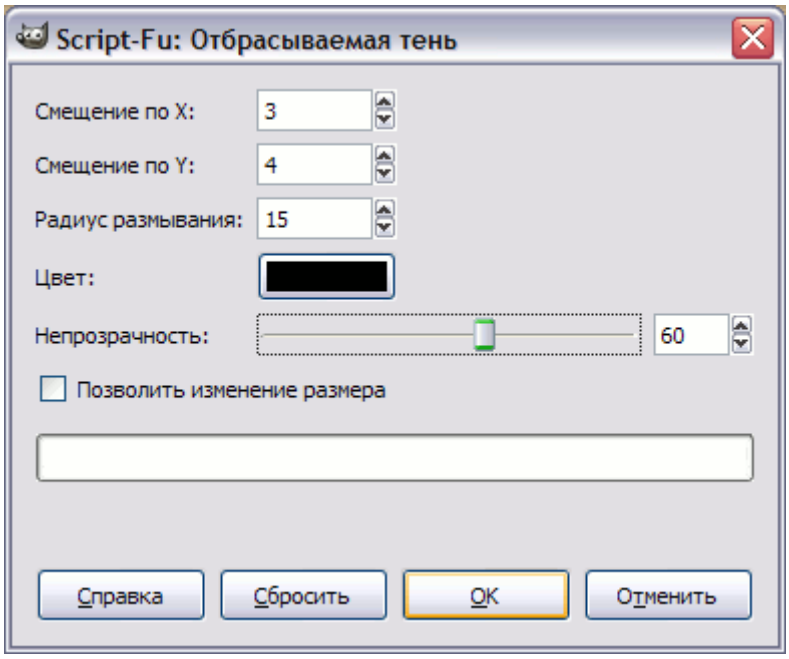
Теперь начинаем самое интересное! Щелкните правой кнопкой мыши на одном из слоев Рамка и в контекстном меню выберите Альфа-канал в выделение (то же самое доступно в меню Слой - Прозрачность). Вокруг выбранной рамки на изображении появится выделение. Теперь уменьшите его на 10 пикселей через меню Выделение - Уменьшить. Контур выделения должен стать чуть меньше. Смело нажимайте клавишу Delete, тем самым удалив внутреннюю часть рамки. Не снимая выделения, переходим на слой с фотографией, у нас это слой Фон, и копируем выделенную часть изображения Ctrl+C (Правка - Копировать). Снова возвращаемся на тот же слой с подготовленной рамкой и жмем Ctrl+V (Правка - Вставить). Нажмите меню Слой/Прикрепить слой. Проделав те же манипуляции с остальными рамками, получим следующее:

Теперь можно отключить видимость слоя с фотографией, нажав слева от названия слоя или вообще удалить этот слой. А вместо него создадим новый слой на который вы сможете поместить какую-нибудь подложку, например изображение стола, на котором должны лежать наши фотографии. Например залив новый слой текстурой Wood инструментом Плоская заливка (Shift+B).



Вот какая у нас получилась подложка:

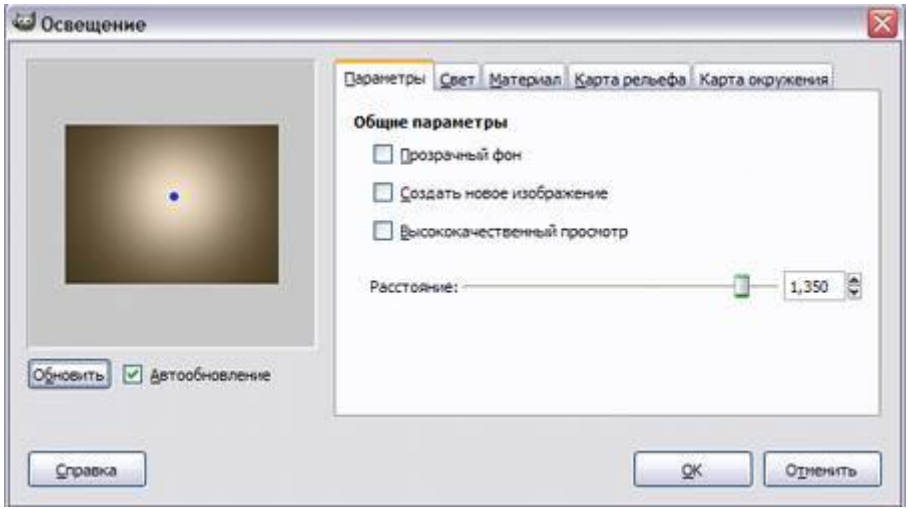
Для пущей реалистичности добавим к нашим фотокарточкам тень в меню Фильтры - Свет и тень - Отбрасываемая тень.

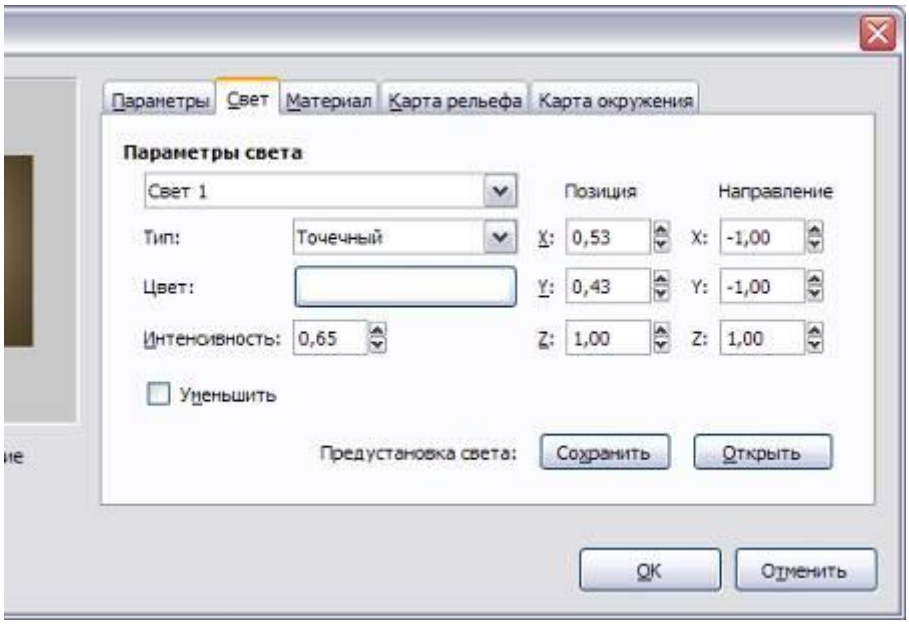


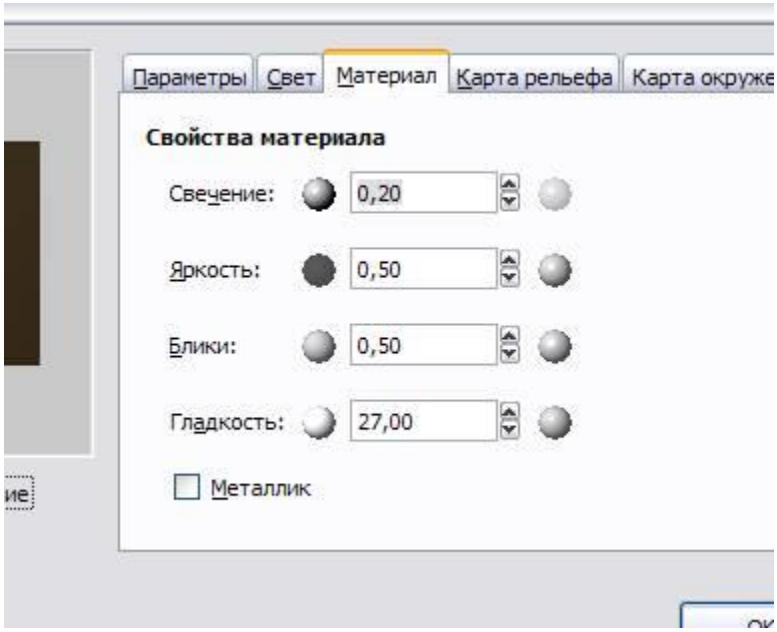
Должно получиться где-то так:

Теперь ту же операцию по созданию тени нужно проделать с остальными рамками. Это очень удобно можно сделать через меню Фильтры - Повторить "Отбрасываемая тень" или еще проще - Ctrl+F. Что может быть проще - выбрали слой с рамкой, нажали Ctrl+F, и тень появилась с теми же параметрами.

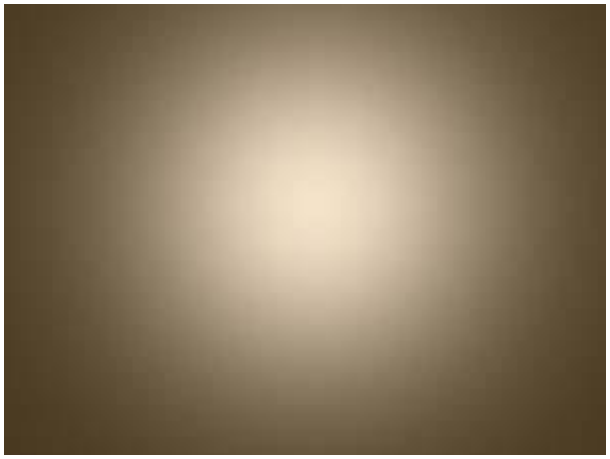
Для большей реалистичности можно добавить эффект освещения. Мы для этого создали еще один слой поверх всех слоев и залили его цветом #5d492b. Далее мы к этому слою применили фильтр Освещение, который находится в меню Фильтры - Свет и тень. Вот настройки, которые мы использовали:







Последние две вкладки Карта рельефа и Карта окружения мы не использовали. Получили вот что:



Теперь для этого слоя мы поставили Режим смешивания - Объединение зерна и непрозрачность - 35%.

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Результат работы сохранить файлом в своей папке
4. Список используемых источников

Практическое занятие №14 Просмотр и разрешение изображения. Выделение областей. Инструменты выделения

Цель занятия:

- Научиться технике «живописи» в графическом редакторе GIMP. Нарисовать традиционный новогодний сюжет: еловую ветку, украшенную ярким шаром

Исходные материалы и данные: ПК, Программа GIMP.

Содержание и порядок выполнения задания:

Задание 1

Шаг 1

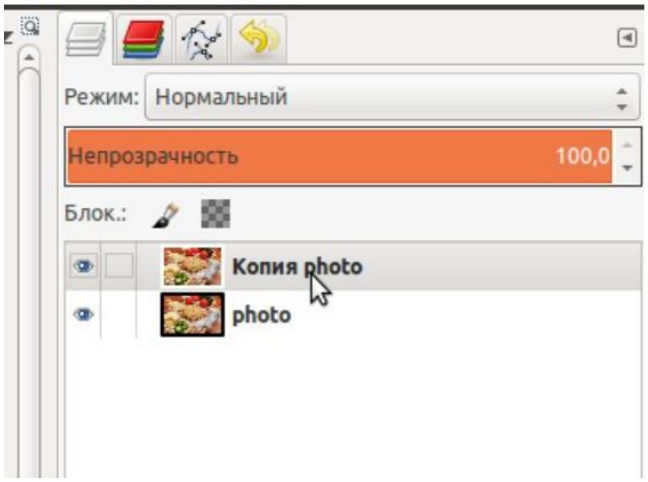
Открываем исходное фото в редакторе

Шаг 2

На следующем шаге нужно выделить основной объект на фото, который мы хотим сделать резким. Для этого самым простым, но и в то же время универсальным методом будет использование инструмента «Свободное выделение» или так называемое «Лассо» (как в фотошопе). После этого аккуратно обводим объект. Чем больше контрольных точек вы поставите, тем лучше.

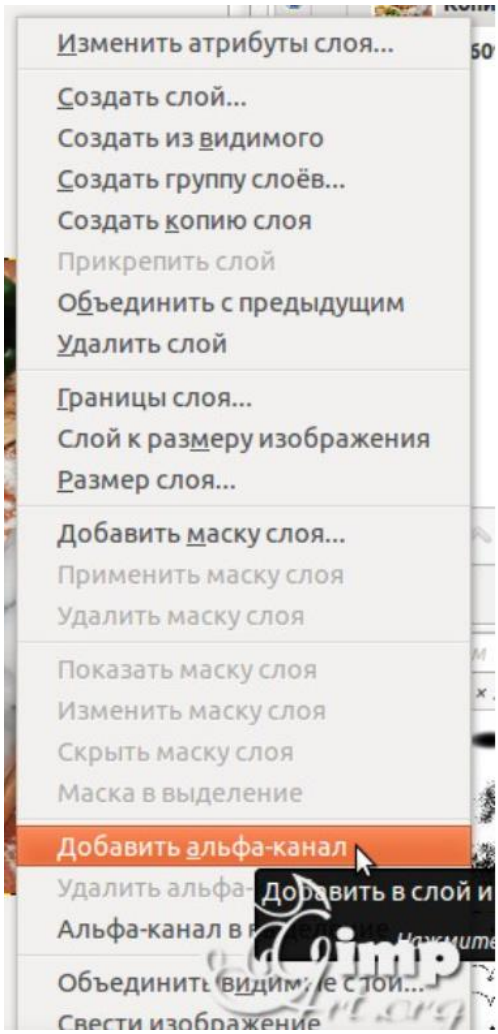
Шаг 3

Пока активно выделение создаем копию исходного фото через меню «Слой – Создать копию» или нажав на пиктограмму «Создать копию слоя»



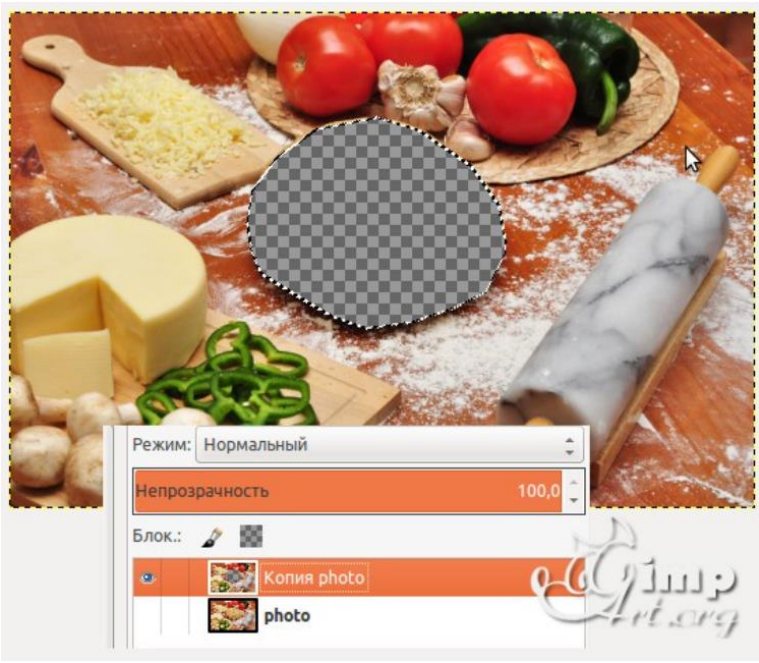
Шаг 4

Теперь нужно щелкнуть правой кнопкой мыши по верхнему слою и из открывшего контекстового меню выбрать пункт «Добавить альфа-канал».



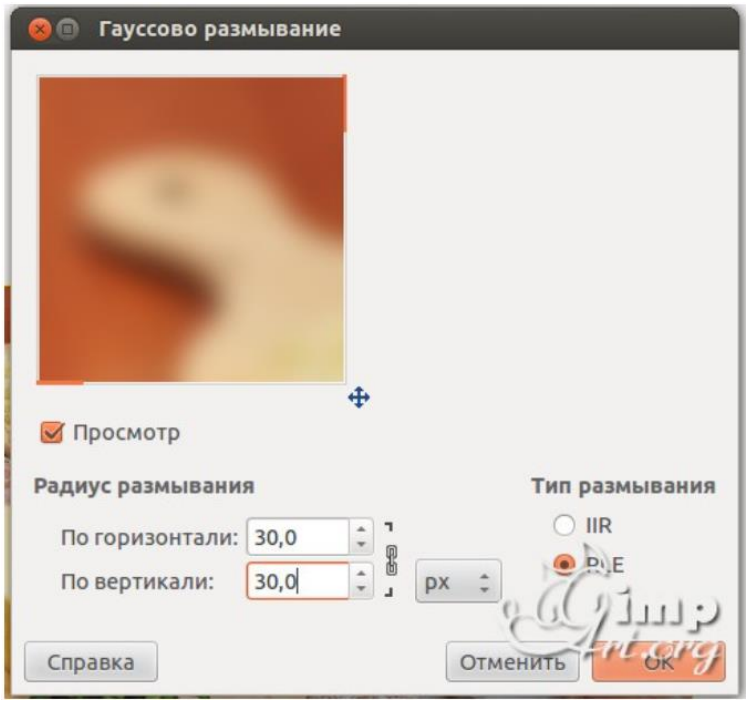
После этого нажимаем кнопку Del. Посмотрите, что должно у вас получиться если временно отключить видимость нижнего слоя.

Снова включаем видимость нижнего слоя и снимаем выделение через «Выделение -Снять»



Шаг 5

Теперь воспользуемся стандартным фильтром редактора размытия через меню «Фильтры – Размывание – Гауссово размывание» и в настройках выставляем нужное значение.



Справка: Данное значение зависит от исходной фотографии. Чем больше разрешение (размер фотки) тем больше вводимый параметр. Для данного примера я взял размер 30px

Шаг 6

По необходимости можно отрегулировать непрозрачность верхнего слоя с эффектом, для этого сдвиньте ползунок непрозрачности влево до получения желаемого эффекта. Например, я поставил значение 80.

Готово

Задание 2 (по желанию)

Создайте новый файл. В поле Шаблон укажите размер - 640x480 точек. Разрешение изображение 300 dpi (точек на дюйм).

Оформление фона

Для оформления фона еловой ветки, украшенной шаром, используйте темные тона.

Выполните градиентную заливку фона в темно-синей гамме:

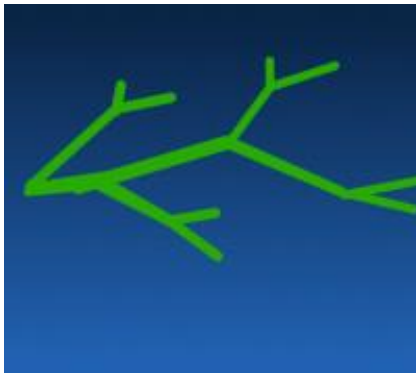
1. Выберите цвета переднего плана и фона: темно-синий и сине-голубой соответственно.
2. Выберите инструмент Градиент.
3. На панели инструментов установите тип градиента: Основной в фоновый (RGB).
4. Проведите вертикальную линию от верхнего края рисунка к нижнему. Градиентная заливка заполнила

все рабочее поле.

Еловая ветка

Нарисуйте еловую веточку:

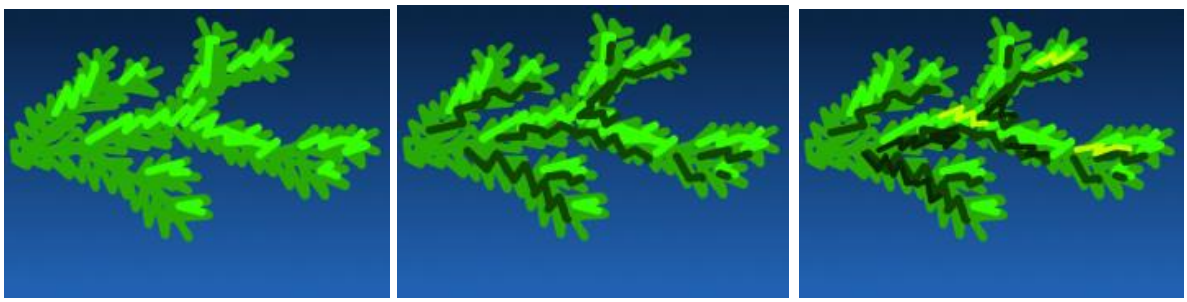
1. Создайте новый слой с помощью кнопки на панели слоев.



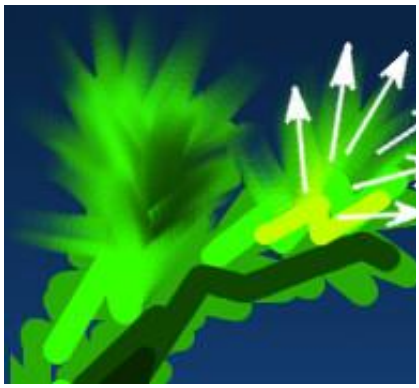
2. Кистью обозначьте основные линии ветки.

3. Нарисуйте «иголки» на ветках. Удобнее всего рисовать кистью, выполняя щелчки мыши при нажатой клавише Shift.

4. Чтобы работа была действительно живописной, добавьте разнообразия красок: светлые и темные цвета, более теплые и более холодные по тону.



5. С помощью инструмента Палец размажьте краски. Внимание: направление воздействия инструмента совпадает с направлением роста иголок на ветке.



6. Оцените полученную композицию. Возможно, есть необходимость переместить, масштабировать или повернуть ветку, чтобы освободить место для елочного украшения. В таком случае, выполните эти действия с помощью специальных инструментов: Перемещение, Вращение и Масштаб.

Елочный шар

Нарисуете яркий елочный шар:

1. Создайте новый слой.
2. Выделите круг и выполните заливку ярким цветом теплой гаммы (желтый-оранжевый-красный).

3. Добавьте блики и тени к изображению шара...

4. ...а также мелкие детали.

5. С помощью инструмента Палец размажьте цветовые пятна круговыми движениями.

Если вы чувствуете, что на шаре не хватает какого-либо цвета, вы можете дорисовать кистью недостающее пятно и размазать его, смешав с окружающими красками.

Попробуйте также оформить фон в «живописной» технике. Вы можете использовать Кисть разной формы, а затем размазывать и смешивать цветовые пятна с помощью инструмента Палец.

Сохраните работу в формате xcf.

Содержание отчета:

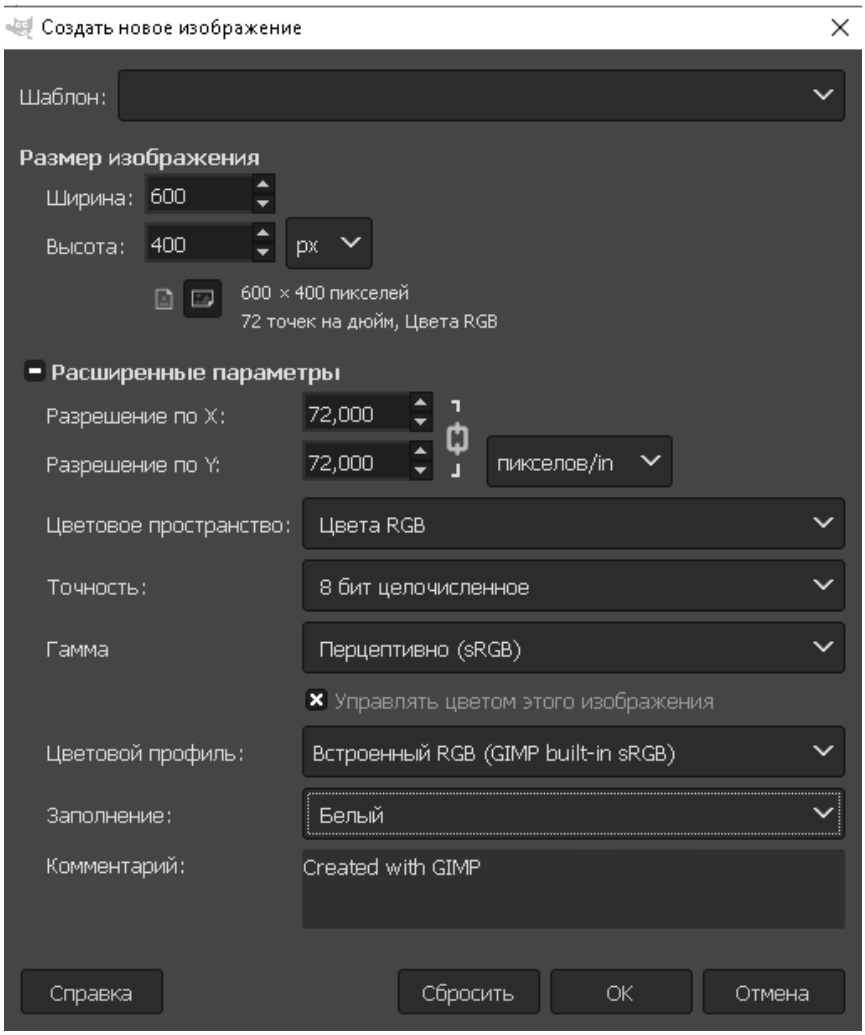
1. Наименование практического занятия
2. Цель занятия

3. Результат работы сохранить файлом в своей папке
4. Список используемых источников

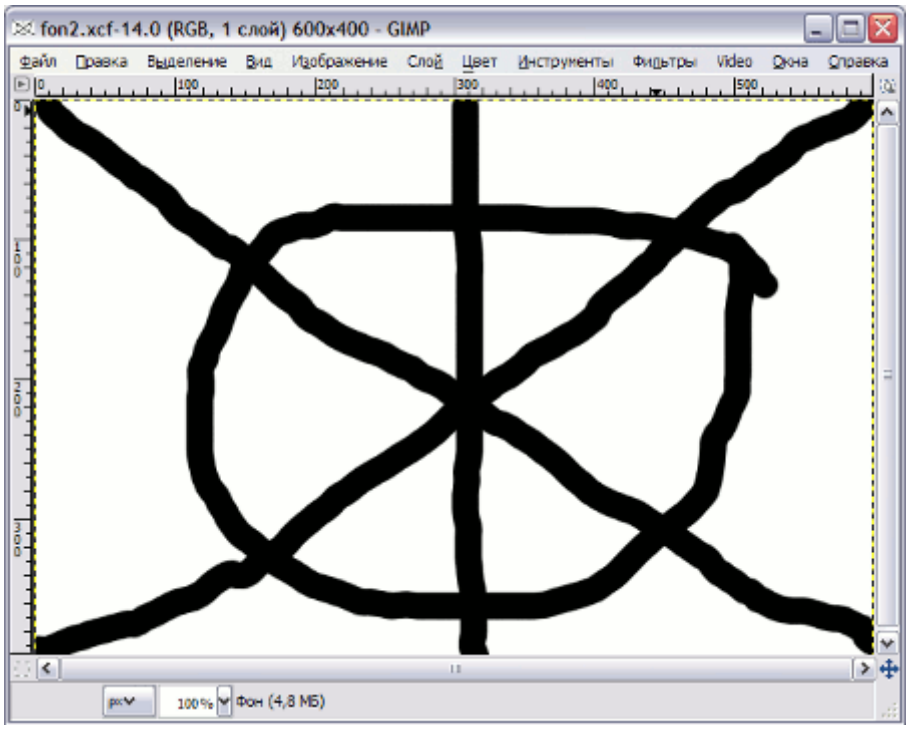
Практическое занятие №15 Основы работы со слоями

Задание 1 Откройте программу Gimp.

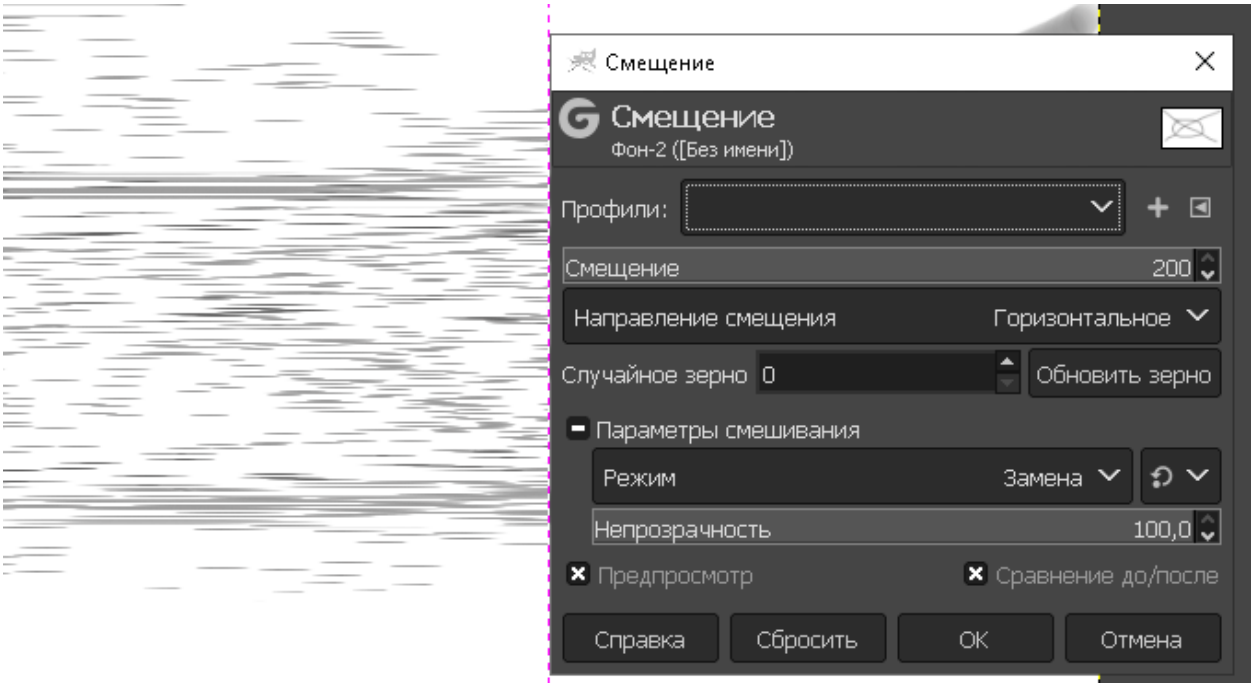
Далее создаем новый документ размером 600х400 пикселей и белым фоном.



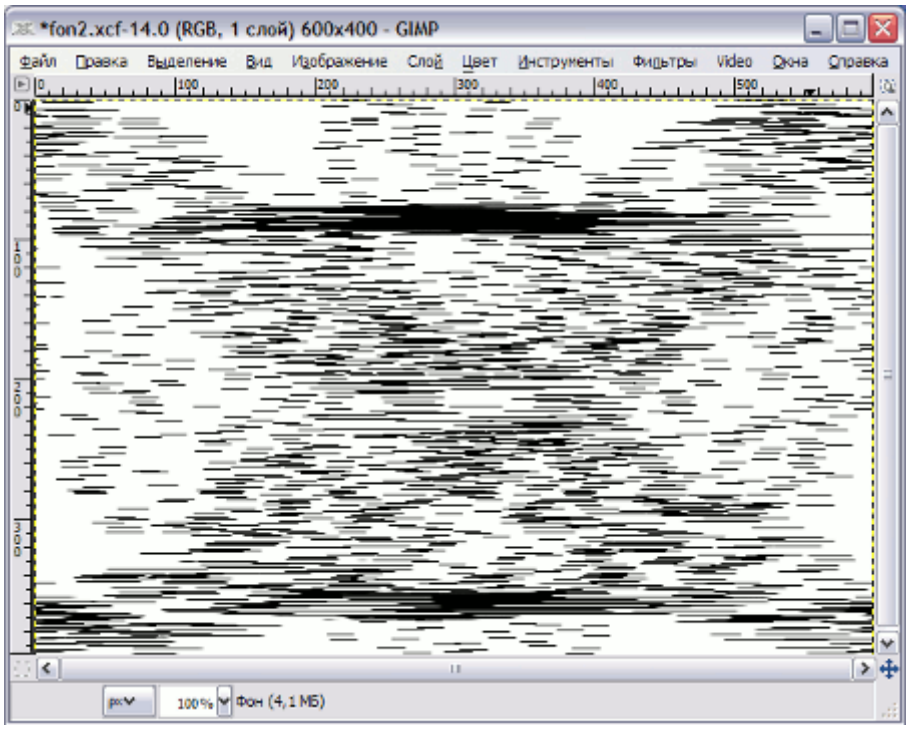
Выбираем жесткую круглую кисть размером 19х19 и рисуем черным цветом кляксу со следующего изображения.



Идем в меню Фильтры и выбираем: Искажения - Смещение и в параметрах устанавливаем значение 200px.

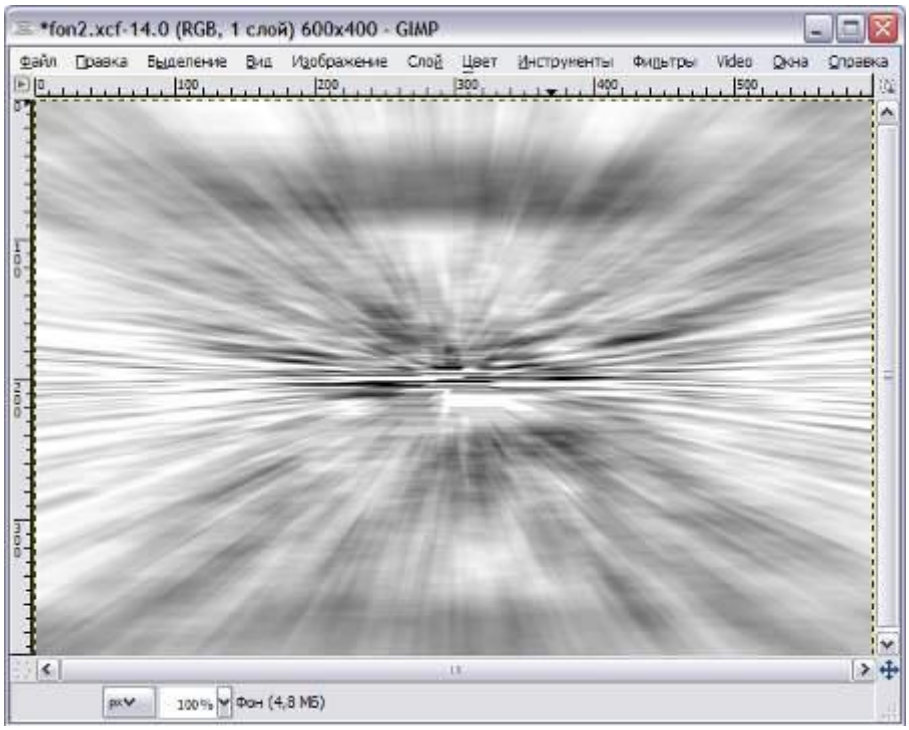


После чего у нас должна получиться примерно такая картина:

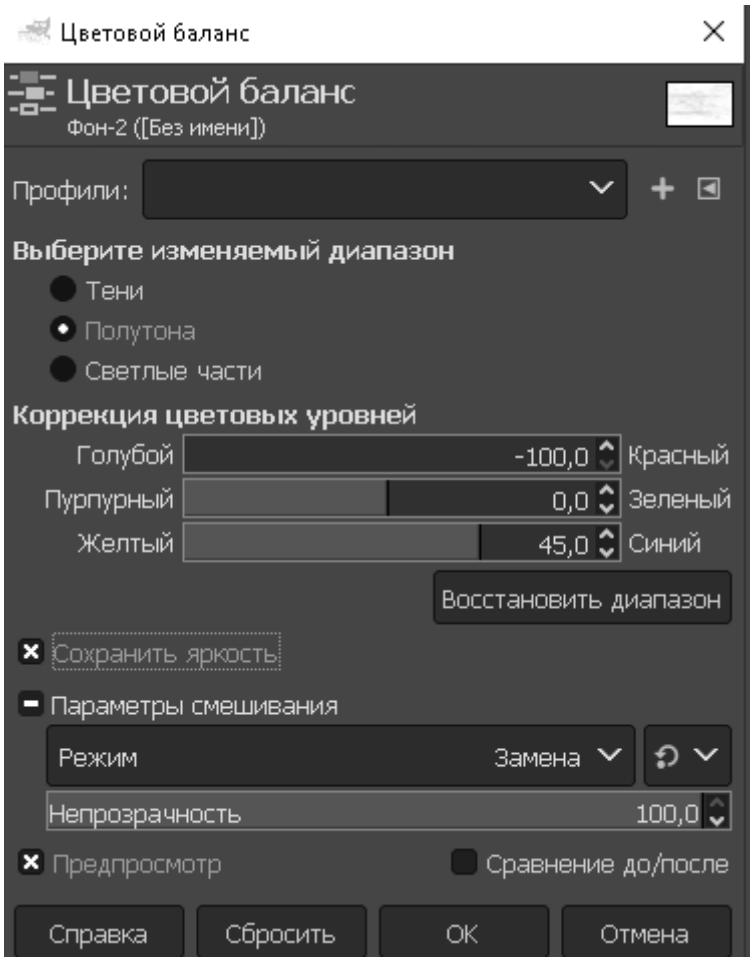


Далее будем применять фильтр Размывание движением. Находится он в меню Фильтры - Размывание. В настройках фильтра можно установить приблизительно следующие параметры размывания:

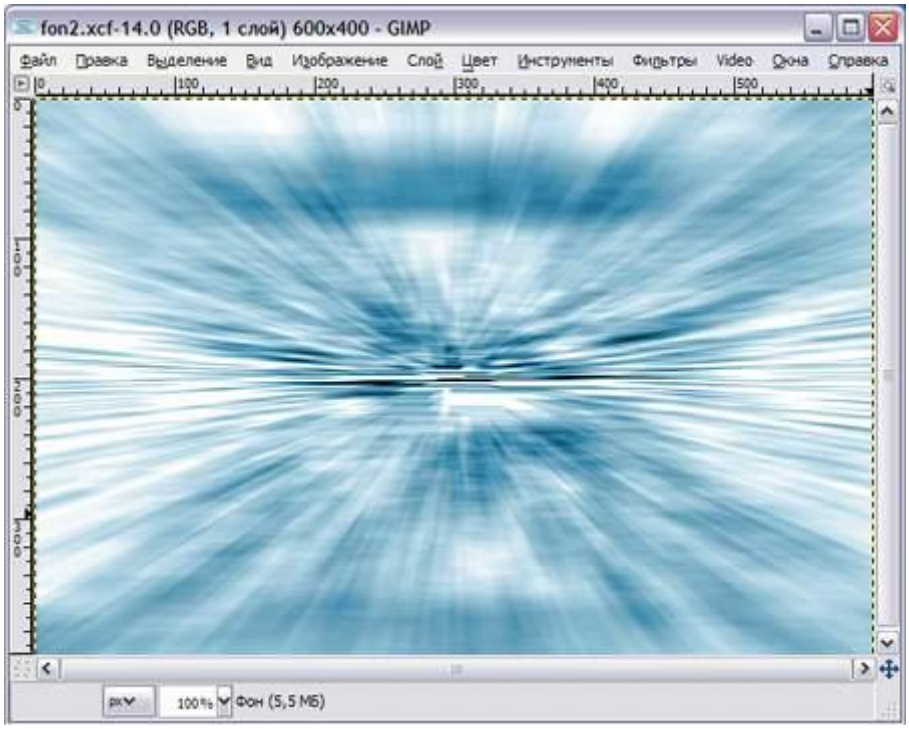
Вот результат применения фильтра размытия движением.



Обратите внимание, какие изменения произвел фильтр! Выглядит как отличная основа для нашего фона. Давайте попробуем раскрасить нашу основу. Для этого в меню Цвет выбираем пункт Цветовой баланс и подберем цвет нашего фона по своему вкусу. Мы выбрали такой вариант:

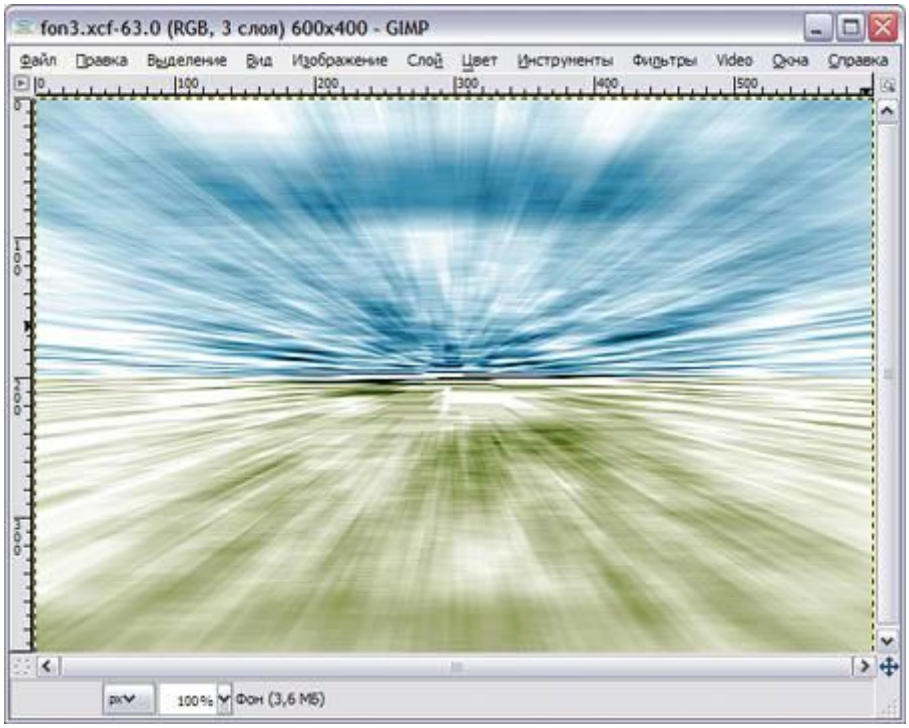


И вот что у нас получилось:



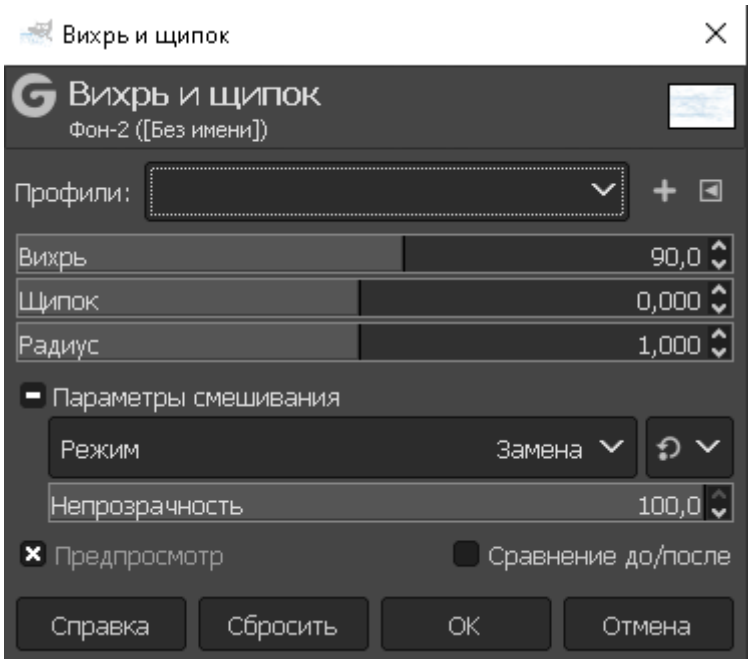
Если изображение кажется излишне размытым, можно в меню Фильтры выбрать Улучшение - Повышение резкости и поставить значение около 60. На этом уже можно и остановиться. Далее фон можно использовать по назначению, накладывать поверх какие-то объекты или надписи.

Но можно и включить фантазию и продолжать дорабатывать эту заготовку. Например, затемнив и проработав нижнюю часть изображения, получим эффект такой "*перспективной*" линии горизонта:



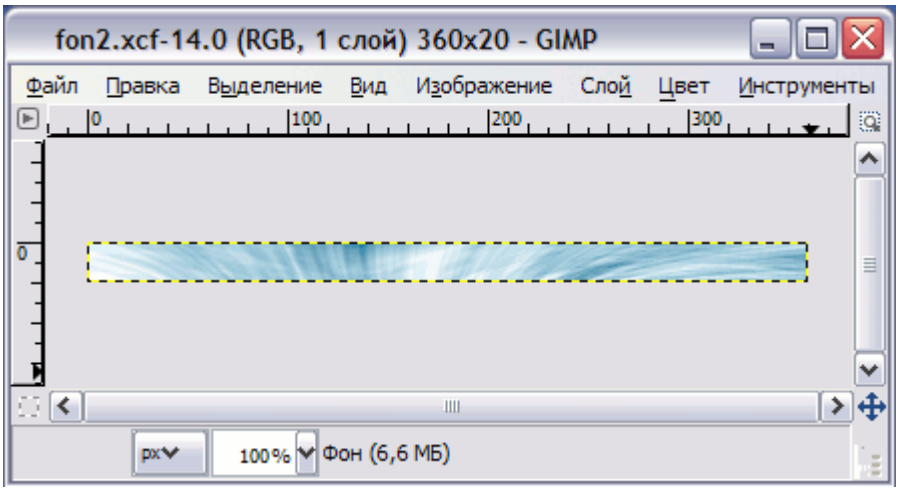
Тут мы просто выделили нижнюю часть нашей картинки и сначала немного обесцветили в меню Цвет - Тон / насыщенность, а затем перекрасив в другой цвет в меню Цвет - Цветовой баланс.

А вот еще одно применение нашего фона. Для нашей цели его сначала нужно немного доработать. Откроем меню Фильтры - Искажения - Вихрь и щипок и установим следующие параметры:



Получим приблизительно следующий результат:

Далее на Панели инструментов выбираем инструмент Кадрирование и зададим в параметрах ширину и высоту области кадрирования как 360px и 20px или если размер не имеет большого значения можно просто на глаз отметить рамкой на изображении нужную область. Не важно, как мы отметили область кадрирования, важно что до того как произвести кадрирование, мы можем двигать отмеченную рамку мышкой по нашему изображению, подбирая нужное ее положение. Щелкнув мышкой внутри этой рамки, мы произведем кадрирование.



Вот мы получили фон для красивого юзербара. Далее можно наложить тень и нужный текст и получим нечто вроде этого:

Задание 1. Попробуйте самостоятельно получить вот такое изображение. Для этого нарисуйте кляксу разными цветами.

Задание 2. Попробуйте самостоятельно нарисовать вот такой рисунок, используя меню Фильтры/Искажения/ Волны

Содержание отчета:

1. Наименование практического занятия
2. Цель занятия
3. Результат работы сохранить файлом в своей папке
4. Список используемых источников

Практическое занятие №16 Рисование в растровом редакторе

Задание 1.

Делаем из летнего пейзажа осенний.
Откроем фото с осенним пейзажем. Дважды кликаем по слою Фон и нажимаем OK.
Выбираем Меню Цвет/Тон-Насыщенность, в выпадающем меню выбираем G и ставим параметр Насыщенность и Тон На -60.
Выбираем Y и ставим параметр Насыщенность на -20.
Выбираем Все цвета и ставим параметр Тон на -30, а Насыщенность на +10.

Задание 2.

Избавляемся от кожных проблем.

Откроем файл Сначала определимся с инструментами, которые мы будем использовать.->

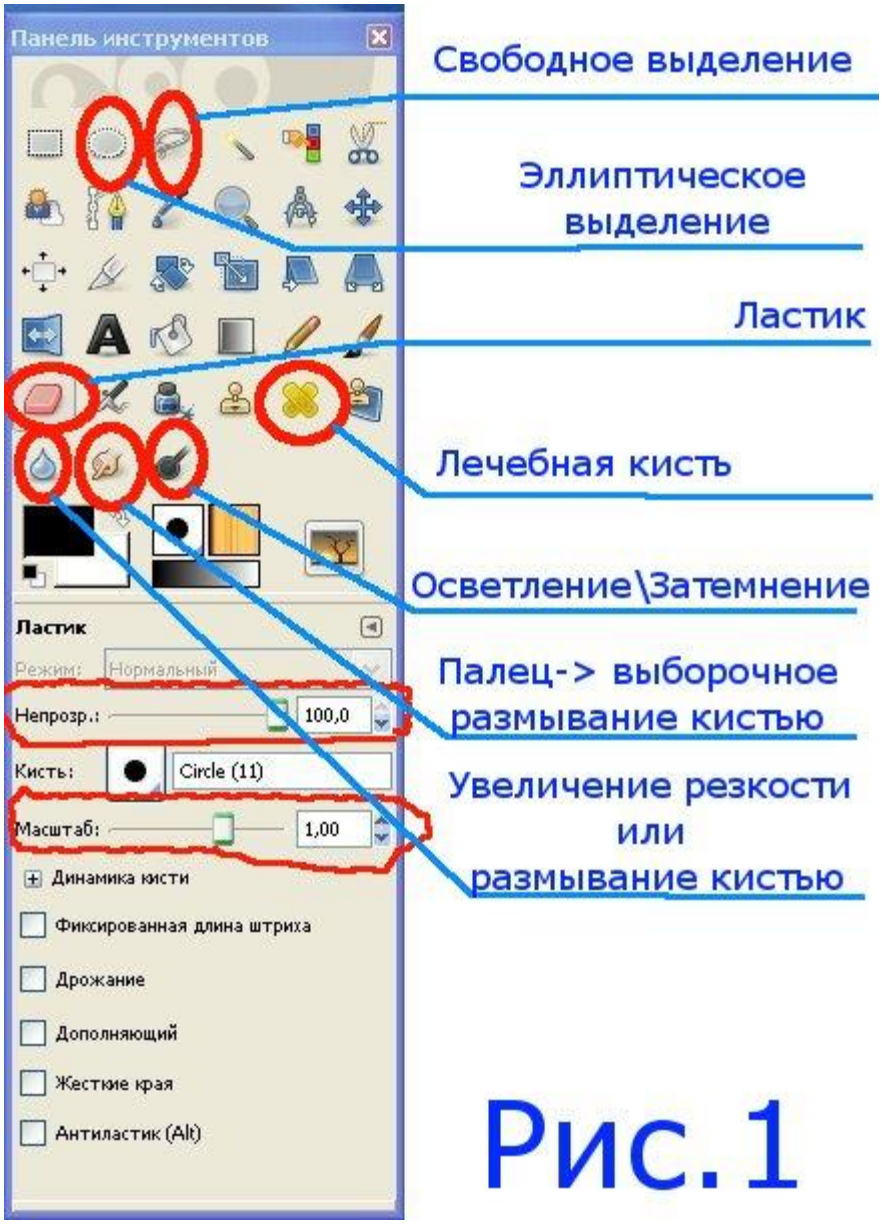


Рис.1

Воспользуемся эллиптическим выделением . Выбираем самую светлую часть лба подводим курсор и нажимаем левую клавишу мышки, и не отпуская рисуем круг на самой светлой части лба.->

Далее воспользуемся инструментом "Палец". Устанавливаем масштаб для этой кисти 6,5 (Рис1, панель управления), и размазываем выделение так, чтобы получилась самая светлая картинка равномерной окраски.->

Для равномерности размывания воспользуемся размыванием Гаусса->"Фильтры"-> "Размывание"->"Гауссово размывание". Сразу скажу, что я использовал все настройки по умолчанию, и их не менял.->

Теперь воспользуемся "Лечебной кистью". Устанавливаем масштаб 6,5, подводим курсор к центру нашего кружочка, нажимаем клавишу на клавиатуре "Ctrl" и не отпуская её щёлкаем левой клавишей мышки. У нас появится кружок с крестиком в центре в этом месте (он так и будет оставаться на месте, пока мы будем работать с "Лечебной кистью"). Теперь подводим курсор к границе нашего кружочка и методично начинаем убирать все дефекты. При этом при необходимости меняем размер и прозрачность лечебной кисти. Особо аккуратно нужно работать на границе тёмных цветов, может произойти не красивое размывание чёрного цвета. Всегда нужно помнить, что в меню "Правка", мы можем отменить предыдущее действие, а в "Диалоге действий" (меню "Окно") любое действие, которое нам не понравилось. Так работает "Лечебная кисть"->

Для обработки границ, я использовал " Лечебную кисть" с масштабом 9, и прозрачностью 13, больше я здесь ни чем не пользовался.
Вот результат->

Ну и в заключении воспользуемся меню "Цвет" окна редактирования-> "Коррекция цветового баланса" и подправим цвет по своему вкусу.
Вот результат->

Ваша оценка «Хорошо»

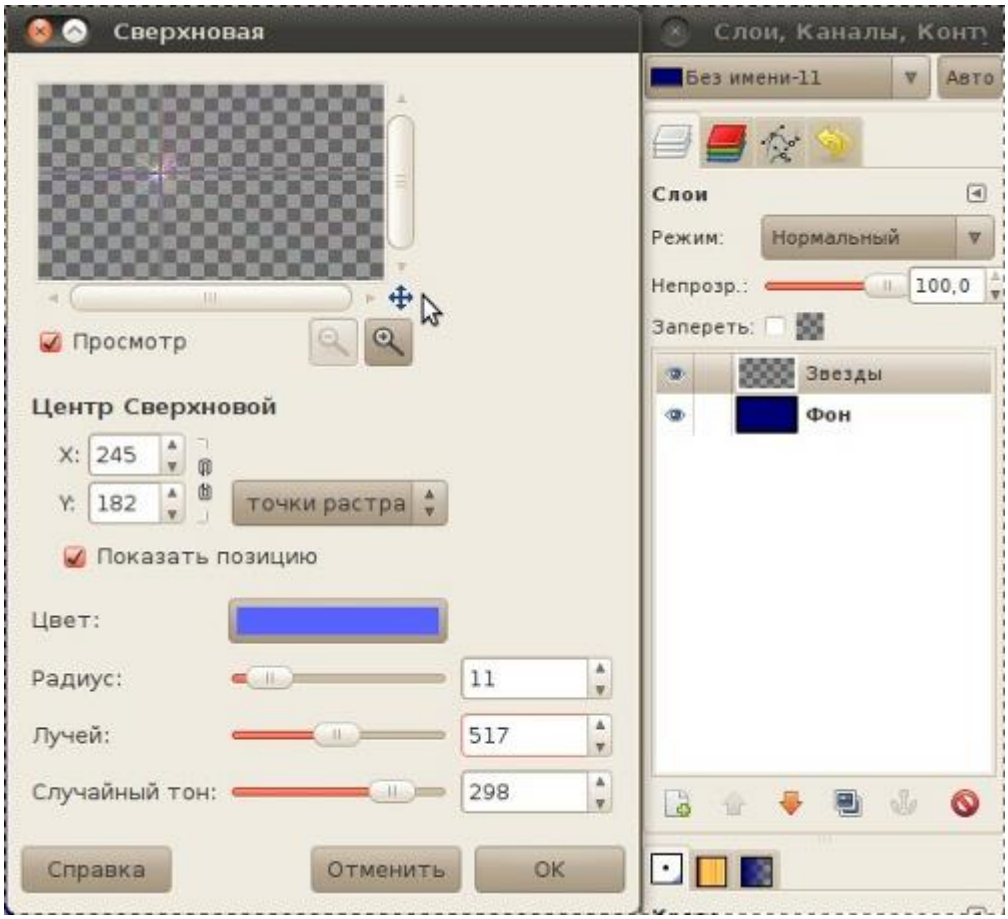
Задание 3

Чтобы свести наши манипуляции к минимуму, начнем с того, что выберем цвет неба, и назначим его в качестве цвета фона. Пускай это будет цвет #000075.

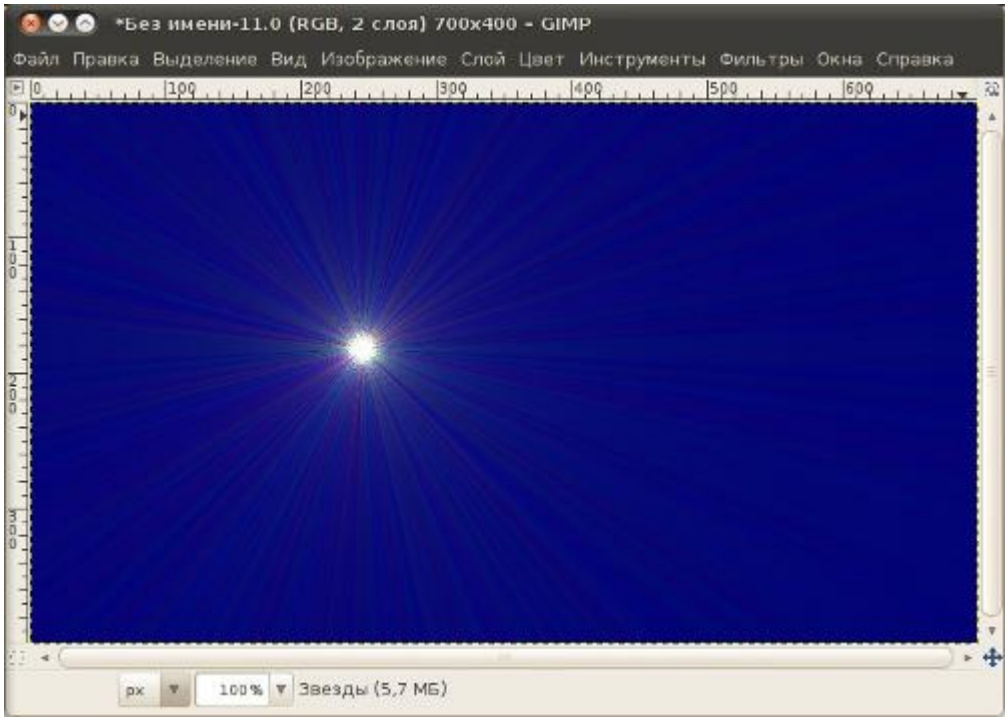
Теперь можно создавать новое изображение. Размер по желанию, мной взят - высота:400, ширина: 700, а в качестве цвета фона укажем Цвет переднего плана.

Создаем новый слой, но на этот раз с прозначным фоном. Назовем его "Звезды". На нем будем создавать звездное небо, но начнем с одной звездочки. Для этого идем в Фильтры

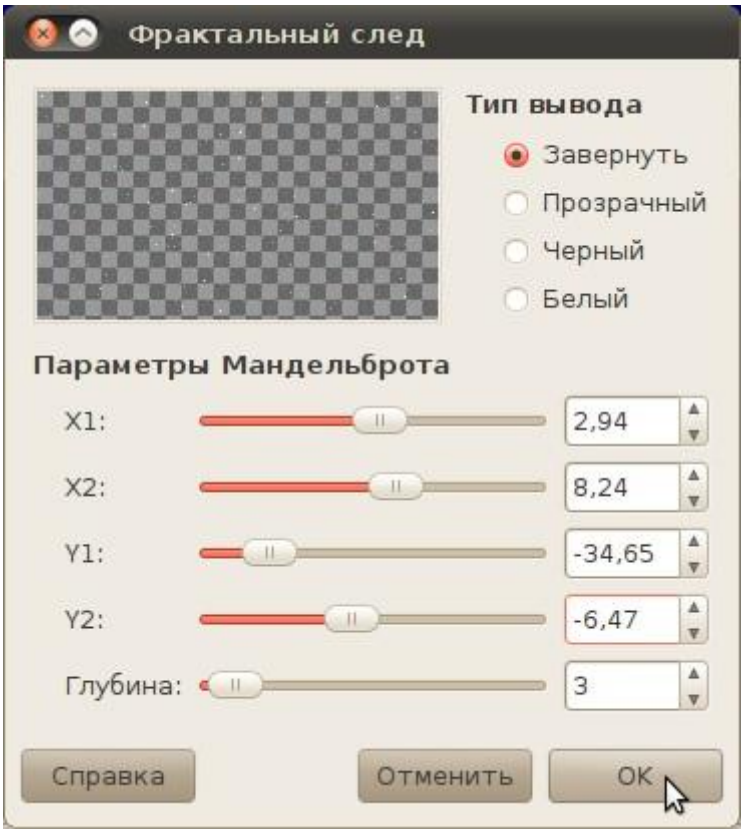
- Свет и тень - Сверхновая. Далее играем с настройками. Центр можно выбрать в любом месте. Радиус зависит от того, насколько насыщенным звездами небо вы хотите получить. Чем больше радиус, тем звезднее (Например 10-12). На насыщенность неба также влияет цвет. Сам цвет роли не играет, вы можете брать любой из спектра, а вот его яркость значение имеет. Чем светлее он будет, тем больше звезд вы уведете. (Например #5059e5). Ну а тон и количество лучей особой роли не играют:



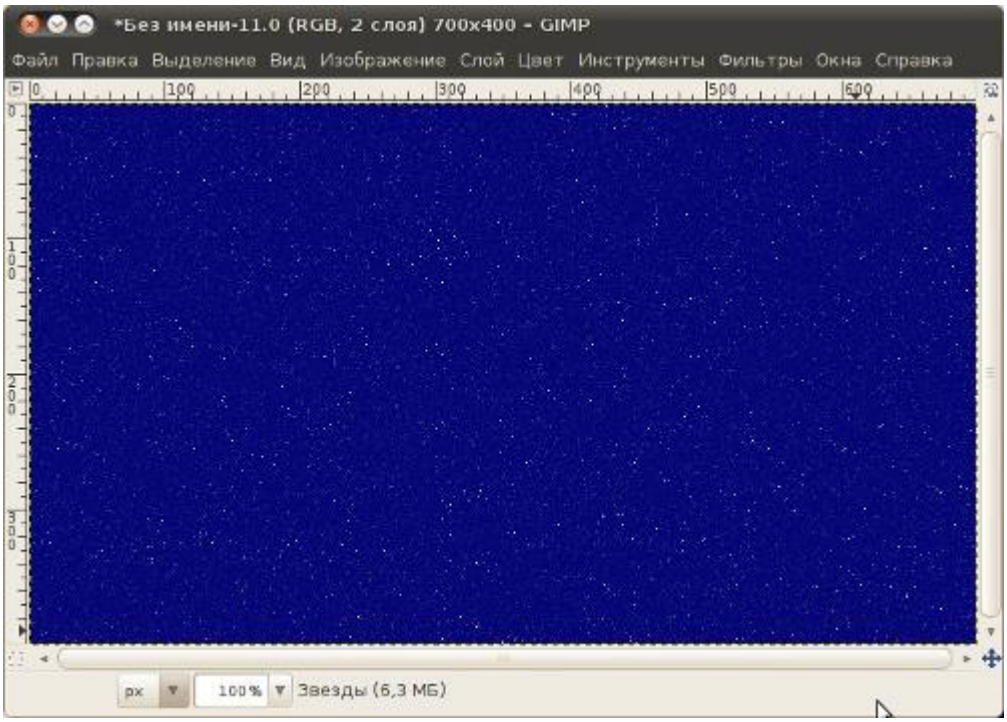
Получили такую звезду.



Одна звезда еще не небо, идем дальше: Фильтры - Карта - Фрактальный след. Значения можно поставить приблизительно следующие:



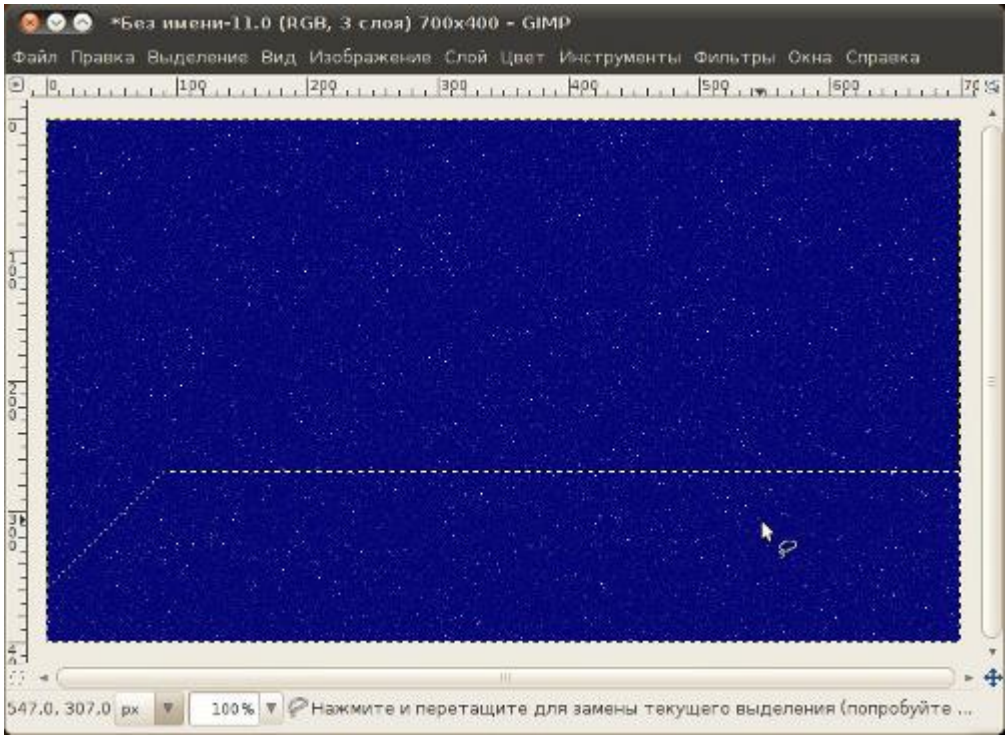
В результате вы получите кучу звездочек.



Над реалистичностью мы поработаем попозже, а сейчас внесем декорации.

Создаем новый прозрачный слой. Назовем его "Крыша". Выбираем инструмент Свободное выделение (для простоты можно воспользоваться и обычным прямоугольным выделением), рисуем крышу. Рисуем ломаной линией, проставляя точки

по углам. Контур выделения при этом необходимо замкнуть, после чего получим выделение, обозначающее будущую крышу.



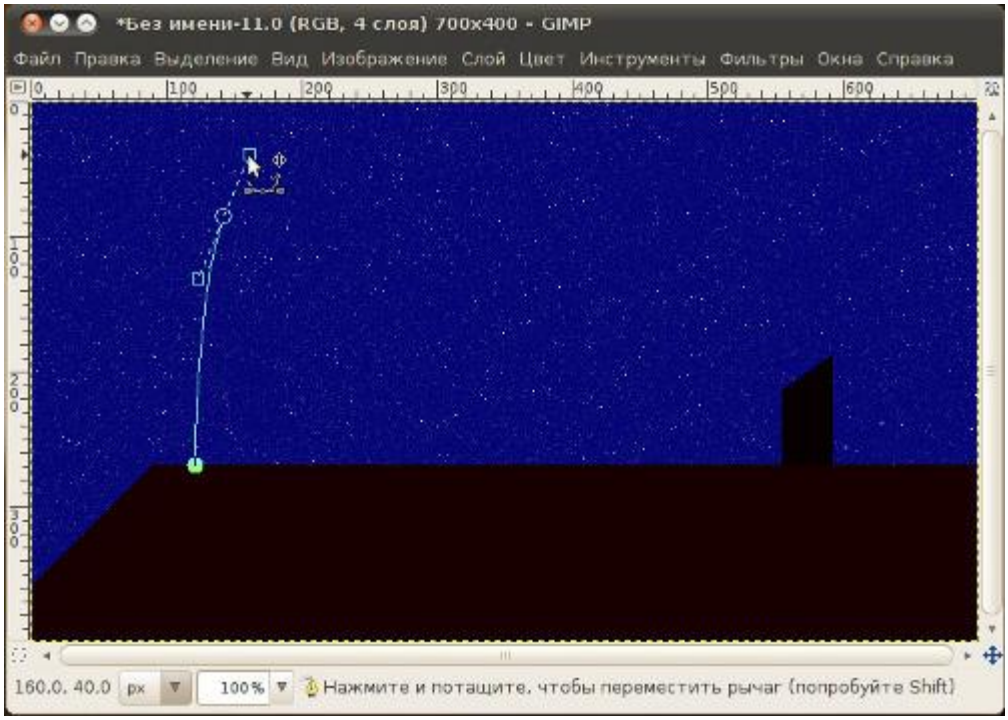
Заливаем нашу крышу темным цветом. Например таким - #190000. Можно взять черный. Не обязательно для заливки использовать инструмент Заливка.


Чтобы залить выделение основным цветом, достаточно в любой момент нажать клавиши Ctrl+<, а фоновым цветом - Ctrl+>.

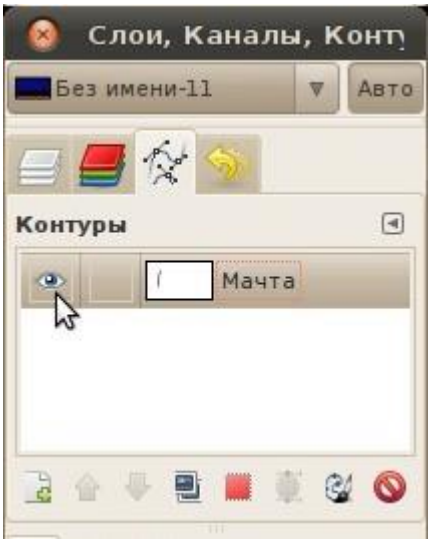
Снимаем выделение: Выделение - Снять. Можно украсить крышу печной трубой и антенной.

Трубу рисуем таким же образом, как и крышу.

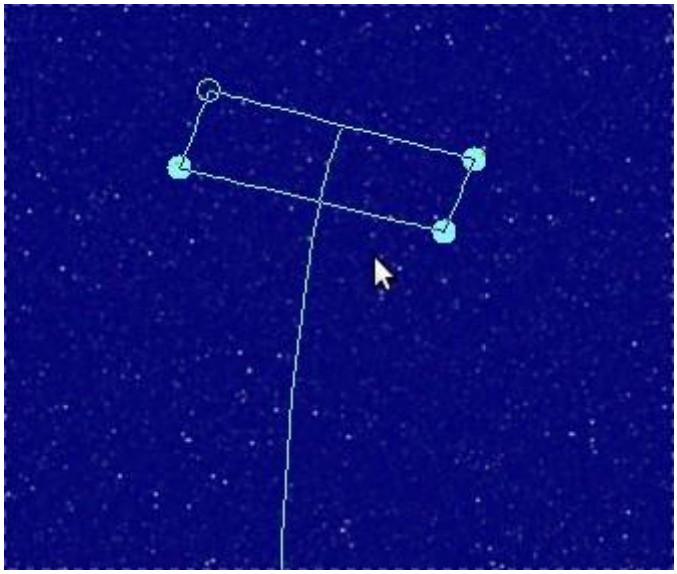
Антенну можно нарисовать кистью, но лучше поупражняться с инструментом Контуры, так как хорошее владение контурами вам обязательно пригодится. Вначале нарисуем мачту антенны. Для этого щелкнем мышкой около края крыши, появится первая точка контура. Отступите вверх и немного вправо, где будет второй конец мачты, нажмите левую клавишу мыши, и, не отпуская, немного потяните дальше, чтобы вышло примерно так:



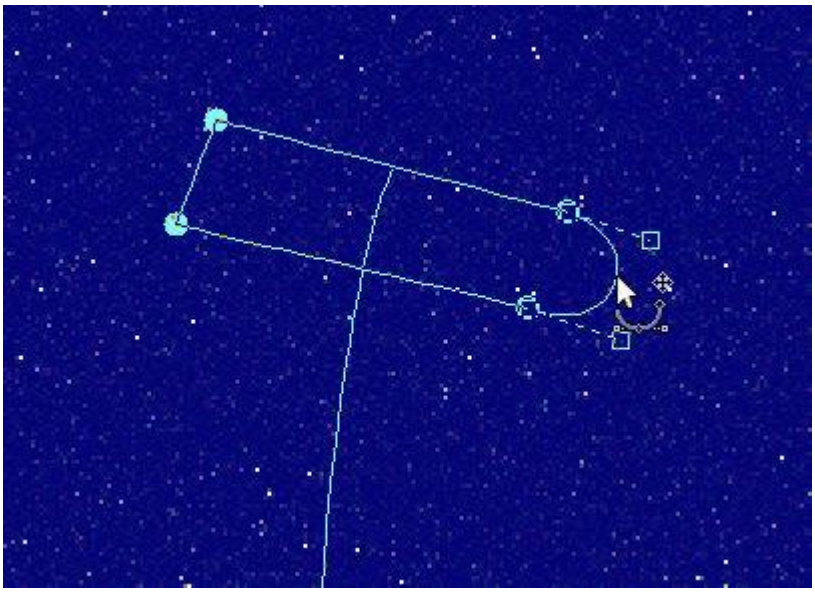
По умолчанию нарисованный контур невидим, чтобы сделать его видимым, перейдите к диалогу Контуры, и кликните на значок видимости контура.



Теперь рисуем остальные элементы антенны. Можете рисовать всю антенну одним контуром, а можете мачту оставить отдельным контуром, что даст возможность сделать мачту и элементы на ней разной толщины. Итак, сначала рисуем прямоугольник, просто проставив точки по углам. Когда дойдем до начальной точки, кликните по ней удерживая клавишу Ctrl, и контур замкнется. Получится так:

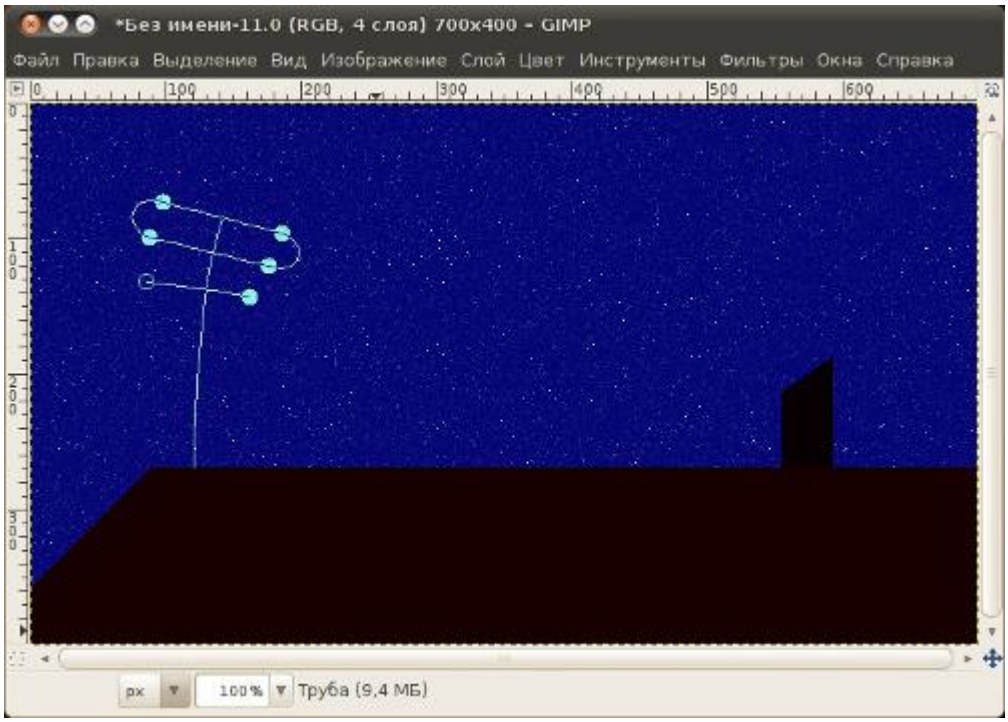


Теперь, не меняя инструмента, потяните за боковую грань прямоугольника в сторону. Прямая линия изогнется в дугу.

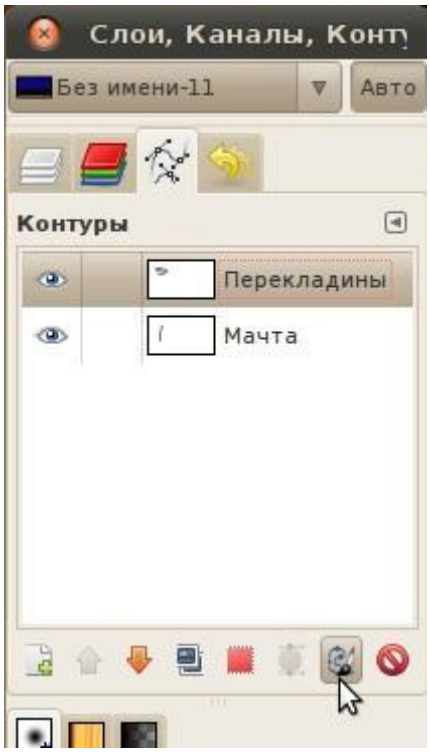


Тоже сделайте с другой стороны.

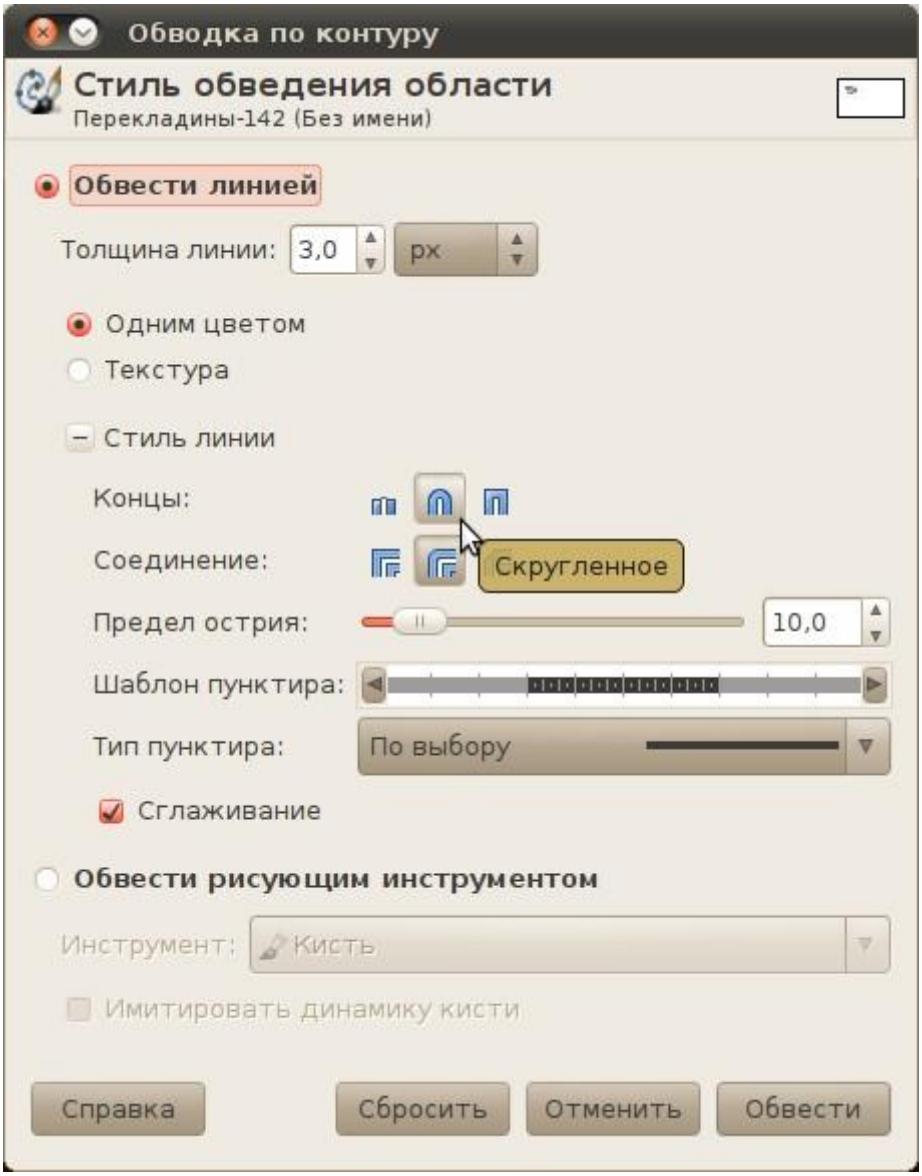
И еще на мачте добавим одну перекладину ниже.



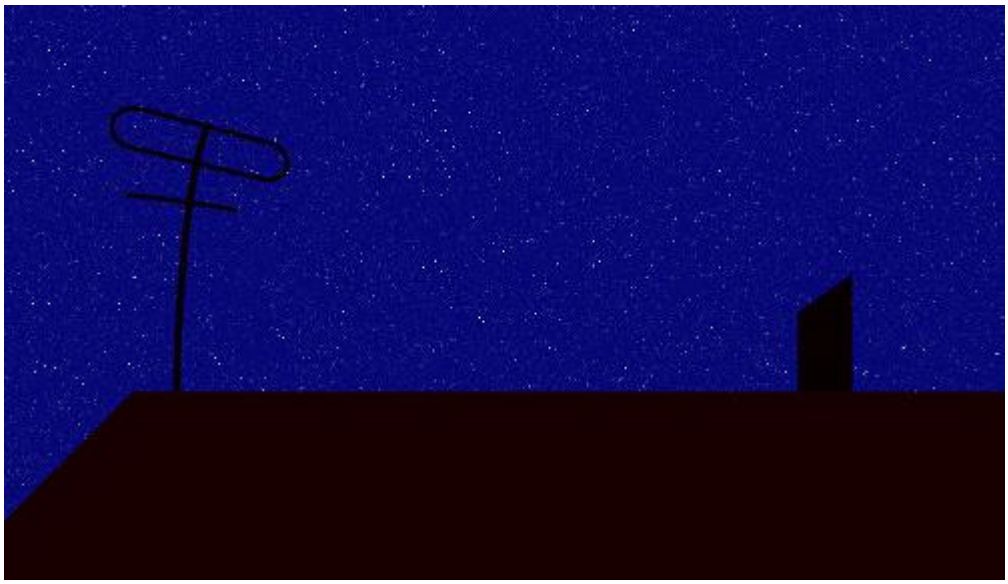
Теперь нарисованные контуры нужно обвести линиями. Это можно сделать либо через меню, либо специально для этого придуманной кнопкой под списком контуров



Вот настройки стиля обводки. Тут мы выбрали только толщину линии, и сделали концы линий закругленными.

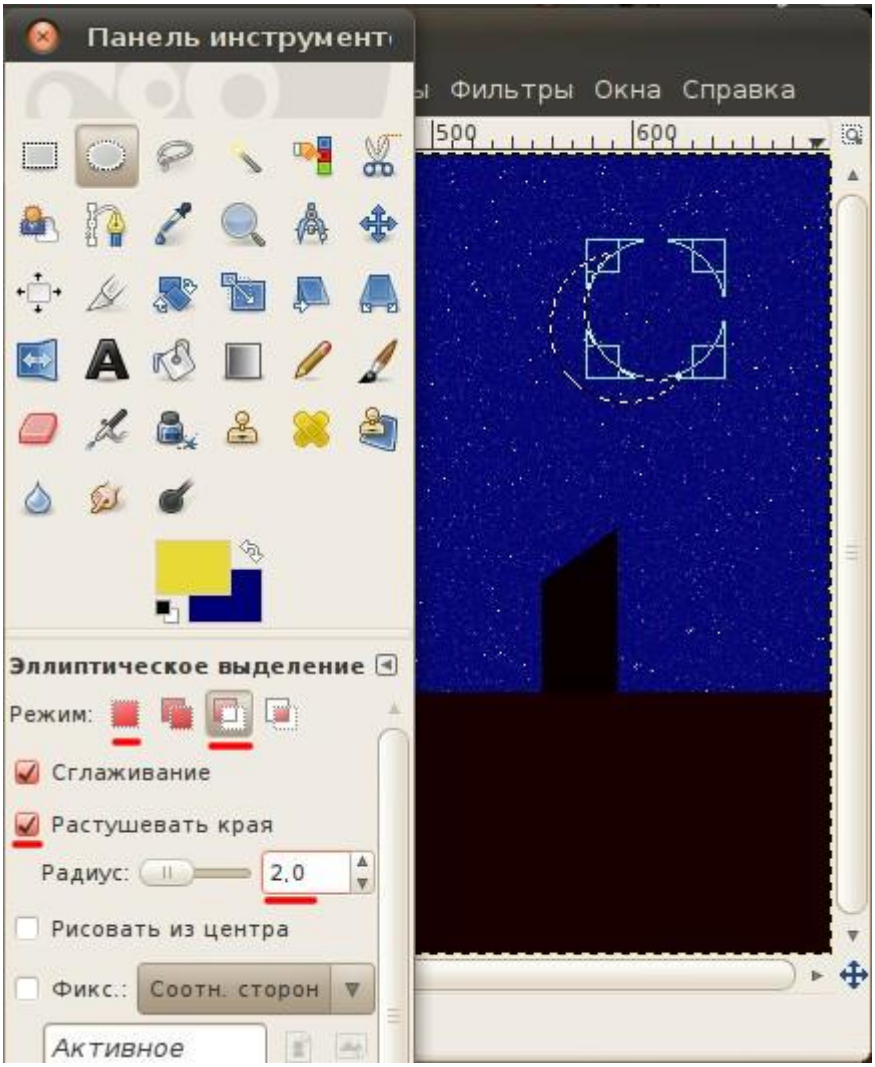


Для мачты толщину линии можно сделать больше. После обводки, сами контуры снова сделаем невидимым, чтобы не мешали.



Настала пора доработать небо. Добавим в него луну и еще немного звезд. Вернемся в диалог Слои, создадим новый прозрачный слой, назовем его "Луна". Берем инструмент

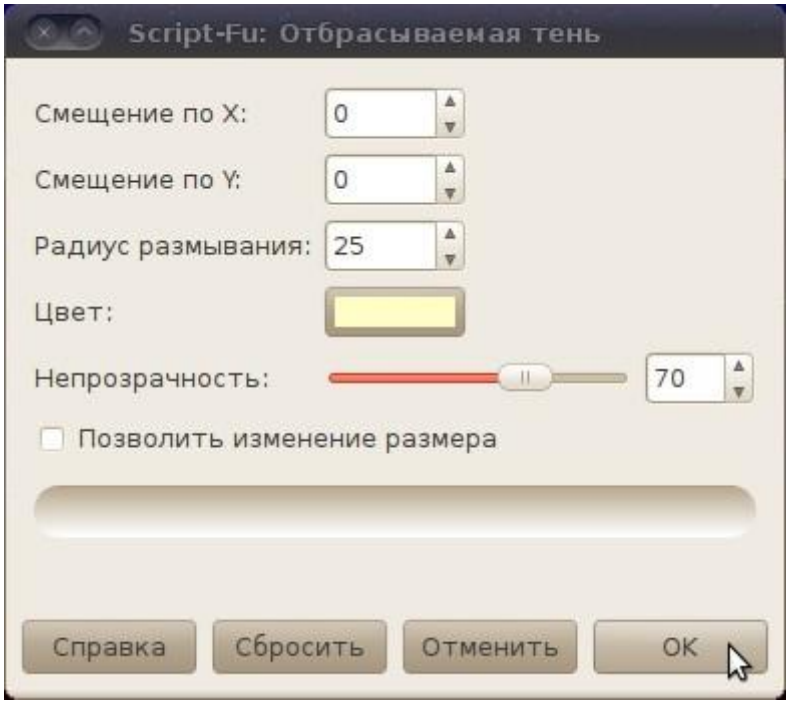
Эллиптическое выделение (режим - заменить текущее выделение), рисуем им круг, удерживая клавишу Shift. Затем переключаем на режим - вычесть из текущего выделения. И теперь рисуем круг поверх предыдущего. В результате должен получиться серп:



Чтобы края были не очень острые, мы сразу в настройках выделения поставили небольшую растушевку краев. Заливаем выделение желтым цветом на ваш вкус. Лучше всего смотрятся бледно- желтые и голубые цвета. А еще лучше залить не сплошным цветом, а какой-нибудь подходящей текстурой, но я решила не усложнять.

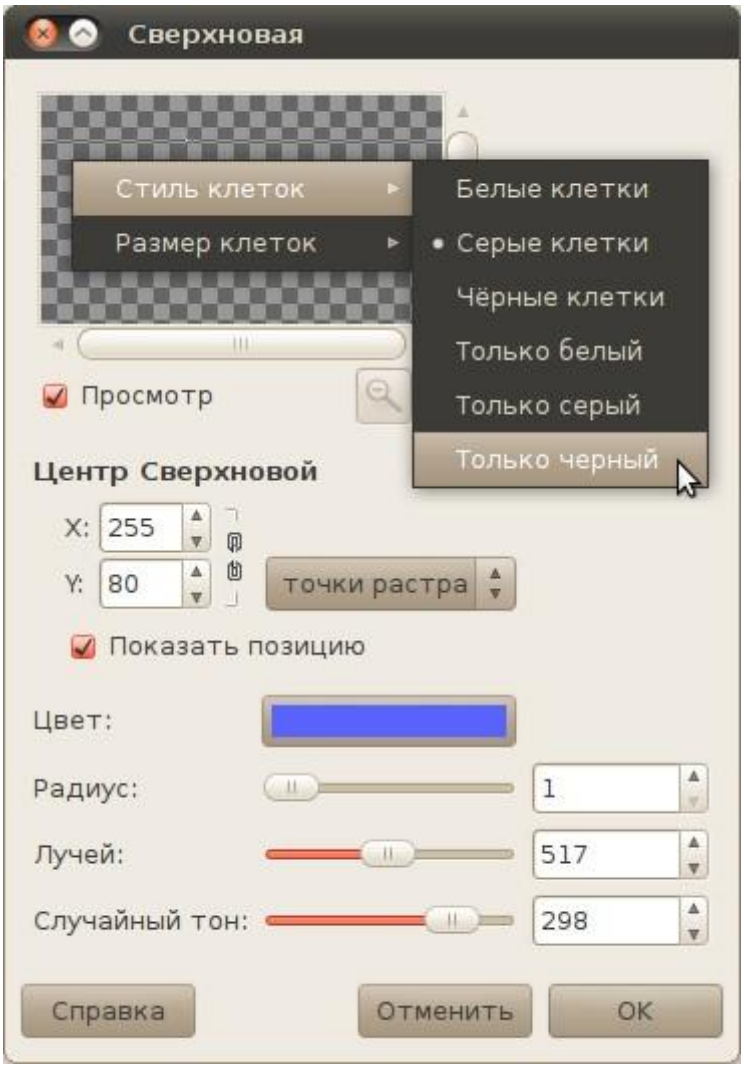
При необходимости луну можно передвинуть или повернуть. Делается это с помощью инструментов Перемещение и Вращение.

Теперь луне добавим лунного свечения. Фильтры - Свет и тень - Отбрасываемая тень. Пусть название вас не смущает, по сути свет и тень - это одно и тоже, разница лишь в цвете. А его лучше выбрать каким-нибудь светлым и бледным, как у луны.



Непрозрачность нашей светлой тени поставим 70-90%.

Теперь для колорита попробуем нарисовать Большую Медведицу. Создадим для нее новый прозрачный слой Медведица. Идем в Фильтры - Свет и тень - Сверхновая. Настройки в фильтре остались те, которые мы ставили, когда создавали звезду в начале урока. Меняем в них только радиус на 1 и координаты. Чтобы звезду было лучше заметно в окне предпросмотра, можете кликнуть там правой кнопкой мыши и выбрать в качестве фона черный цвет.



Неудобство фильтра в том, что за один раз мы можем поставить только одну звезду, поэтому придется использовать его столько раз, сколько звезд нам потребуется. Для удобства можно увеличить поле предпосмотра, когда вы будете рисовать созвездие. Чтобы быстро вызывать фильтр Сверхновая, не обращая каждый раз к меню, используйте комбинацию клавиш Ctrl+Shift+F.

Когда мы закончим рисовать звезды в созвездии Большой Медведицы, у нас должна быть где-то такая картина:

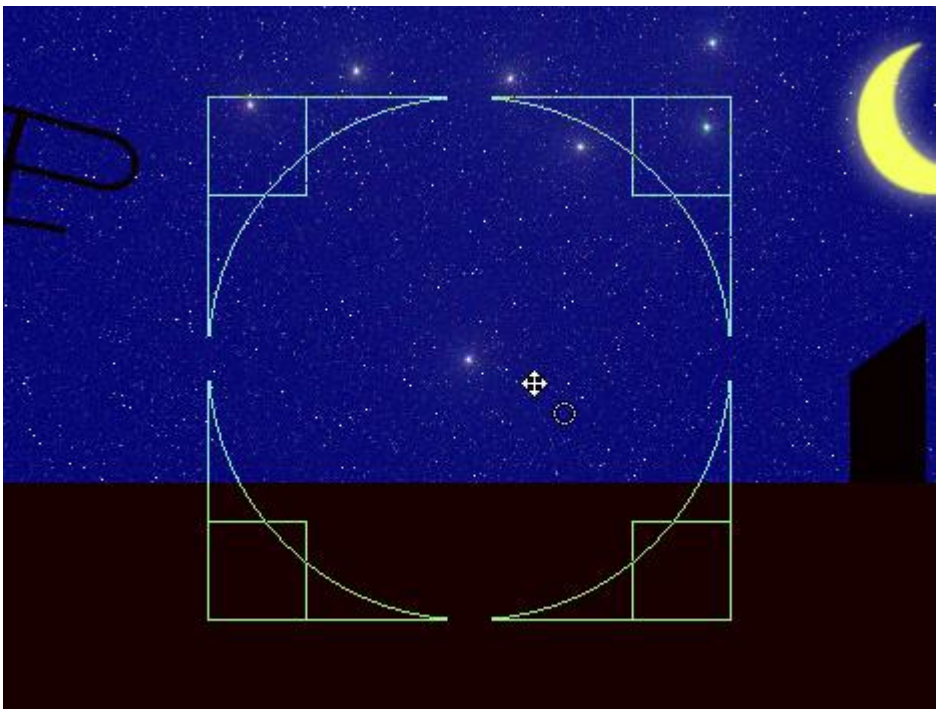
Если созвездие вам показалось не выразительным, можете один или несколько раз дублировать слой Медведица. Звезды в созвездии станут ярче.

Рисование созвездий с помощью фильтра Сверхновая мы уже освоили. Дело оказалось нехитрое, но довольно нудное. А теперь пришло время освоить еще один способ рисования созвездий и галактик.

Начинаем настоящую магию!

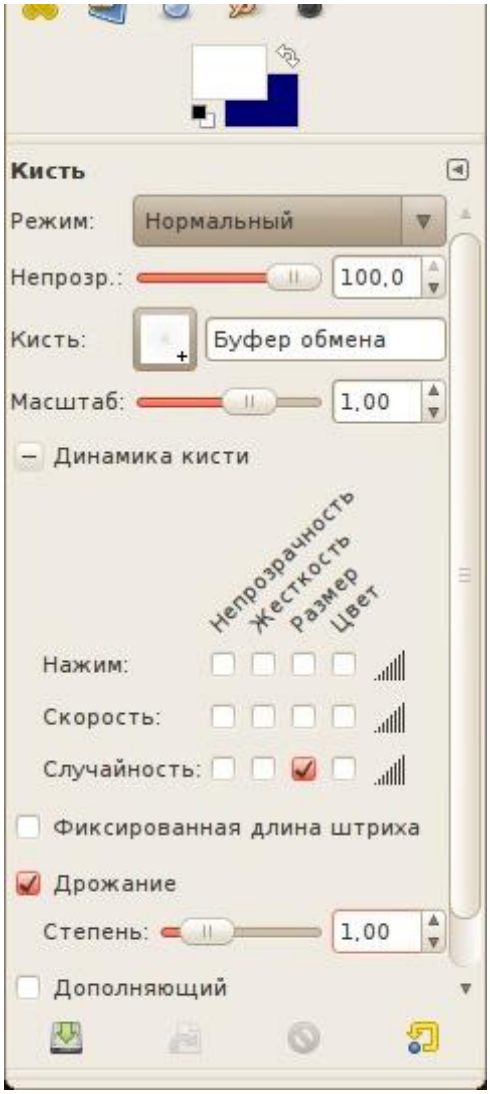
Создайте еще один новый прозрачный слой, назовите его, например, Млечный путь. Уже привычным способом создайте на нем еще одну Сверхновую с теми же настройками, как в последний раз, только где-нибудь по середине изображения.

Теперь эллиптическим выделением , удерживая клавишу Shift, обведите эту звезду, и если нужно, подвиньте выделение так, чтобы его центр попал как-раз на звезду.



Теперь скопируйте выделенное, и можете отменить последние 2 действия (Выделение и Сверхновая), они нам уже не нужны. Должен остаться только чистый слой Млечный путь.

Теперь выбираем инструмент Кисть , и в списке кистей выбираем самую первую кисть - в ней как-раз и содержится звезда, которую мы скопировали в буфер обмена. В настройках кисти поставьте следующие настройки:

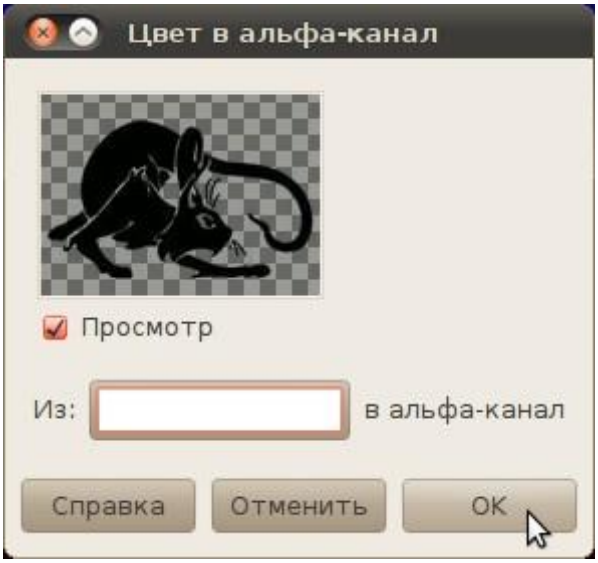


Теперь проведите кистью по небу, и посмотрите что получается. На небосводе тут и там должны появляться новые звезды разного размера. Кисть с такими настройками называется динамической кистью, т. к. её параметры меняются во время движения. Подберите значения диаметра кисти и её дрожания на свое усмотрение.

Для разнообразия можно добавить еще несколько больших звездочек, как в созвездии медведицы. Получается где-то так:

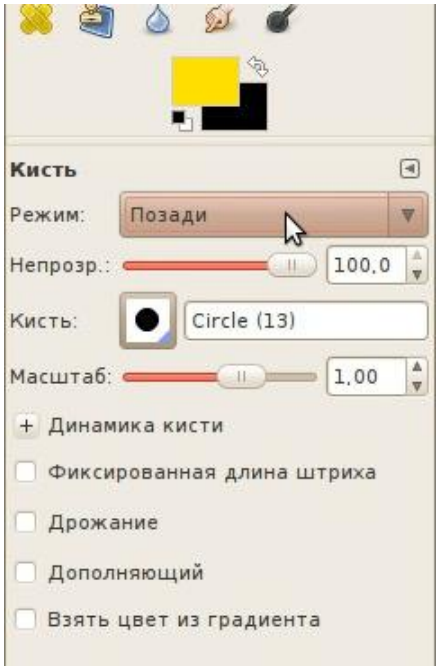
Осталось добавить главного героя. Вот такого кота:

Добавляем его в изображение - Файл - Открыть как слои или Ctrl + Alt + O. На картинке кот находится на белом фоне, от которого нам нужно избавится. В нашем случае это легко можно сделать так: выберите в меню Цвет - Цвет в альфа-канал, в качестве удаляемого цвета выберите белый. В окошке предпросмотра вы тут же увидите, что белый фон исчез.



Теперь инструментом Перемещение  двигаем кота в нужно место.

Удаляя белый фон, мы случайно оставили кота без глаза, поэтому глаз вернем на место, а заодно и цвет подберем более подходящий. Выберите цвет для глаз кота, потом возьмите какую-нибудь небольшую жесткую Кисть  , и установите в её настройках Режим наложения - Позади.



Теперь просто закрасьте глаз.

Ну вот, в общих чертах, и все. Можете добавить свои детали, если вам кажется, что чего-то не хватает. Например, если добавить подпись, то получится неплохая открытка.

Последний штрих - сохранение изображения в нужном формате, например JPG.

Готово

Ваша оценка «Отлично»

Практическое занятие №17 Основы работы с масками и каналами. Работа с палитрами, цветовыми моделями, фильтрами.

Цель:

1. Ознакомиться с элементами рабочего окна программы Inkscape.
2. Научиться создавать простейшие векторные объекты: прямоугольники,

овалы, многоугольники, звезды, спирали.

3. Научиться редактировать заливку и штрих объекта.

Теоретическая часть

Перемещение холста в inkscape

При работе с холстом inkscape очень полезны навыки работы с масштабированием и перемещением рабочей области. В инструкции можно получить базовые знания о том, что такое холст в inkscape. В этой части мы уделим время прокрутке и масштабу холста. Есть много способов для прокрутки документа. Можно использовать Ctrl+Стрелочки прокрутки холста с клавиатуры. Можно также перемещать полотно, средней кнопкой (колесиком) мыши. Или можете использовать полосы прокрутки (комбинация клавиш Ctrl+B позволяет показать или скрыть их). Колесико мыши также работает для прокрутки по вертикали, но если удерживать нажатой клавишу Shift, то колесико будет прокручивать холст по горизонтали.

Масштабирование холста

Самый простой способ для увеличения масштаба - это нажать на клавиатуре - и + (или =). Можно сделать это и, пользуясь мышью, нажмите Ctrl + щелчок колесиком или Ctrl + щелчок правой кнопкой для увеличения изображения, Shift + щелчок колесиком илиShift + щелчок правой кнопкой чтобы уменьшить масштаб. Можно просто вращать колесико, удерживая клавишу Ctrl. А еще можно воспользоваться полем "масштаб" в нижнем правом углу окна документа, здесь можно установить точное значение масштаба в %. А еще в боковой панели инструментов есть инструмент "Масштаб" в виде значка с лупой.

А еще inkscape хранит историю изменения масштаба, которые вы выполняли в ходе работы, используйте клавиши ` или Shift+` что бы вернуться к предыдущим настройкам или последующим.

Создание и работа с документами в inkscape

Чтобы создать новый пустой документ, используйте главное меню "Файл" - "Создать" или нажмитеCtrl+N. Чтобы открыть существующий SVG документ, используйте "Файл" - "Открыть" илиCtrl+O. Чтобы сохранить документ, используйте главное меню "Файл" - "Сохранить" или Ctrl+S, либо "Сохранить как" Shift+Ctrl+S, чтобы сохранить под новым именем.

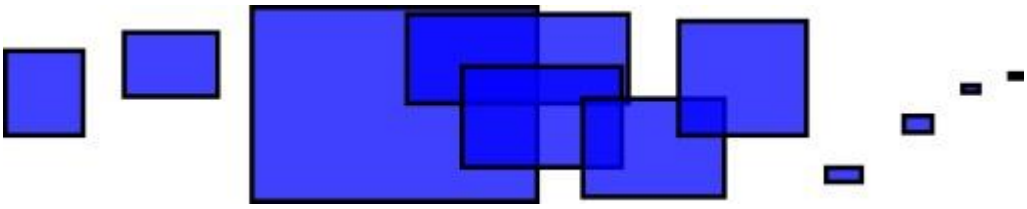
Inkscape использует формат SVG (масштабируемая векторная графика) для своих файлов. SVGявляется открытым стандартом векторной графики. SVG файлы базируются

на XML и могут редактироваться в любом редакторе текста или XML (т.е. даже без inkscape). Кроме этого SVGфайлы Inkscape можно импортировать и экспортировать в другие форматы, например, EPS, PNG.

Inkscape открывает каждый новый документ в отдельном окне. Таким образом, перемещаться между документами можно так же, как вы переключаетесь между окнами Windows. Возможно, вам будет интересно узнать, что переключаться между окнами в Windows можно с помощью сочетания клавиш Alt+Tab.

Создаём фигуры в inkscape

Щелкните мышью по инструменту "прямоугольник" в боковой панели инструментов (или нажмите клавишу F4). При активном инструменте для рисования прямоугольников вы легко можете рисовать их на холсте.



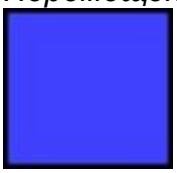
Как вы видите, у нас на картинке получились синие прямоугольники с черной рамкой, а некоторые из них вообще полупрозрачные. Если у вас получились просто белые, не расстраивайтесь, о том, как менять цвета и прозрачность, будет написано ниже. А сейчас потренируйтесь рисовать овалы и круги, звездочки и спирали.

Для каждой из этих фигур следует использовать свои инструменты и боковой панели инструментов. Где, какой инструмент в панели понятно по их значкам, но на всякий случай: "круги и эллипсы" кнопка F5, "звездочки и многоугольники" кнопка *, "спирали" - кнопка F9.

Каждая нарисованная вами фигура имеет на своем контуре ромбики - это узлы. О том, как работать с узлами inkscape можно прочитать в инструкции. Пока попробуйте перетащить их мышью и посмотрите, как будет вести себя фигура. Для более точной настройки каждой отдельной фигуры на верхней контекстной панели инструментов, есть параметры ее настройки.

Чтобы отменить последнее действие, нажмите Ctrl+Z. (Или, если вы передумали отменять и снова хотите вернуть назад, вы можете повторить отмененное действие Shift+Ctrl+Z)

Перемещение, масштабирование, поворот



Наиболее часто используемый инструмент inkscape - это инструмент выделения и трансформации. Это верхний инструмент в боковой панели инструментов, выглядит он как черная стрелка. Инструмент выделения можно активировать с клавиатуры по кнопке F1 или Пробел (самая большая кнопка на клавиатуре). Этим инструментом вы можете выбрать любой объект на холсте. Просто щелкните по нему.

Если объектом является прямоугольник, например, такой как на рисунке справа, то вы увидите восемь маркеров в форме стрелок вокруг объекта. Теперь вы можете:

- Перемещать объект простым перетаскиванием мыши. (Удерживайте Ctrl чтобы ограничить движение по горизонтали и по вертикали).
- Изменять размер объекта, перемещая любую из черных стрелочек. (Удерживайте Ctrl для того чтобы сохранять пропорции).

Теперь еще раз щелкните кнопкой мыши по прямоугольнику. Черные стрелочки по краям изменят форму. Теперь вы можете:

- Поворачивать объект путем перетаскивания угловых маркеров-стрелочек. (Удерживайте Ctrl для ограничения шага угла поворота 15-ю градусами. Перетащите крестик, чтобы изменить положение центра вращения).
- Наклонять объект путем перетаскивания серединных маркеров-стрелочек. (Удерживайте Ctrl, чтобы ограничить шаг угла наклона 15-ю градусами).

Во время работы с инструментом выделения и перемещения, можно использовать числовые поля параметров на верхней контекстной панели инструментов, чтобы задать точные значения для координат (X и Y) и размера (W и H) выделения. Подробнее про работу инструмента выделения и трансформации смотрите в инструкции.

Управление с клавиатуры

Одно из возможностей Inkscape, которая отличает его от большинства векторных редакторов, является возможность выполнения практически всех действий с клавиатуры без использования мыши. Нет практически ни одного действия, которого нельзя было бы сделать с помощью клавиатуры и преобразование объектов не является исключением.

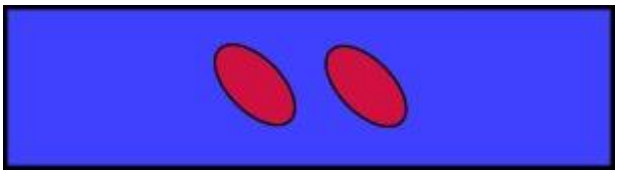
Вы можете использовать стрелочки клавиатуры для перемещения объектов, кнопки < и > для изменения размера, а вращать объекты можно кнопками [и]. По умолчанию размером шага являются 2 пикселя. Удержание клавиши Shift увеличивает этот шаг в 10 раз. Ctrl+> и Ctrl+<увеличивает и уменьшает размер до 200% или 50% от исходного, соответственно. Вращение с нажатой клавишей Ctrl происходит на 90 градусов.

Пожалуй, наиболее полезными являются изменения размеров для точных преобразований, с помощью клавиши Alt. Например, Alt + стрелочки позволяет перемещать объект на 1 пиксель. Аналогично Alt+> и Alt+< изменяют размер объекта изменяется на одну экранную точку, aAlt+[и Alt+] поворачивают его, таким образом, чтобы его смещение от центральной точки было на один пиксель.

Выбор нескольких объектов

Вы можете выбрать любое количество объектов одновременно, удерживая клавишу Shift при выборе. Или, вы можете, используя вышеописанный работу инструмент выделения и трансформации создать контур вокруг объектов, которые необходимо выбрать. Попробуйте выбрать несколько фигур. Например, создайте фигуры, как на рисунке ниже, и попробуйте выбрать их все.

А теперь посмотрите, насколько удобен способ выбора с помощью рамки, например для случая, когда вам надо выбрать два эллипса, но не надо выбирать прямоугольник.



Каждый отдельный объект, который был выбран, окружается прямоугольником из пунктирной линии. Это позволяет легко увидеть, какой объект выбран, а какой нет. Например, когда мы выбирали два эллипса без прямоугольника, без подсказки было бы трудно угадать, выбраны эллипсы или нет.

Повторный щелчок мышью по выбранному объекту с нажатой клавишей Shift приводит к его исключению из выделения. Выберите все три объекта из примера на рисунке выше, а затем, используя Shift + щелчок мыши исключите оба эллипса из отбора, оставив только прямоугольник.

Нажатие Esc для снятия выделения со всех выбранных объектов. И наоборот, Ctrl+A выбирает все объекты в текущем слое (а если вы не создавали слоёв, то все объекты в документе).

Группировка объектов

Несколько объектов могут быть объединены в группу. Группа объектов ведет себя как единый объект при перетаскивании или изменении его размера. На рисунке ниже три объекта в левой части являются независимыми, сгруппируйте их. Те же три объекта на правом рисунке сгруппированы, визуально ничего не изменилось, но вы попробуйте перетащить группу.

Для создания группы, выберите несколько объектов и нажмите Ctrl+G. Чтобы убрать объекты из группы, выберите их и нажмите Ctrl+U. Можно группировать не только

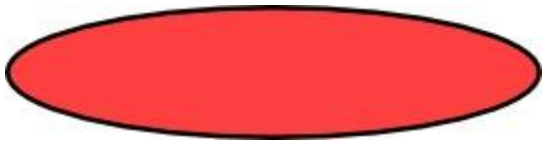
несколько объектов, но и несколько групп между собой. Сами группы могут быть сгруппированы, так же, как любые другие объекты, однако, Ctrl+U позволит разгруппировать только верхний уровень группировки. Для того, чтобы разгруппировать такую конструкцию нужно будет нажатьCtrl+U несколько раз.

Разгруппировать объекты не обязательно, если вы хотите, изменить только один объект внутри группы. Просто щелкните по нему мышью, удерживая Ctrl и он будет выбран для редактирования отдельно от группы. Или удерживайте Shift+Ctrl для выделения нескольких объектов (внутри или вне каких-либо групп) для множественного выбора независимо от группировки. Попробуйте переместить или трансформировать отдельные фигуры в группе без разгруппирования, а затем отмените выделение с конкретного объекта и выберите группу, чтобы увидеть, что она по-прежнему остается сгруппированной.

Заливки и обводки

Доступ ко многим функциям inkscape реализован через диалоговые окна. Наверное самым простым способ покрасить объект в разные цвета будет открыть активировать его и выбрать цвет на палитре цветов.

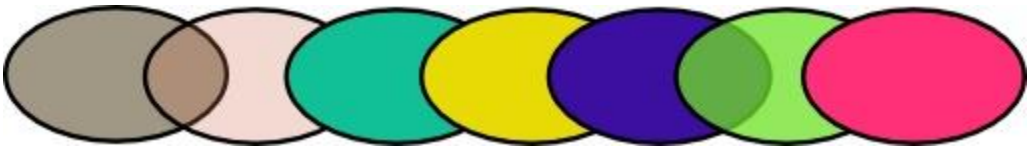
Но более мощный инструмент управления цветами объекта находится в диалоге заливки и обводки. Этот диалог доступен в верхнем пункте главного меню "Объект" или по комбинации клавиш Shift+Ctrl+F. Выберите какой-нибудь объект, например, эллипс, как на рисунке ниже, и откройте диалоговое окно заливки и обводки.



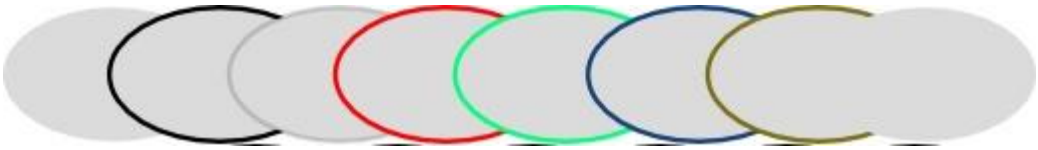
Вы увидите, что диалоговое окно имеет три вкладки: "заливка", "обводка" и "стиль обводки". Закладка "Заливка" позволяет редактировать заливку выбранных объектов. С помощью кнопок сразу под названием закладки, можно выбрать типы заливки, включая и первый пункт "нет заливки" (кнопка с крестиком), сплошной цвет заливки, а также линейные и радиальные градиенты. Для рисунка выше была активирована вторая кнопка сплошной заливки.

Ниже кнопок с видами заливки на закладке "Заливка" вы увидите варианты выбора цвета. Для каждого способа выбора цвета есть своя собственная вкладка: RGB, CMYK, HSL и "Круг". Довольно удобно выбирать цвет на закладке "Круг", где вы можете повернуть треугольник внутри круга, чтобы выбрать цвет на цветовом круге, а затем выберите оттенок этого цвета в треугольнике. Для всех цветов изменять прозрачность можно с помощью ползунка "Альфа-канала" (прозрачность). Прозрачность измеряется в % непрозрачности, т.е. 100% абсолютно непрозрачный цвет и, наоборот, 0% абсолютно прозрачный.

Всякий раз, когда вы выбираете объект, цвет в описанном выше диалоговом окне обновляется, показывая цвет текущей заливки и обводки. Если выбрано несколько объектов, диалоговое окно показывает их средний цвет. Поэкспериментируйте с этим диалоговым окном. Теперь вы можете создавать разноцветные фигуры и даже полупрозрачные.



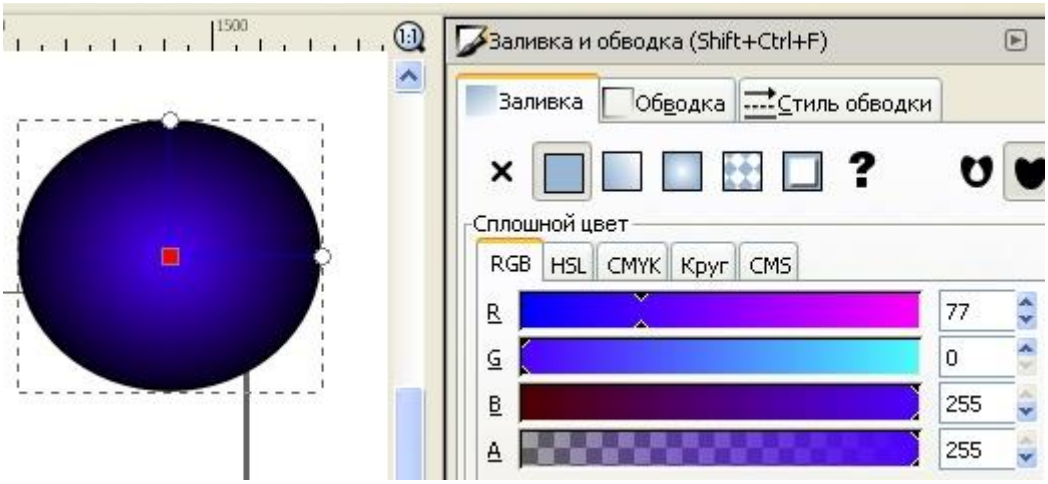
На закладке "Обводка", можно сделать обводку объекта без заливки или сделать заливку сплошного цвета и настроить ее прозрачность:



Последняя закладка в этом диалоговом окне "стиль обводки", позволяет задать ширину, тип линии обводки и другие параметры обводки объекта:

Ну, и, наконец, вместо того, чтобы заливать фигуры сплошным цветом, можно использовать градиенты для заливки или обводки:

При переходе от сплошной заливки цветом к градиенту создается настройка градиента, для идентификации которой программа присваивает настройке номер.



Только что созданная настройка градиента использует предыдущий цвет сплошной заливки фигуры, который переходит из непрозрачного цвета в прозрачный. Для управлении градиентом заливки фигуры активируйте инструмент градиент в боковом окне инструментов или нажмите Ctrl+F1. Появятся две направляющие градиента в заливке фигуры. Перемещаете направляющие (ручки) градиента и вы увидите как он изменяется. Направляющие градиента в зависимости от их длины изменяют насыщенность и форму градиента. Так изменяя длину ручек (направляющих) градиента можно из кругового

градиента сделать овальный и т.д. Цвет градиента можно менять в палитре цветов inkscape или в диалоговом окне по кнопке "изменить". У градиента есть два цвета, цвет ручки и цвет центрального маркера. Если нажать на центральный маркер, то в диалоговом окне отобразится его цвет и прозрачность и их можно поменять и то же самое можно сделать для цвета ручки. Для этого надо активировать маркер любой из двух ручек градиента.

Еще один достаточно удобный способ изменить цвет объекта - это воспользоваться инструментом "Пипетка" в боковом окне инструментов или F7. Просто щелкните в то место рисунка, где уже есть нужный вам цвет и этот цвет будет присвоено заливке выбранного объекта. Щелчок пипеткой с нажатой клавишей Shift назначит выбранный цвет цвету обводки.

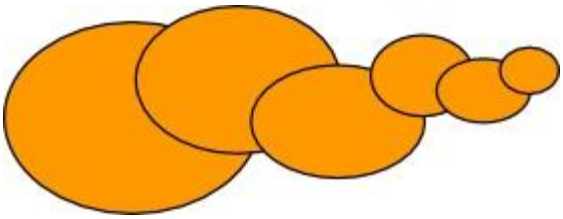
Дублирование, выравнивание и распределение объектов

Одна из наиболее часто используемых и необходимых операций - это дублирование объекта - комбинация клавиш Ctrl+D. Дубликат помещается точно над оригинала. При необходимости дубликат можно перетащить мышью или стрелочками клавиатуры в другое место. Для практики нарисуйте узкий прямоугольник и попробуйте составить из его копий квадрат, как на рисунке слева.

Скорее всего, точно выровнять прямоугольники у вас не получится или это потребует от вас значительных усилий. Но здесь на помощь вам придет диалоговое окно "Выровнять и расставить". Активировать его можно по комбинации клавиш Ctrl+Shift+A или из главного меню "Объект" предпоследний снизу пункт. Как им пользоваться. Выберите все нарисованные вами фигуры, точнее фигуру и все ее копии. Выделять фигуры мы уже научились в начале этого урока. Скорее всего оптимальным способом выделения будет в данном случае выделение рамкой. Откройте диалоговое окно "Выровнять и расставить" и нажмите на кнопку "Центрировать по горизонтальной оси", а затем кнопку "Переместить объекты так, что бы их рамки едва не пересекались". Если навести указатель мыши на кнопку, то в подсказке будет отражено название этой кнопки. Объекты будут расставлены аккуратно друг за другом. Вот некоторые примеры выравнивания и распределения:

Термин Z порядка (порядок по вертикали) относится к порядку наложения объектов в документе. То есть если какие-то объекты находятся выше других, то они должны перекрывать находящиеся ниже объекты. В главном меню "Объект" есть две команды "поднять на передний план" - клавиша Home и "Опустить на задний план" - клавиша End. С помощью этих команд можно поднимать и опускать объекты текущего слоя. Две другие команды "Поднять" Page Up и "Опустить" Page Down, будет действовать только на один один шаг, т.е. изменят порядок только одного объекта на один уровень в Z порядке. Выполняя эти действия несколько раз можно перемещать объекты последовательно от самого верхнего уровня до самого нижнего соответственно.

Попробуйте нарисовать несколько объектов и поэкспериментируйте с порядком их наложения, как мы это сделали на примере ниже:



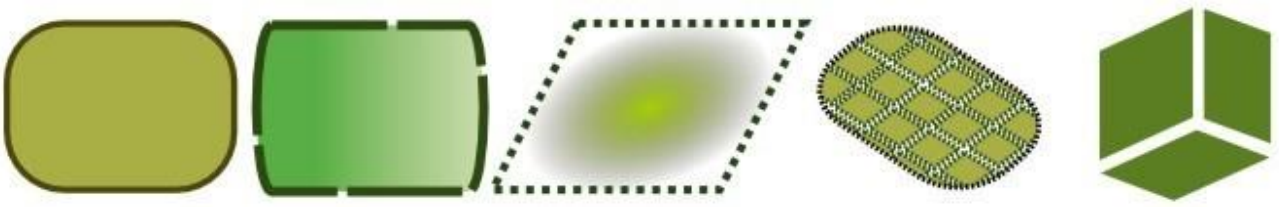
Очень полезной будет в этом случае клавиша Tab которая позволяет выделять объекты по очереди Z порядка. Если изначально не был выбран ни один объект, то нажатие этой клавиши выделит самый нижний объект, в противном случае она выбирает объект на уровень выше выбранных объектов в Z порядке. Комбинация клавиш Shift+Tab работает в обратном порядке, начиная с верхнего объекта и переходя вниз. Изначально Z порядок расставляется по мере создания вами объектов, т.е. вверху находится последний созданный объект. В этом случае, если ни один из объектов не выбран, то нажав Shift+Tab очень удобно выбирать первые созданные объекты.

Практическая часть

Задание 1.

1. Загрузите программу Inkscape, используя для этого ярлык на Рабочем столе или в Главном меню.
2. Ознакомьтесь с элементами программы: Меню, Панели инструментов, Рабочая область, Рабочий лист, Панель свойств.
3. Используя строку состояния, определите расположение начала координат на Рабочем листе.
4. Используя инструменты с панели инструментов и панель свойств, создайте следующие фигуры:

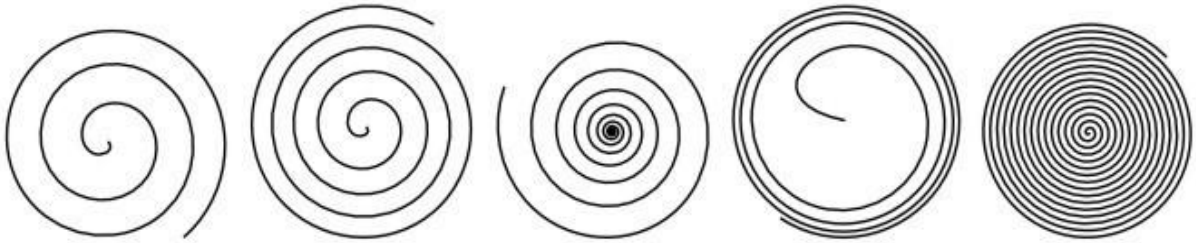
a. Прямоугольники и квадраты:



b. Круги, эллипсы и дуги:

c. Звезды и многоугольники:

d. Спирали:



5. Сохраните работу.

Практическое занятие №18 Создание коллажей и монтажей

Цель:

1. Научиться реализовывать технику работы с узлами при редактировании кривых.
2. Научиться реализовывать эффекты Вытеснение и Интерполяция для различных графических объектов.

Теоретическая часть

Теоретическая часть работы опирается на положения практической работы № 1 «Знакомство с редактором обработки графических изображений Inkscape. Панель инструментов»

Практическая часть

Задание 1. Создайте фирменный знак компании (рис. 1).

Рис. 1

1. Загрузите программу Inkscape, используя для этого ярлык на Рабочем столе или в Главном меню.
2. Нарисуйте внутреннюю окружность, используя соответствующий инструмент и клавишу Ctrl (рис. 2).
3. Аналогично нарисуйте еще три окружности.
4. Расставьте все окружности как показано на рис. 3, используя команду Объект Þ Выровнять и расставить.
5. Преобразуйте окружности в кривые, используя команду Контуры Þ Оконтурить объект.
6. Выделите верхний узел внешней окружности инструментом (Редактировать узлы контура).

Разорвите контур, нажав на (Разорвать контур в выделенном узле).

7. Переместите узлы в месте разрыва (рис. 4).

8. Выделите верхний левый сегмент, образовавшийся в месте разрыва. Вставьте новый узел . Рис. 5.

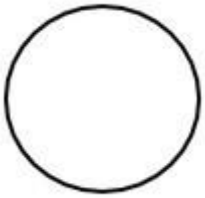


Рис. 2

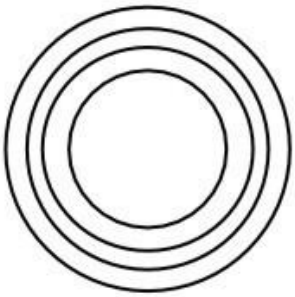


Рис. 3

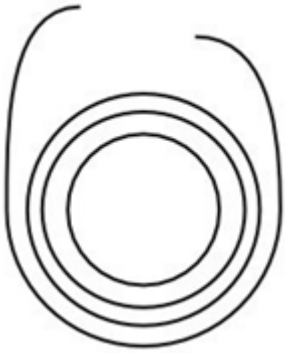


Рис. 4

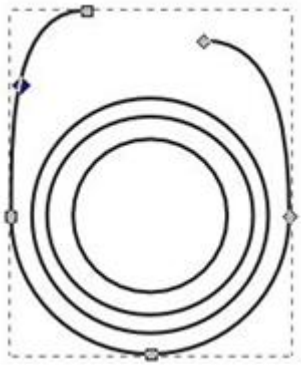


Рис. 5

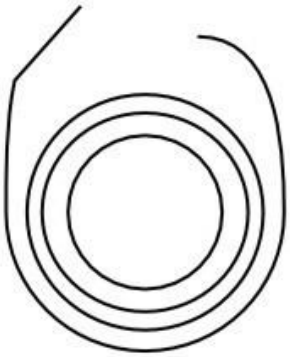


Рис. 6

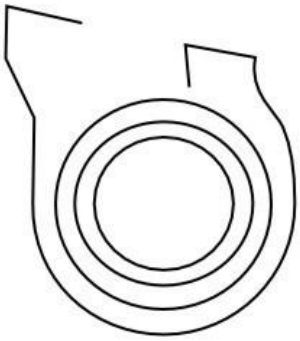
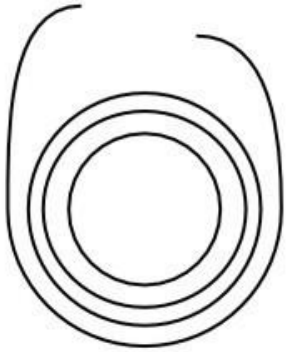
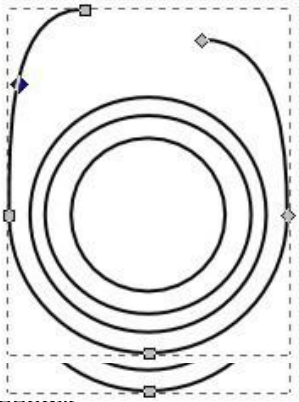


Рис. 7

Рис. 8

Рис. 9





9.  Преобразуйте новый узел в острый .

10. Преобразуйте верхний левый сегмент в отрезок прямой  (Сделать выделенные сегменты прямыми). См. рис. 6.

11. Добавляя новые узлы, добейтесь структуры кривой как на рис. 7.

12. Внутренний круг разорвите и преобразуйте аналогично внешнему кругу. Рис. 8.

13. Выделите два внешних контура, используя клавишу Shift. Выполните команду Контуры Þ Объединить. Выделите конечные узлы справа. Объедините выделенные узлы  (Соединить контуры). Тоже сделайте для левых узлов.

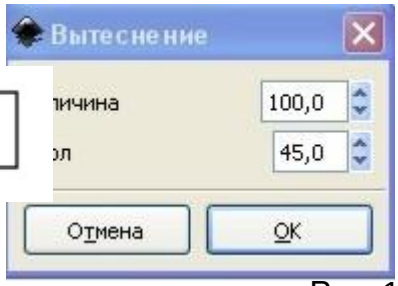
14. Выделите два внутренних контура, создайте их пересечение (Контуры Þ Исключающее ИЛИ).

15. Напишите каллиграфическим пером во внутренней окружности символ, используя соответствующий инструмент. Рис. 9.

16. Уменьшите количество узлов у символа (Контур Þ Упростить, Контур Þ Втянуть/Вытянуть).

17. Создайте пересечение контуров и символа. См. рис. 1.

Задание 2. Смоделируйте объем (рис. 12).



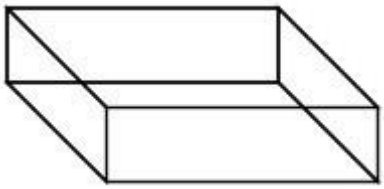


Рис. 10 Рис. 11 Рис. 12

1. Нарисуйте плоскую фигуру (рис. 10). Преобразуйте её в контур.

2. Выполните команду Эффекты Þ Использование контура Þ Вытеснение (рис. 11).

3. Результат см. на рис. 12.

Задание 3. Самостоятельно постройте изображения (рис. 13, 14), используя эффект Интерполяция.

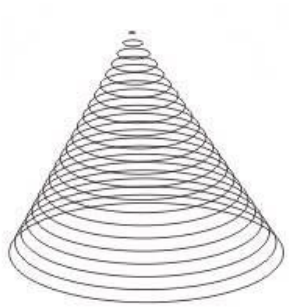


Рис. 13

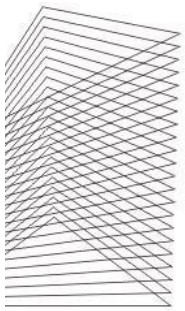


Рис. 14

Практическое занятие № 19 Ретуширование фотографий

Цель занятия:

1. *Познакомиться с объектами в программе INKSCAPE.*

2. Изучит графический интерфейс и панель инструментов программы INKSCAPE.

3. Научиться строить графические примитивы

4. Научиться выполнять операции над объектами INKSCAPE: перемещение, копирование, удаление, отражение,вращение и масштабирование

Исходные материалы и данные:

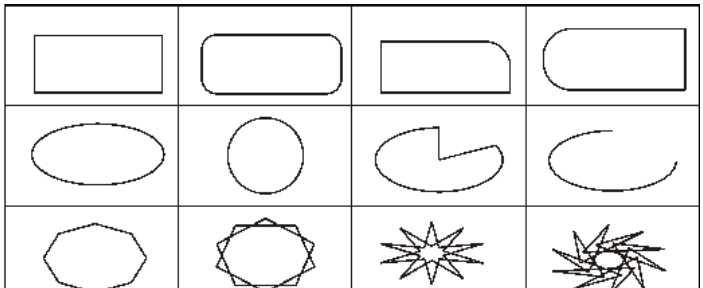
Программа INKSCAPE *Использованные источники:* [Электронный учебник INKSCAPE].

Содержание и порядок выполнения задания:

Задание №1

Открыть программу Мой компьютер-Общие документы-Мои рисунки-INKSCAPE

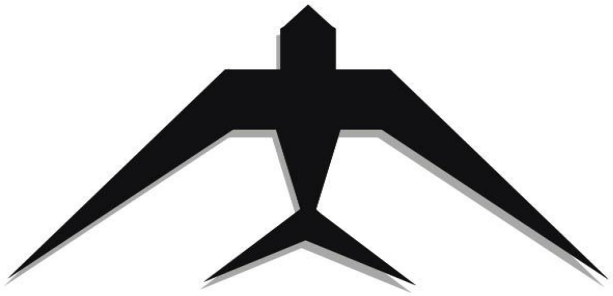
Используя инструменты Прямоугольник, Эллипс, Спираль, Полигон и Миллиметровка, построить следующие изображения:



Задание №2

С помощью простых фигур создайте следующее изображение:

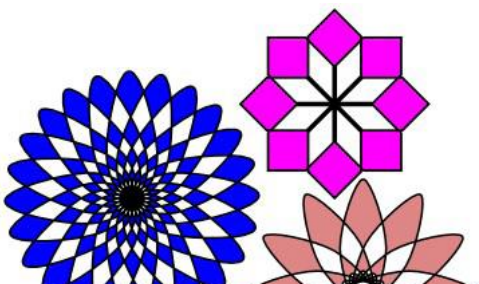
р. Вам необходимо повторить рисунок.



Простые фигуры надо объединить командой Группировать Ctrl+G, для редактирования объекта группы используйте команду Ctrl + щелчок по объекту, несколько объектов для редактирование комбинация клавиш Shift+ Ctrl+ щелчок. Разгруппировать – нажатие Ctrl+U.

Задание №4*

Постройте фигуры, представленные на рисунке



Выводы и предложения проделанной работы:
Содержание отчета:

5. Наименование практического занятия
6. Цель занятия
7. Вариант занятия
8. Результат работы сохранить файлом в своей папке
9. Список используемых источников
10. Выводы и предложения
11. Даты и подписи курсанта и преподавателя

Вопросы для самопроверки:

1. *Что является объектом в программе INKSCAPE*
2. *Структура объекта INKSCAPE*
3. Как выполняется копирование объектов
4. Как выполняется окрашивание объектов
5. Как выполняется объединение объектов в группу

Практическое занятие №20 Работа с контурами

Цель:

1. Научиться создавать текстовые объекты и изменять их свойства.
2. Научиться создавать и редактировать линейные, эллиптические градиенты.

Теоретическая часть

Теоретическая часть работы опирается на положения практической работы № 1 «Знакомство с редактором обработки графических изображений Inkscape. Панель инструментов»

Практическая часть

Задание 1. Создание текста с многоступенчатой градиентной заливкой

1. Откройте программу Inkscape, используя для этого ярлык на Рабочем столе или в Главном меню.

2. С помощью инструмента Текст создайте надпись «Радуга», измените ее размер.

3. Преобразуйте текст в контур: Контуры Þ Оконтурить объект.

4. Уменьшите количество узлов в контуре: Контуры Þ Упростить.

5. С помощью перемещения узлов измените форму контура. См. рис. 1.

6. Активизируйте инструмент Создать и править градиенты, “растяните” градиент над текстом.

7. Откройте Редактор градиентов, щелкнув два раза ПКМ по начальной или конечной точке градиента.

8. В диалоговом окне Редактор градиентов выберите из раскрывающегося списка первую точку, укажите ее цвет - красный.

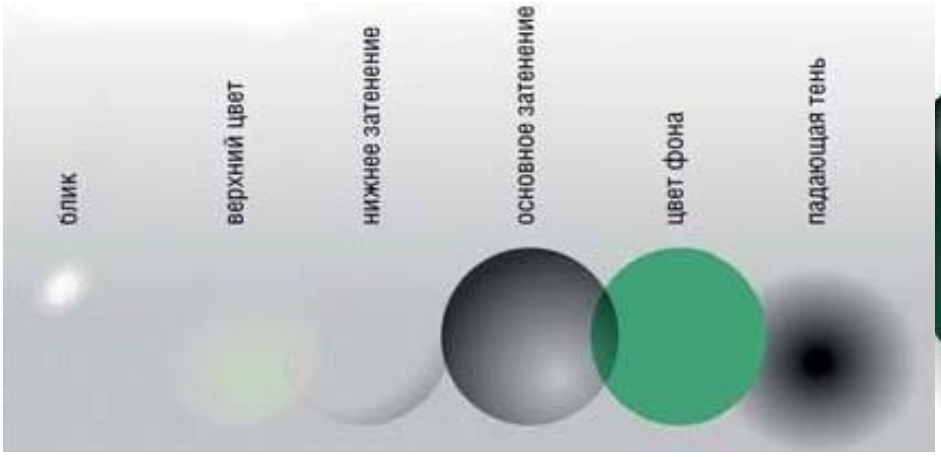
9. Добавьте опорную точку, укажите ее цвет – оранжевый и смещение.

10. Добавляйте точки в градиент и изменяйте их свойства, чтобы в градиенте появились все цвета радуги. См. рис. 1.

Рис. 1

Задание 2. Создание цветной капли

1. Создайте шесть окружностей с эллиптическими градиентами и различной прозрачностью. См. рис. 2.



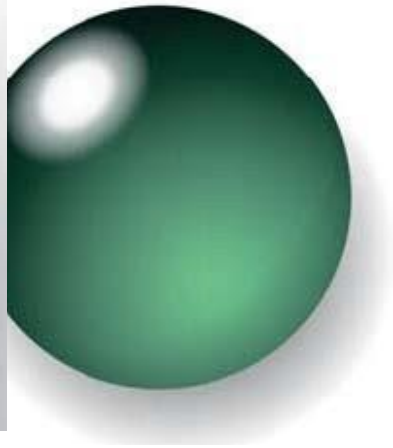


Рис. 2 Рис.3

2. Совместите окружности и разместите их друг над другом (Объект Þ Поднять, Опустить) как показано на рис. 3.

Задание 3. Заверстайте текст в фигуру

1. Создайте текстовый объект (рис. 4а), фигуру (рис. 4в).
2. Выделите оба объекта, выполните команду: Текст Þ Заверстать текст в блок (рис. 4с).
3. Самостоятельно создайте фигуру (рис. 4d).

Рис. 4

Задание 4. Создание визитной карточки в стиле конструктивизма

1. Создайте новый документ, выбрав шаблон под названием «Business Card 90x50mm» в меню Файл Þ Новый.

2. С помощью инструмента Текст создайте текстовые объекты для каждого элемента – имя, должность, адрес, номер телефона и т.д. Все они должны быть независимыми объектами, потому что их придется двигать и трансформировать. См. рис. 5.
3. Выберите шрифт (шрифты, но не более двух) для ваших текстовых объектов и измените их размеры (Текст Þ Текст и шрифт). См. рис. 6.

Рис. 5

Рис. 6

4. Разместите компоненты адреса вокруг своего имени, выровняв их по нескольким невидимым линиям. См. рис. 7.
5. Выделите все компоненты и немного поверните, используя инструмент Выделение и трансформация объектов. Добавьте в макет карточки три черных уголка и большой красный круг в центре. См. рис. 8.

Рис. 7

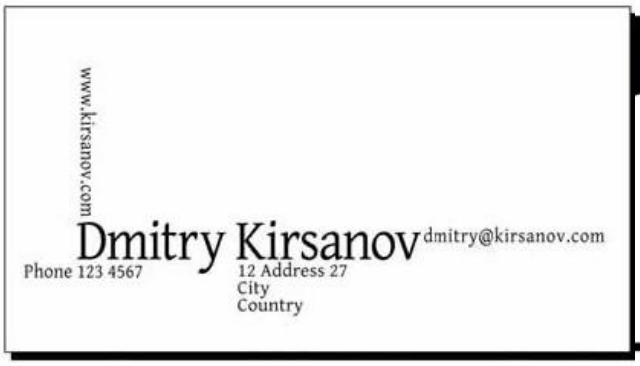


Рис. 8

6. Сохраните результат в формате PS или EPS. В диалоговом окне параметров EPS включите параметры Convert Text To Path (Преобразовать текст в контур) и Make Bounding Box Around Full Page (Создать рамку вокруг страницы).

Задание 5. Создание визитной карточки со стилизованными инициалами

1. Создайте новый документ, выбрав шаблон под названием «Business Card 90x50mm» в меню Файл Þ Новый.
2. Инструментом Каллиграфическое перо (угол 90, фиксация 1.0) нарисуйте

вензель из инициалов. См. рис. 9.

3. Чтобы сгладить рисунок, объедите все штрихи в один контур (Контуры Þ Объединить) и несколько раз примените упрощение (Контуры Þ Упростить), втяжку (Контуры Þ Втянуть) и растяжку (Контуры Þ Вытянуть). См. рис. 10.

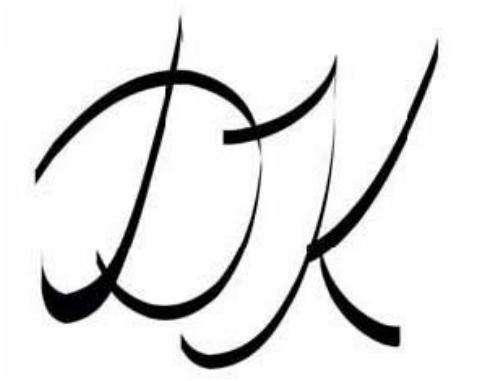


Рис. 9

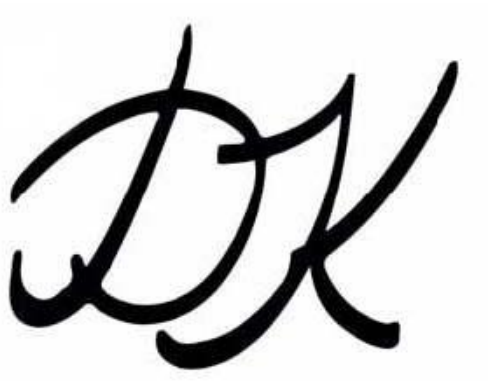


Рис. 10

4. С помощью тонкого пера с максимальным параметром Дрожание нарисуйте штрихи вокруг вензеля. См. рис. 11.
5. Объедините полученную фигуру, примените команды Упрощение, Втянуть, Вытянуть. См. рис. 12.

Рис. 11

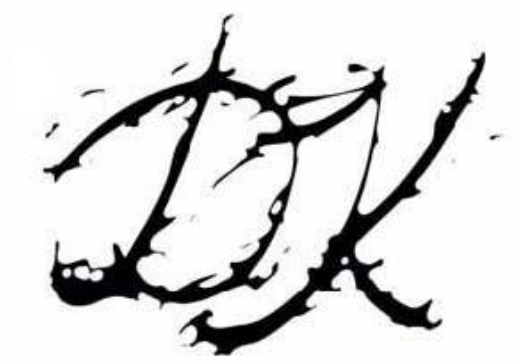


Рис. 12

6. С помощью инструмента Текст создайте независимые текстовые объекты для каждого элемента – имя, должность, адрес, номер телефона и т.д. Разместите текстовые объекты как показано на рис. 13, 14.

Рис. 13

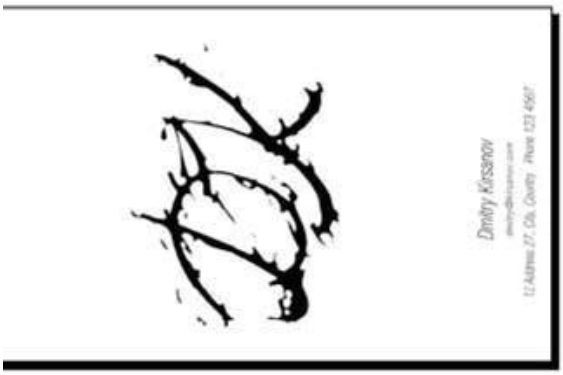


Рис. 14

7. Добавьте четыре несимметричных прямоугольника по краям карточки. С помощью инструмента Градиент создайте в прямоугольниках цветовые переход из любого цвета в белый. Результат см. на рис. 15.

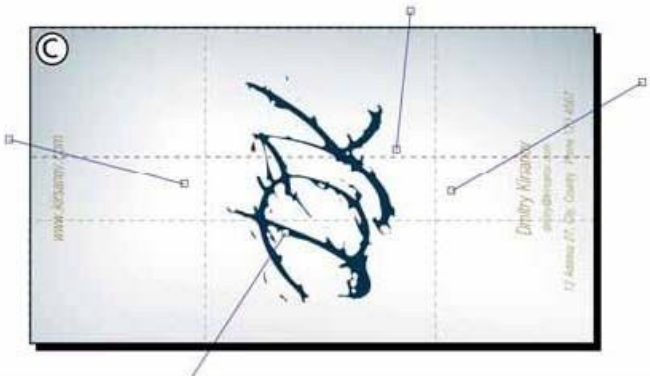


Рис. 15

8. Создайте сетку из полупрозрачных линий: начертите узкий горизонтальный прямоугольник белого цвета, выполните команду Правка Þ Клоны Þ Создать узор из клонов, выберите симметрию P1, смещение по Y на 100%, укажите количество строк, столбцов 100 x 1.
9. Сгруппируйте прямоугольники и поместите их поверх градиента, но под вензелем и текстом (Объект Þ Поднять, Опустить). Отрегулируйте прозрачность прямоугольников.
10. Экспортируйте визитную карточку в растровый формат PNG.

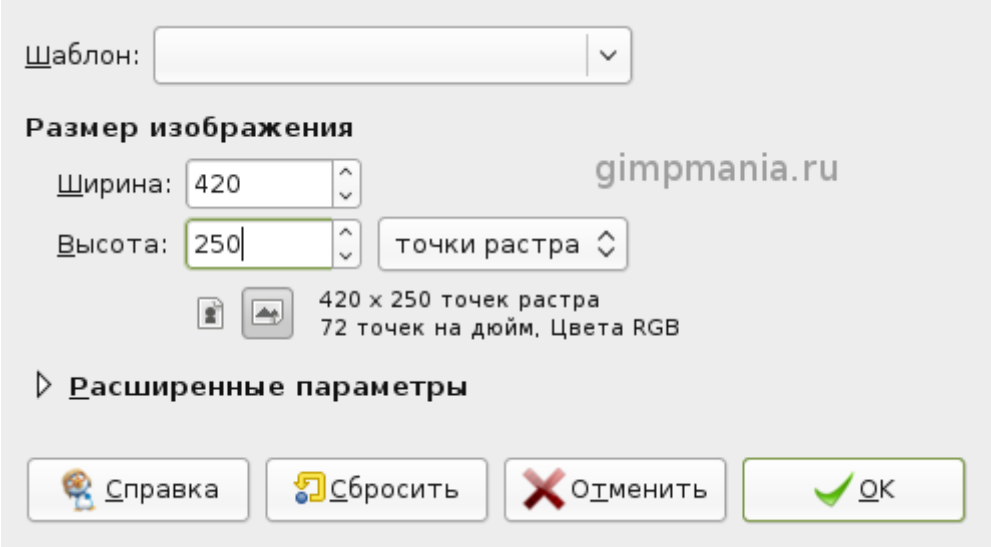
Практическое занятие №21 Основы анимации

Задание1: средствами GIMP повторить рисунки по предложенным примерам или проявить фантазию.

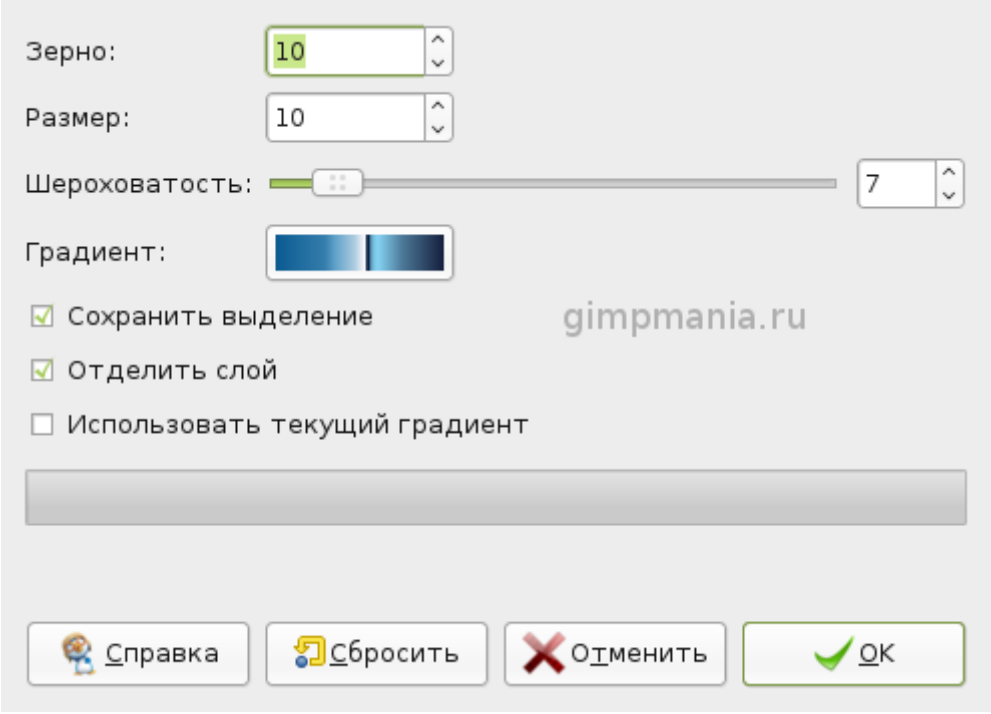
Пример 1: Анимация воды

В этом уроке мы будем делать анимацию воды.

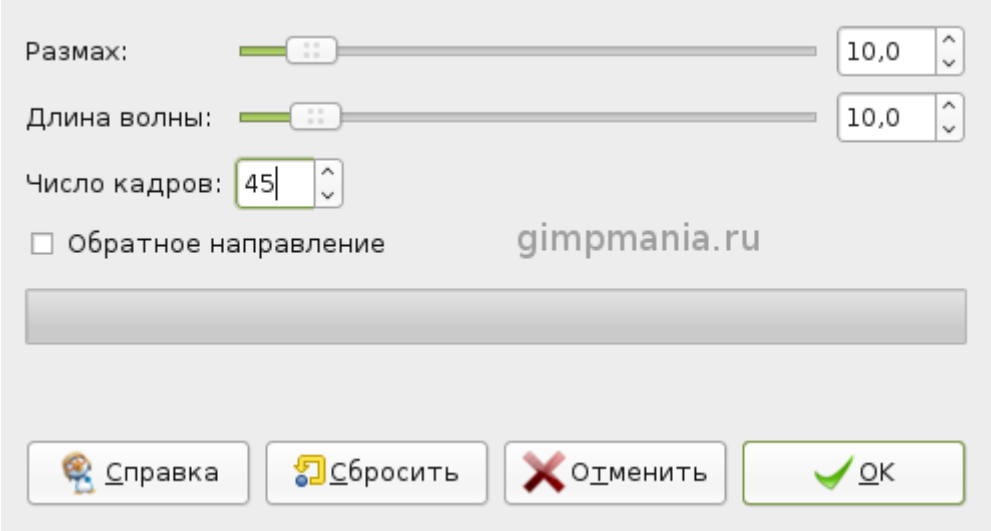
Шаг1.
Создаём изображение произвольного размера, я выбрал 420*250



Шаг2.
Идём в Фильтры > Визуализация > Лава. Оставляем всё по умолчанию кроме градиента, он должен
быть "Horizon 2"



Шаг3.
Сейчас можно делать анимацию. Для начала нужно удалить ваш фоновый слой и выбрать залитый градиентом. Потом идём в Фильтры > Анимация > Волны. И выставляем такие параметры (хотя можно подобрать свои):

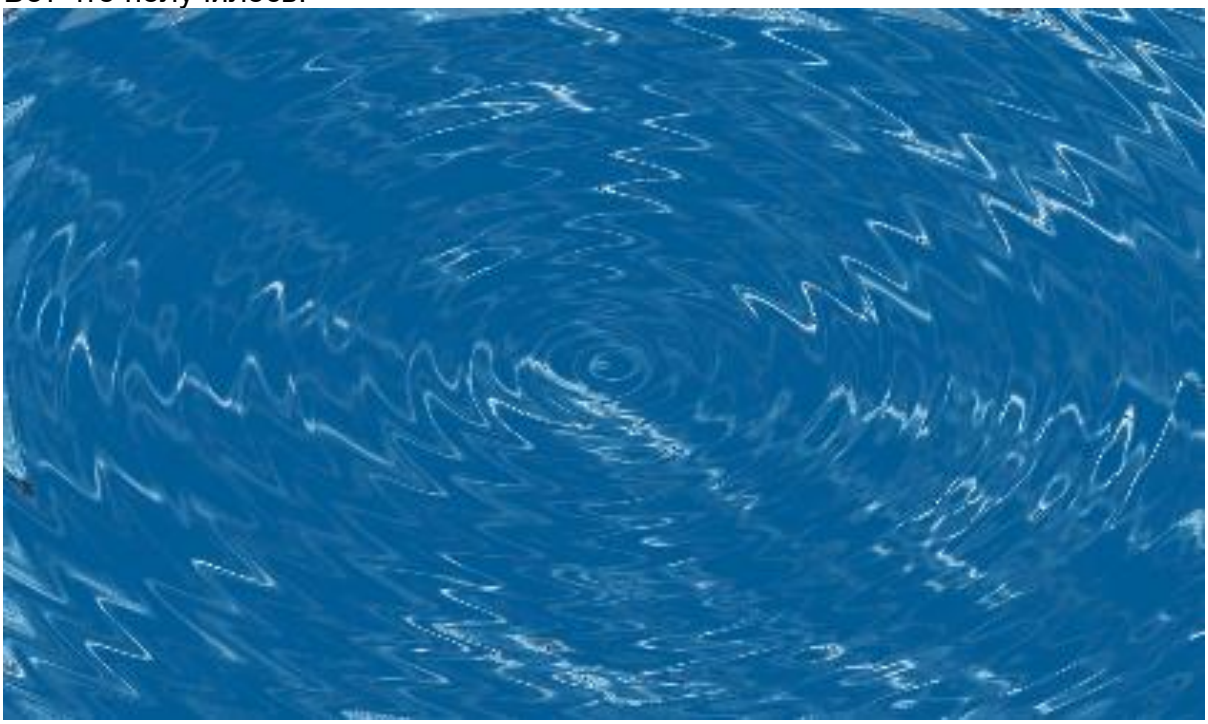


Заметьте что чем больше кадров, тем лучше качество отображения, но и размер картинки тоже намного увеличивается.

Шаг.4

Идём в Файл > Сохранить как. Тип GIF, сохранить как анимацию.

Вот что получилось:



Задание 2: Создать анимационный баннер средствами редактора растровой графики Gimp.

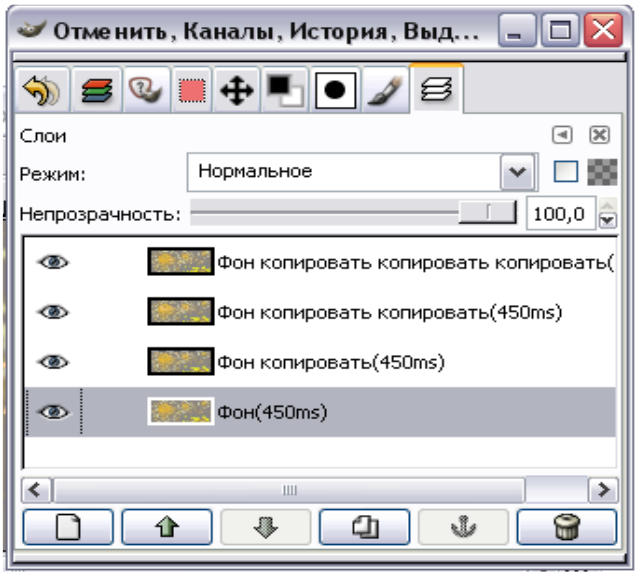
Пример 2. Создание анимированного баннера-логотипа

Шаг 1

Создадим баннер, состоящий из трех анимированных строк текста. Каждый кадр будет состоять из изображения цветка и некоторого текста, в итоге получим анимацию из трех кадров. Используя кнопку *Создать копию слоя* в диалоге слоев, дважды сделаем копии нашего изображения. После этого, используя стандартный инструмент *Добавить текст к изображению* добавляю небольшой текст к каждому кадру. После помещения текста в кадр, GIMP создаст выделенную область, позиционирую ее справа, используя инструмент *Перемещение* и прикрепляю (ставим якорь) используя комбинацию *Ctrl+H*. Перед добавлением текста установливаю прозрачность слоя (кадра) в более низкое значение так, чтобы я могла видеть слой, находящийся перед ним. Таким образом, я могу позиционировать новый текст относительно предыдущего (того, который находится в предыдущем кадре).

Шаг 2

Установим задержку, указыв её в названии слоя. В названии слоя укажем временной интервал в ms (450ms), заключенный в круглые скобки. Проверить правильность определения времени можно используя предварительный просмотр анимации - <Изображение> Фильтры > Анимация > Воспроизведение.



Шаг 3

Теперь осталось сохранить анимацию в формате GIF. Используем оптимизацию Анимация > Оптимизация использует режим combine. Использю <Изображение> Изображение > Режим > Индексированное (*Alt+I*). Затем сохраняем в формате GIF.

Шаг 4

Создадим логотип сайта. Создаём надпись с фоновым рисунком, это будет основа нашего банера. Рис1.

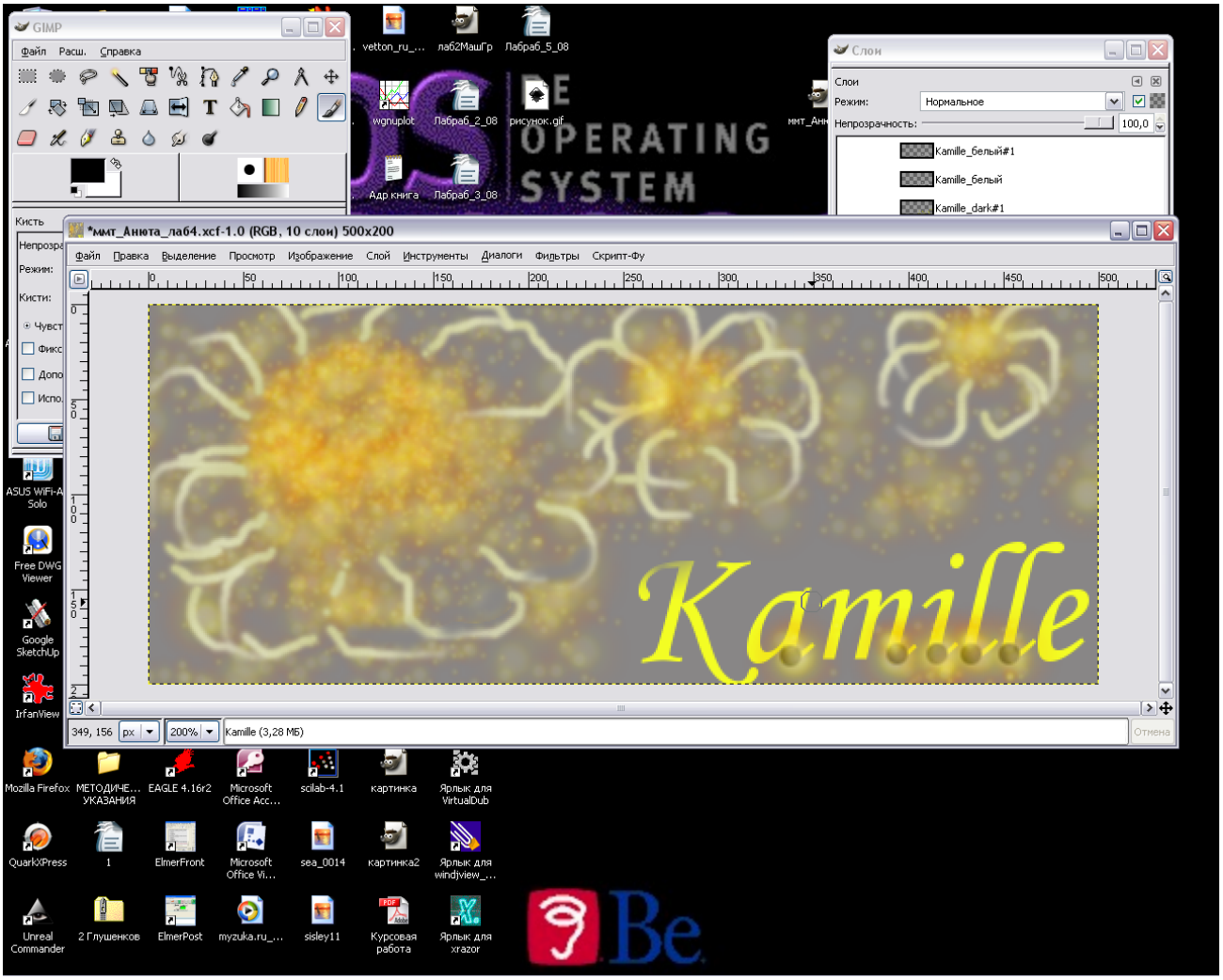


Рис1.

Шаг 5

Дважды создайте копию слоя с текстом. Одна для создания эффекта осветления, другая для эффекта затенения. Слой с осветлением должен быть белым, так что выбираем его, включаем переключатель “Сохранять прозрачность” в диалоге “Слои, Каналы, Контуры” и заливаем белым цветом. Это легко делается перетаскиванием белого цвета из Панели инструментов (Toolbox) на изображение (при условии, что цветной слой в данный момент выделен) (такого же эффекта можно достичь перетаскиванием цвета сразу на слой в диалоге “Слои, Каналы, Контуры” - прим.). “Сохранять прозрачность” гарантирует нам, что все трансформации или заполнения этого слоя будут применяться только к непрозрачной части изображения.

Шаг 6

Создайте копию текстового слоя еще раз и переместите ее на самый верх стека слоев (используйте маленькую стрелку вверх внизу диалога “Слои, Каналы, Контуры”). Убедитесь, что “Сохранять прозрачность” включено и примените к этому слою “размывание”. Можно использовать “Гауссово размывание” (Gaussian Blur, -

прим.) (RLE) в 10px. Когда создаёте оригинальный логотип, оно было установлено в меньшее значение для того, чтобы сделать эффект менее округлым. Вы должны удостовериться, что граница слоя больше чем слой, тогда “размывание” ложится красиво. Это упоминается во втором Действии. Сделайте две копии “размытого” слоя (tmp1 и tmp2). Эти слои используются для создания светлой границы текста.Переместите один из “размытых” слоев вниз и вправо на 5 пикселей. Точное количество пикселей зависит от того, как было проделано “размытие” и от того, как размытие должно располагаться вокруг текста, словом, выбирайте на свой вкус. Вы можете перемещать слои используя инструмент “Перемещение слоев и выделенных областей” (Move) и клавиши-стрелки на вашей клавиатуре. Неважно, какой из “размытых” слоев вы использовали на данном Действии, но другой вы не должны перемещать. Теперь создадим выделенную область, используя “размытый” слой (который мы двигали), используя “Альфа-канал в выделенную область” (Alpha to Selection). Это точное выделение слоя, включающее информацию о прозрачности. Полезная вещь. Лучше выключить другой “размытый” слой, это делается для того, чтобы видеть, как далеко перемещается слой. Вырежьте (Правка/Вырезать или *CTRL+X* на PC или *Command+X* на Mac). Это уберет выделенную область “размытого” слоя из осветленного слоя. Вы можете использовать маски каналов. Следующее Действие, – сделайте тоже с другим “размытым” слоем. В итоге мы получим чтото похожее на изоброжение на рис2.

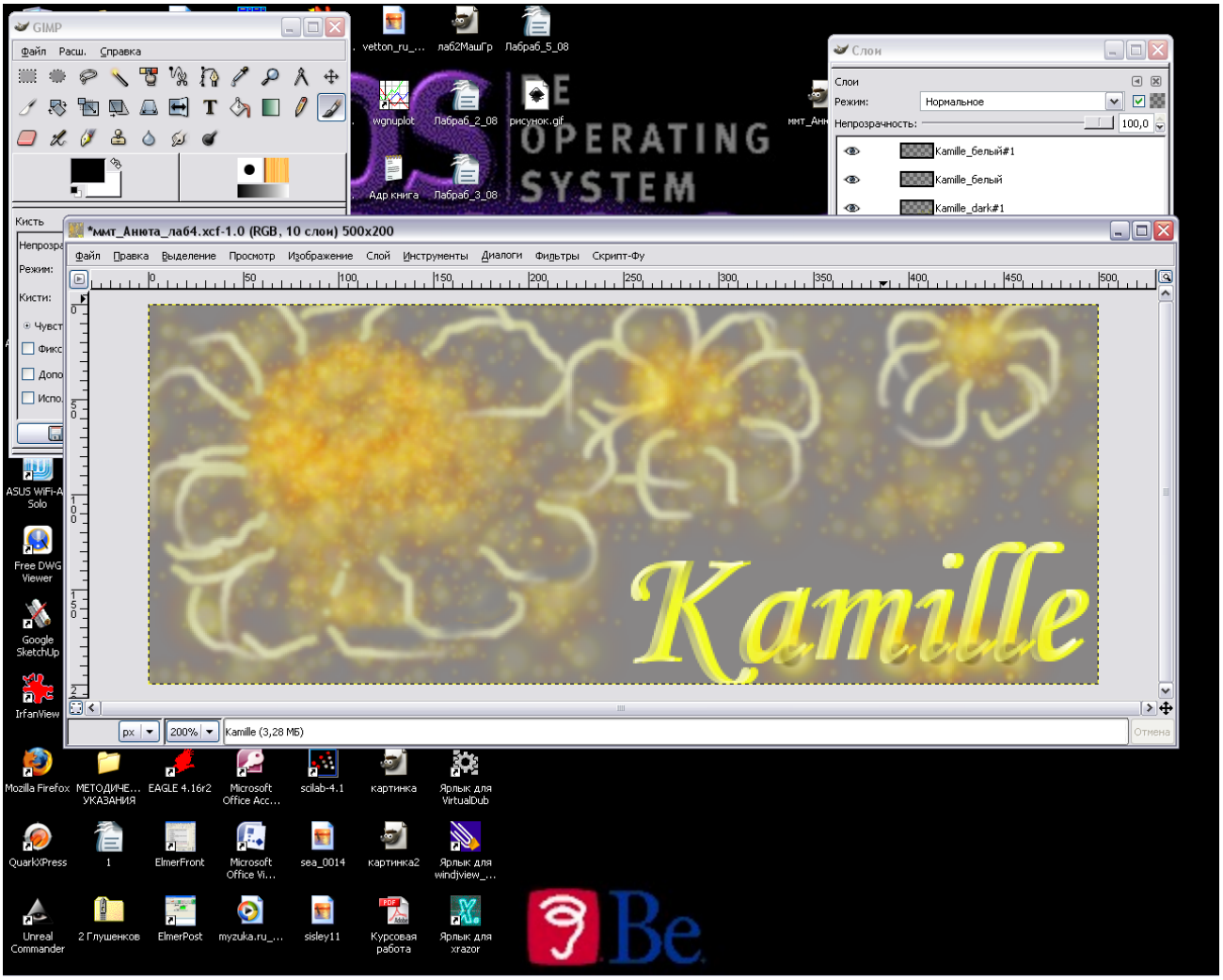


рис 2.

Шаг 7

Затем создадим копию исходного слоя надписи и применим к ней размытее. Рис3.

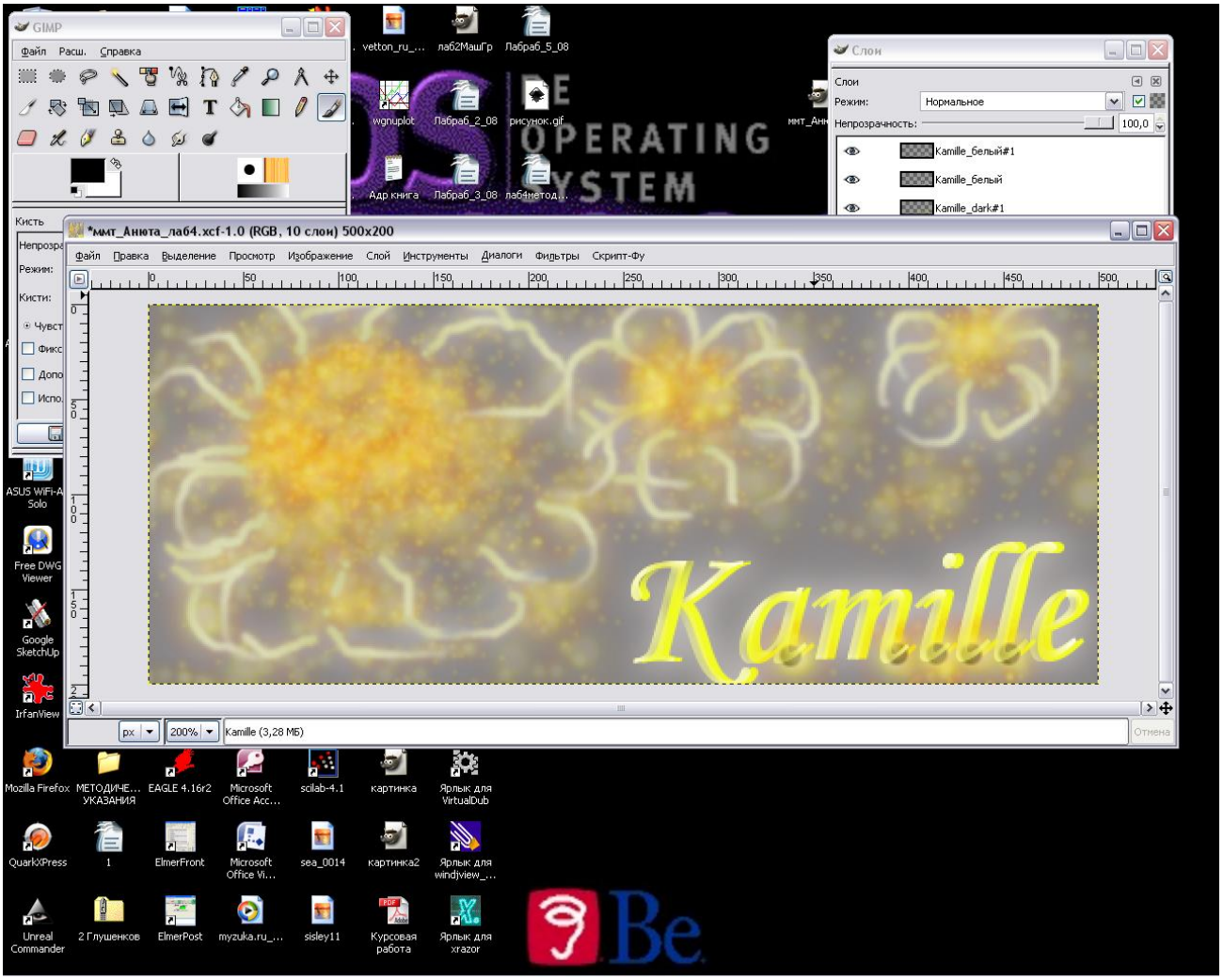


рис3.

Шаг 8

Затем, создаём новый прозрачный слой, назовем его rust, и выберите инструмент “Создание и редактирование контуров” (bezier tool). Создаём подобное выделение. Рис4.

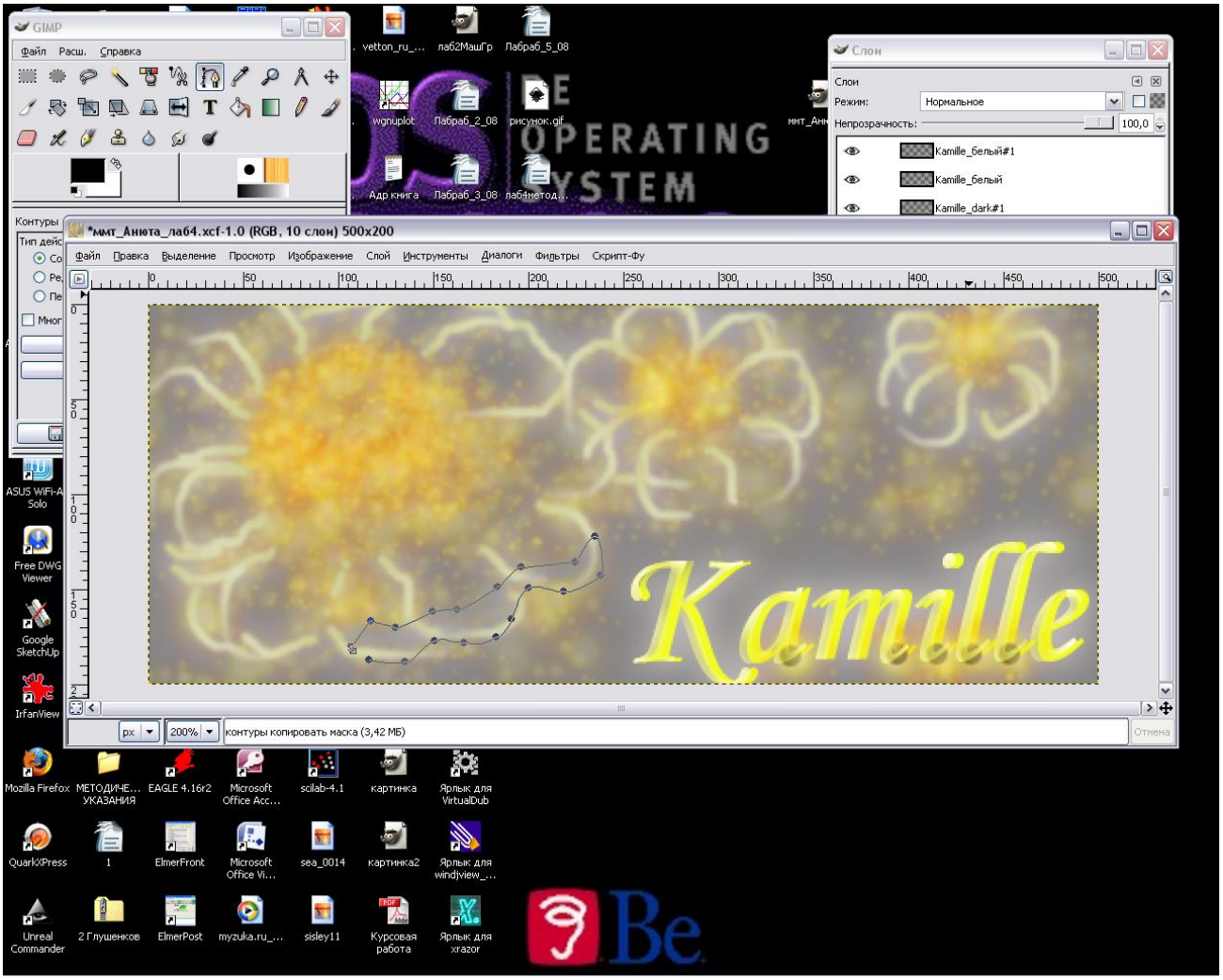


рис4.

Шаг 9

Создайте из контура выделенную область. Используйте Выделение → Растушевка (<Image> Select → Feather, - ориг.), чтобы размыть (округлить) выделенную область. Использовалось значение 10 пикселей. Выбираем понравившыйся нам цвет для слоя rust. Заливаем выделенную область этим цветом. После этого делаем активный слой rust. Инвертируем выделение (*CTRL+I* или *Command+I*) и вырезаем. Это должно оставить только ту часть слоя rust, которая перекрывает буквы и ничего более. Применяем размытие по собственному усматрению. Измените режим слоя rust на умножение (Muliply). Добавляем маску слоя rust (он должен быть активным). Проверьте, что вы сбросили цвета на панели инструментов, используйте инструмент “Градиент” на маску слоя. Вы получите что-то, похожее на это. Получем чтото похожее на рис5.

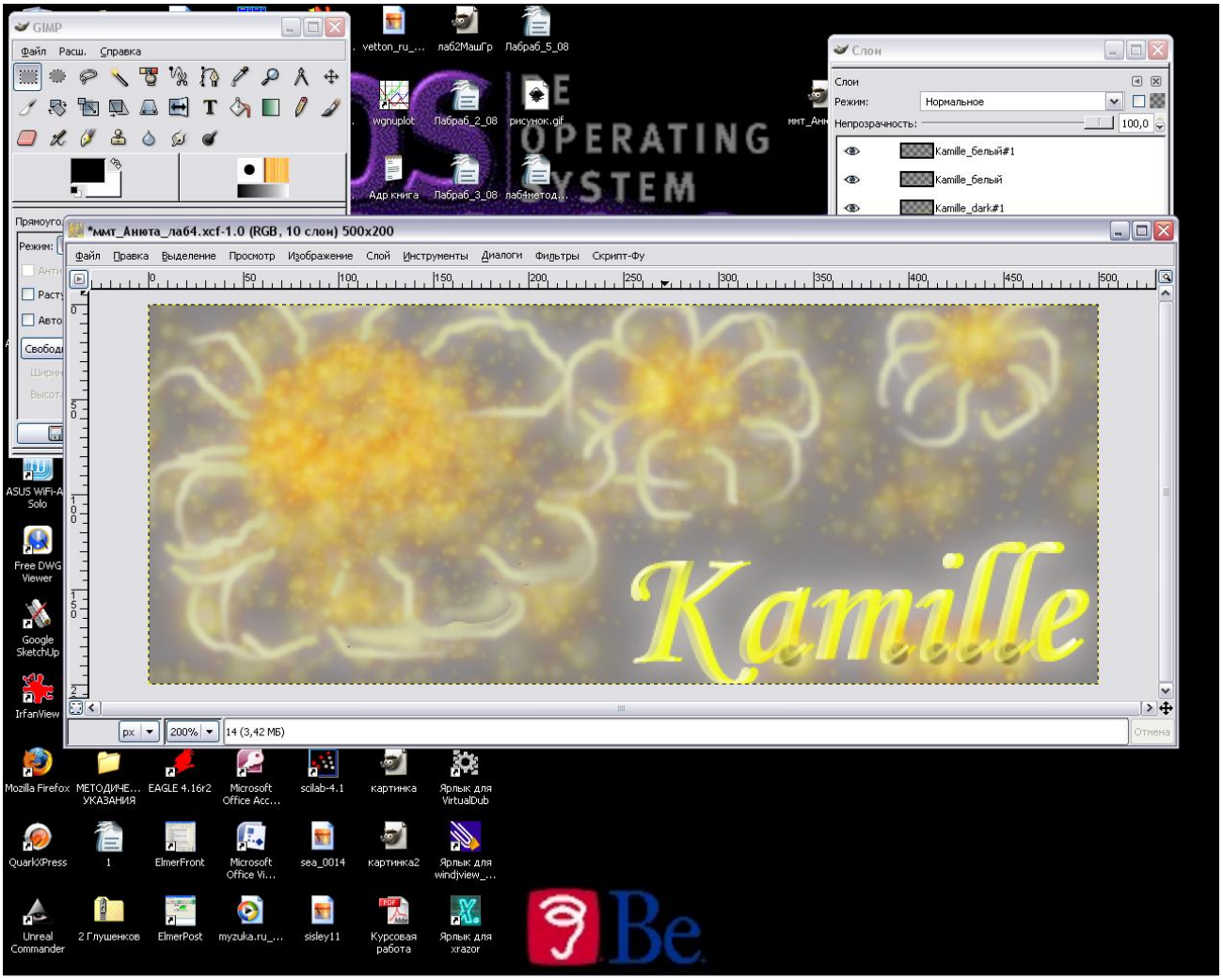


рис5.

Шаг 10

Создайте новый прозрачный слой, назовем его outline. Перемещайте его вниз в стеке слоев, пока он не окажется под слоем с оригинальным текстом. Выберите оригинальный текстовый слой (цвет которого вы изменяли) и повторите “Альфа-канал в выделенную область” снова. Нажмите правой кнопкой на изображении, выберите “Выделение” →“Увеличение”. Это сделает новую область выделения больше на указанное количество пикселей. Я использовал значение в 4 пикселя. Теперь, когда у вас есть выделенная область, переключитесь на слой outline, чтобы можно было залить его. Вы увидите это ниже.Залейте слой outline нужным нам цветом и отмените выделение (*CTRL+SHIFT+A* на PC или *Command+SHIFT+A* на Mac). Как вы можете видеть, теперь у изображения более контрастный фон. Рис6.

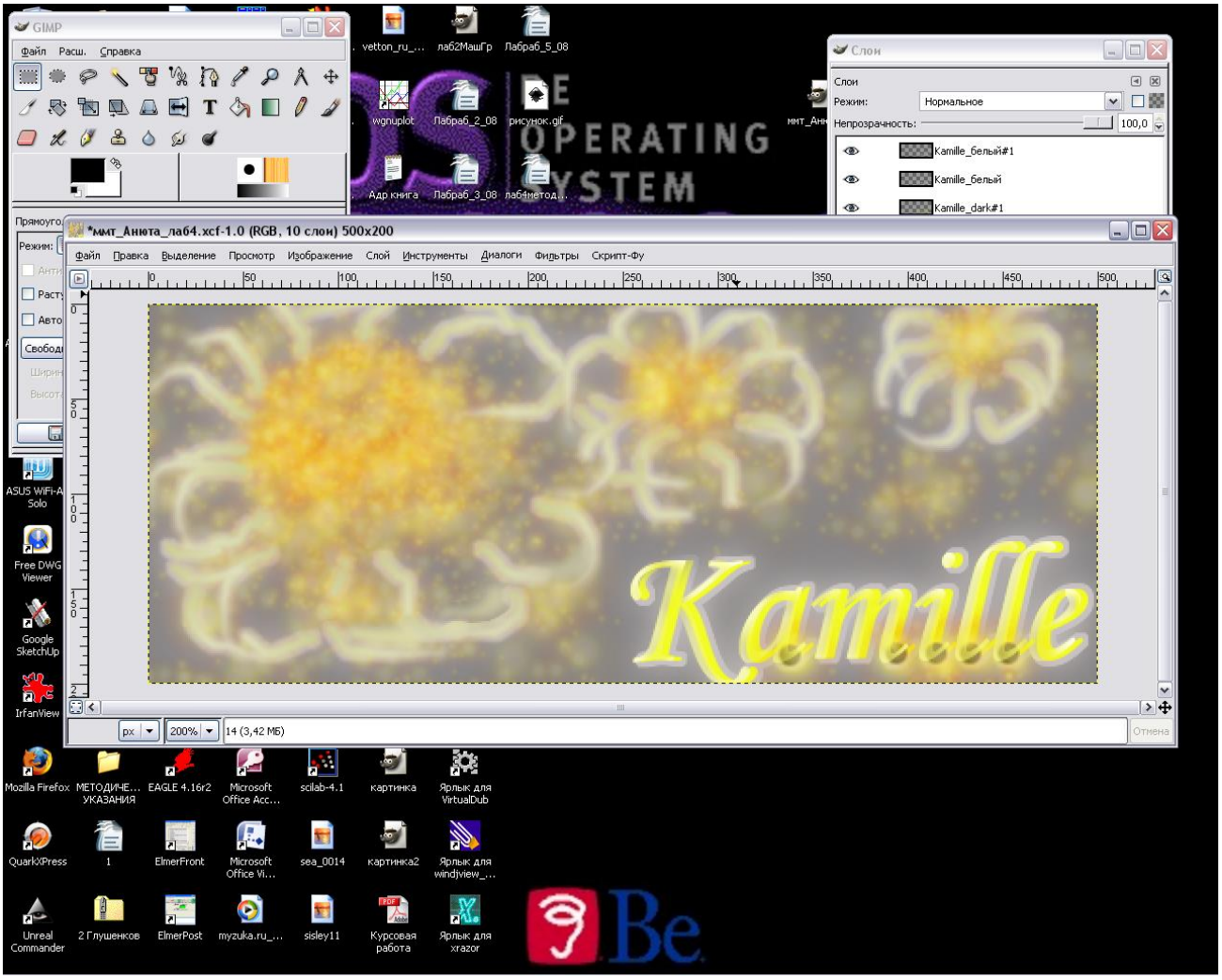
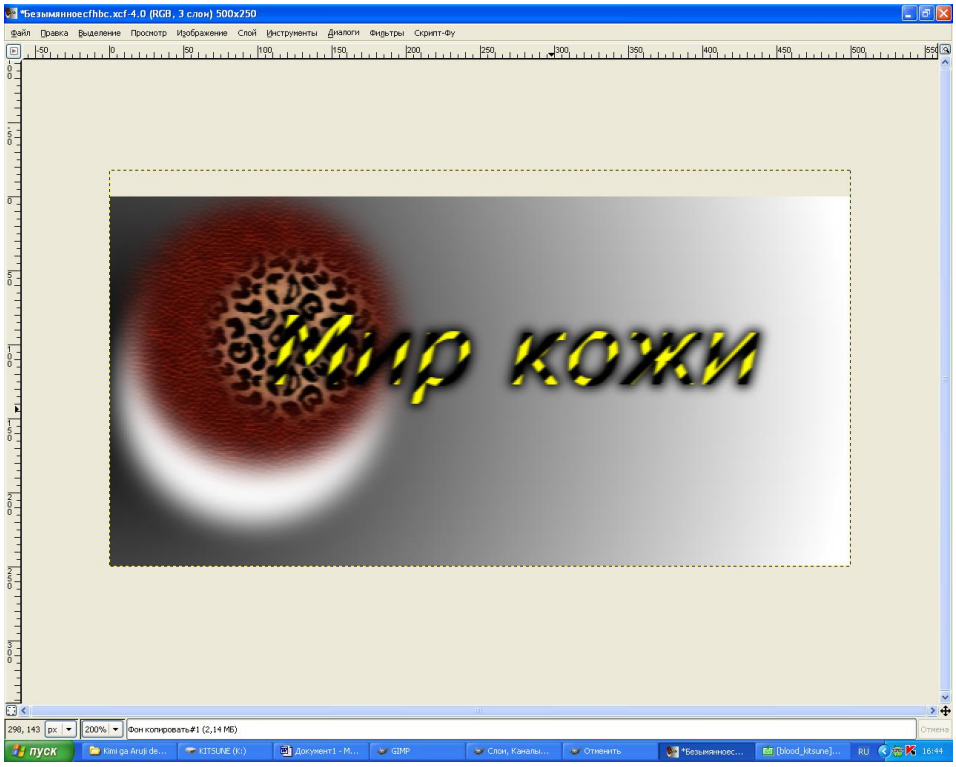


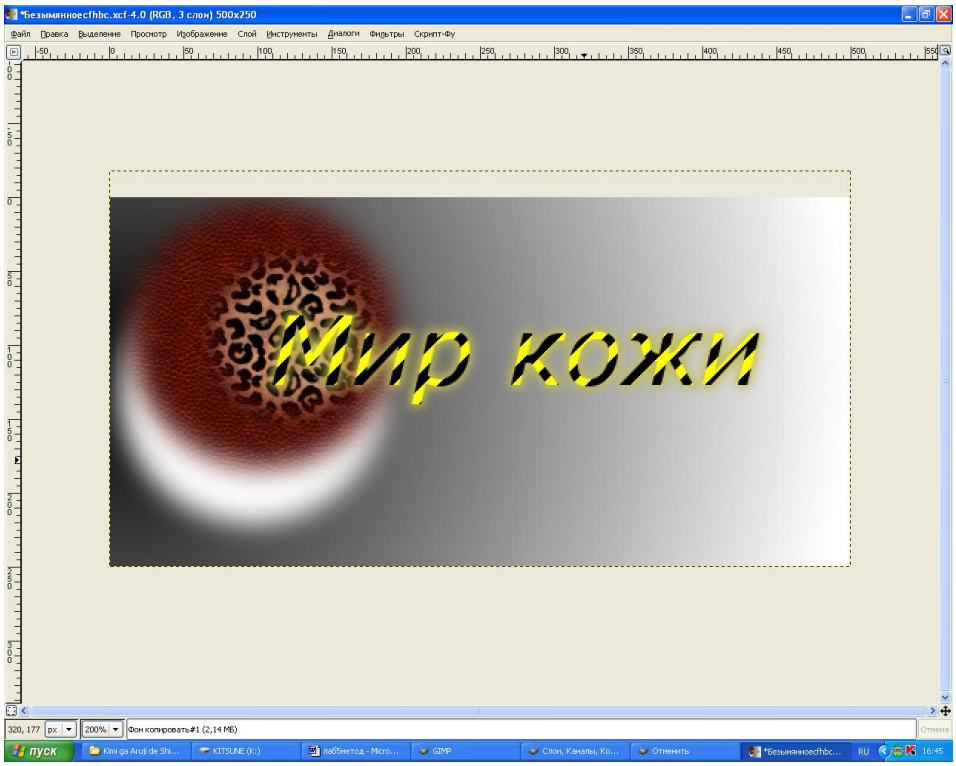
рис6.

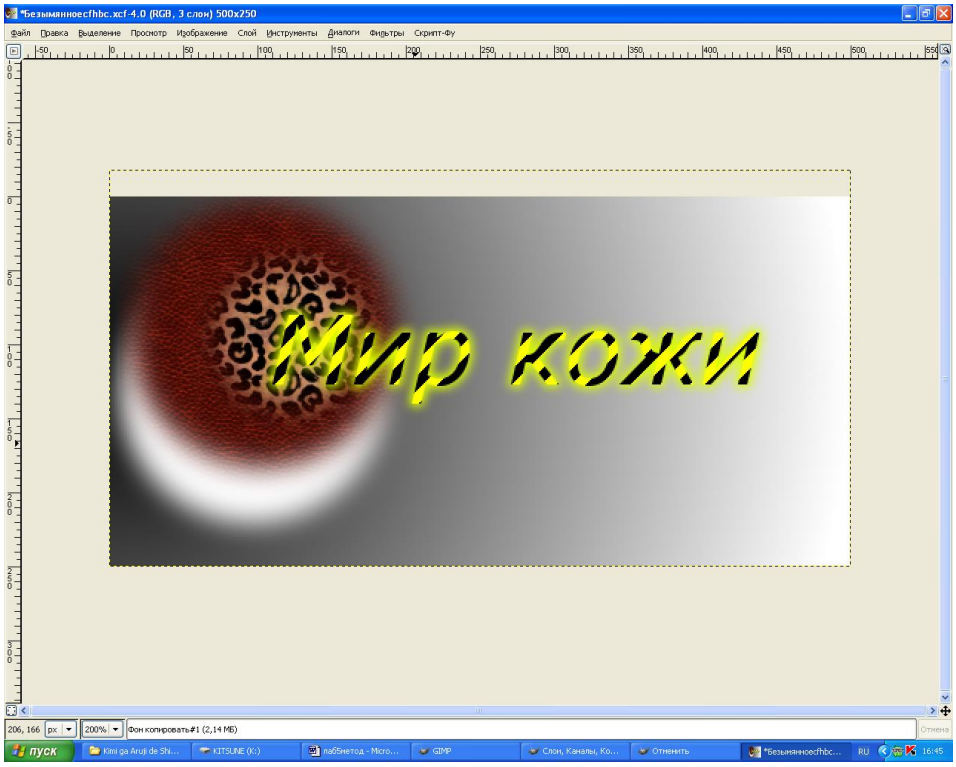
Ну вот логотип и готов.=)

Пример 3. Рисование баннера

Сейчас мы рассмотрим как создать анимационный баннер. Для начала создадим слой и нарисуем на нём фон баннера. Создадим ещё 2 копии этого слоя. Затем создадим надпись которую мы будем анимировать. Так же создадим ещё 3 копии этого слоя. В каждом из этих слоёв поменяем цвет надписи или цвет его тени. Затем расположим слои попарно слой надписи, слой фона и затем попарно объединим их. Получаем 3 слоя:


*Безымянноесfhbc.xcf-4.0 (RGB, 3 слои) 500x250
Мир кожи


*Безымянноесfhbc.xcf-4.0 (RGB, 3 слои) 500x250
Мир кожи



Затем укажем время задержки каждого кадра-слоя. Для этого в конце названия добавим в круглых скобочках время задержки, например: …(1000ms). Время указывается в миллисекундах.

Теперь осталось сохранить анимацию в формате GIF. Но перед этим надо оптимизировать изображение, используя <Изображение> Фильтры → Анимация → Оптимизация. Вы можете создавать изображение, используя два разных режима для каждого кадра: режим замена (replace) (используется по умолчанию, старый кадр заменяется новым) или режим составление (combine), когда каждый новый кадр добавляется к предыдущему (тогда обновляются только изменения). Анимация → Оптимизация использует режим combine, что делает размер изображения существенно меньше. Теперь нам нужно сделать изображение индексированным используя <Изображение> Изображение → Режим → Индексированное (*Alt+I*). Пытайтесь использовать как можно меньше цветов и избегать смешивания цветов. Оба этих фактора очень сильно увеличивают размер файла. В примере не использованы смешивания цветов и сгенерирована 32- цветная палитра. Размер баннера получился около 7 килобайт. Когда вы будете сохранять изображение в формате GIF, GIMP предложит сохранить его как анимацию, чего мы собственно и добивались. В диалоге сохранения вы можете задать временную задержку между кадрами по умолчанию (в нашем случае это будет использованно для кадров с размытым текстом) и режим отображения кадров (combine или replace). Для проверки баннера вы можете использовать свой браузер или воспользоваться функцией Воспроизведение в разделе Анимация в меню изображения.


МИР КОЖИ