

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Е. Ю. Заболотнова
С. А. Калинина

ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА

Учебно-методическое пособие
для студентов, обучающихся в бакалавриате по направлению подготовки
09.03.03 – Прикладная информатика
(профиль «Прикладная информатика в экономике»)
на очной и заочной формах обучения

Калининград
2023

УДК 004.9(075)

Рецензент:
кандидат экономических наук,
и. о. заведующего кафедрой прикладной информатики ФГБОУ ВО
«Калининградский государственный технический университет»
М. В. Соловей

Заболотнова, Е. Ю.

Ознакомительная практика : учеб.-метод. пособие для студентов, обучающихся на бакалавриате по направлению подготовки 09.03.03 – Прикладная информатика (профиль «Прикладная информатика в экономике») на очной и заочной формах обучения / **Е. Ю. Заболотнова, С. А. Калинина.** – Калининград : Изд-во ФГБОУ ВО «КГТУ», 2023. – 26 с.

В учебно-методическом пособии приведены задания и содержание отчетов по ознакомительной практике для студентов очной и заочной форм обучения.

Табл. 6, рис. 5, список лит. – 4 наименования

Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы «Учебная практика – ознакомительная» направления подготовки 09.03.03 – Прикладная информатика.

Учебно-методическое пособие рассмотрено и одобрено в качестве локального электронного методического материала кафедрой прикладной информатики Института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 19 сентября 2022 г., протокол № 3.

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 17 января 2023 г., протокол № 11.

© Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Калининградский государственный
технический университет», 2023 г.

© Заболотнова Е. Ю., Калинина С. А., 2023 г.

Оглавление

Введение	4
Задание на ознакомительную практику для студентов очной формы обучения.	5
Порядок оформления отчета и защита ознакомительной практики для студентов очной формы обучения	8
Задание на ознакомительную практику для студентов заочной формы обучения.....	9
Порядок оформления отчета и защита ознакомительной практики для студентов заочной формы обучения	20
Список литературы.....	22
Приложение 1	23
Приложение 2	24

Введение

Целью учебной практики является закрепление и расширение теоретических знаний и практических навыков, получаемых студентами в ходе учебного процесса. После первого курса практика нацелена на освоение студентами основ программирования на алгоритмическом языке Python, закрепление навыков работы с файлами, разработку программ на основе графического интерфейса и методов и принципов объектно-ориентированного программирования.

Учебная – ознакомительная практика входит в состав обязательной части основной профессиональной образовательной программы бакалавриата и проводится после теоретического обучения и экзаменационной сессии во втором семестре. Трудоемкость учебной – ознакомительной практики составляет 3 зачетные единицы (ЗЕТ), 108 академических часов (81 астр. часа) контактной работы, продолжительность практики – 2 недели. Форма аттестации по практикам – дифференцированный зачет (зачет с оценкой).

Задание на ознакомительную практику для студентов очной формы обучения

Написать программу для прохождения тестирования по теоретическому материалу дисциплины «Программирование». Тема для тестирования совпадает с темой из первой части курсовой работы студента:

1. Арифметические операции в Python. Операции над целыми и вещественными числами.
2. Оператор присваивания в Python. Ввод данных.
3. Вывод данных. Форматы вывода.
4. Функции в Python. Глобальные и локальные переменные. Создание функции.
5. Логический тип данных в Python. Операции отношения. Логические операции.
6. Условный оператор в Python.
7. Циклы с заданным числом повторений и циклы по условию.
8. Списки в Python. Операции со списками и основные методы списков.
9. Списки списков в Python (матрицы). Создание и обработка.
10. Строки в Python. Операции со строками и основные методы строк.
11. Кортежи в Python. Операции с кортежами и методы кортежей.
12. Словари в Python. Операции со словарями и методы словарей.
13. Работа с файлами.
14. Модуль graph. Создание графических примитивов.
15. Модуль graph. Создание анимированных изображений.
16. Понятие ООП. Основные принципы ООП.
17. Описание классов. Атрибуты и методы класса. Экземпляры класса.
18. Реализация принципов ООП в Python.
19. Событийно-ориентированное программирование. Программы с графическим интерфейсом.
20. Модуль tkinter. Создание компонентов (виджетов): кнопка, метка (надпись), текст однострочный и многострочный. Свойства и методы этих виджетов.
21. Модуль tkinter. Создание компонентов (виджетов): рамка, флажки и радиокнопки, списки, шкала, окно верхнего уровня. Свойства и методы этих виджетов.
22. Модуль tkinter. Менеджеры геометрии.
23. Модуль tkinter. Метод bind. Типы переменных и события.
24. Исключения. Обработка исключений.

Тест должен состоять из 10 вопросов для выбора одного (или нескольких) правильных ответов. Количество предложенных вариантов ответа на вопрос должно быть не менее трех. Формулировки вопросов и ответов на них хранятся в тестовом файле.

Вопросы и варианты ответов студент придумывает самостоятельно.

Не допускается, чтобы верными были все первые (вторые или третьи) ответы в тесте. Возможен случайный выбор вопросов из файла и случайное расположение вариантов ответов на экране. По желанию студента можно ввести временное ограничение на выполнение каждого вопроса или всего теста полностью.

Программа работает с пользовательским графическим интерфейсом (GUI), который должен содержать следующие элементы, в скобках указано название виджета для реализации:

Название дисциплины: Программирование (метка);

Название темы: смотри первую часть своей курсовой работы (метка);

Кнопки для начала и завершения тестирования;

Номер и текст вопроса (метка или текстовое поле);

Варианты ответов (метки или текстовые поля);

Выбор правильного ответа (ответов) можно реализовать через радиокнопки или флажки или путем ввода номера правильного ответа в текстовое поле.

По окончании теста программа должна вывести правильные ответы и ответы студента, возможно обозначение цветом или иным способом правильных и неправильных ответов студента, подсчитать количество правильных ответов и набранный балл по результатам теста (1 балл за один верный ответ).

Дизайн макета GUI определяется студентом самостоятельно. Ниже приведены примеры реализации программ тестирования.

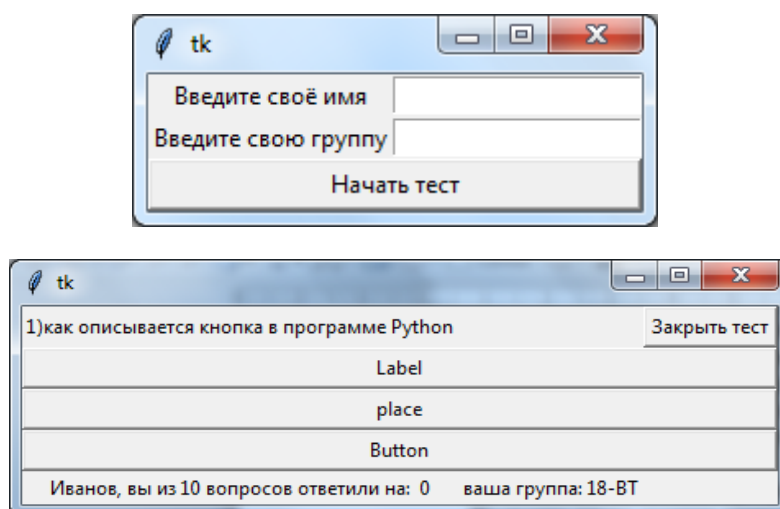


Рисунок 1. Пример работы Программы 1

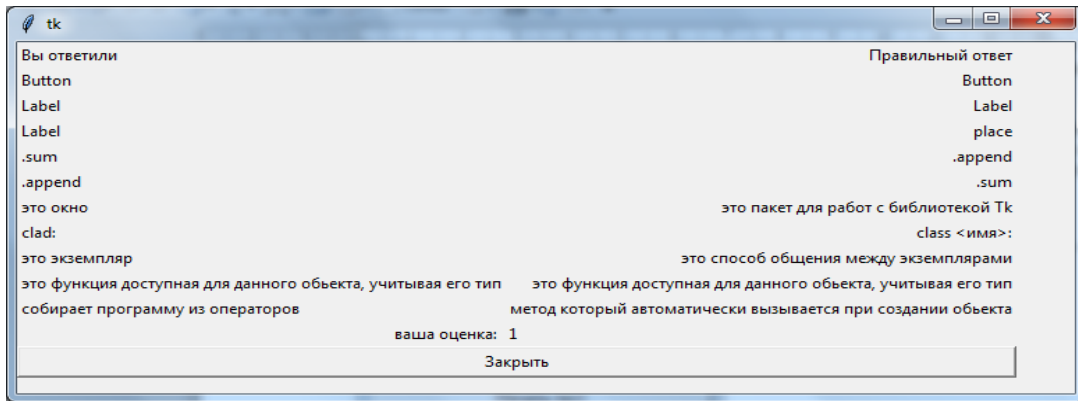


Рисунок 1. Продолжение

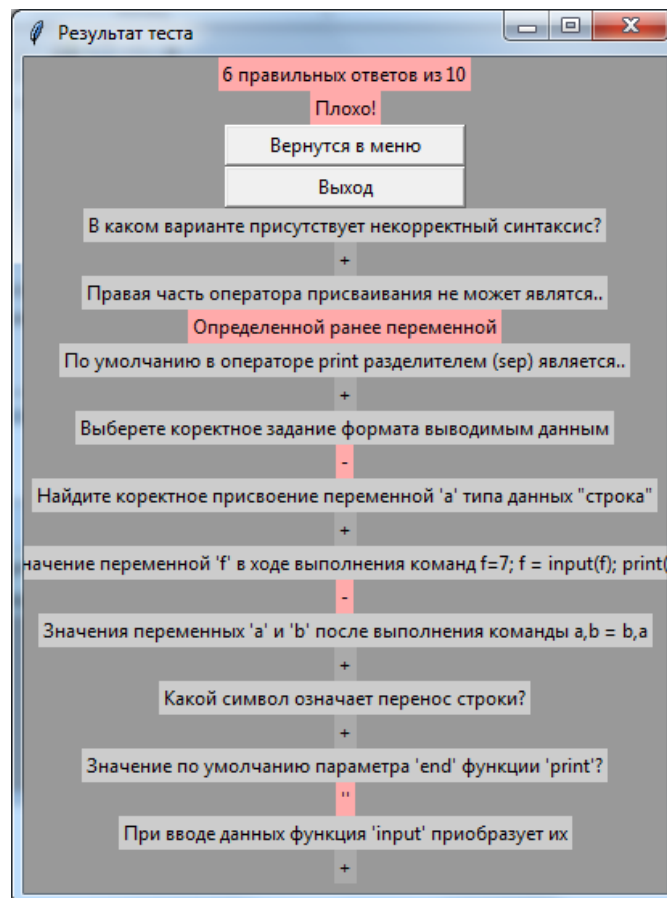
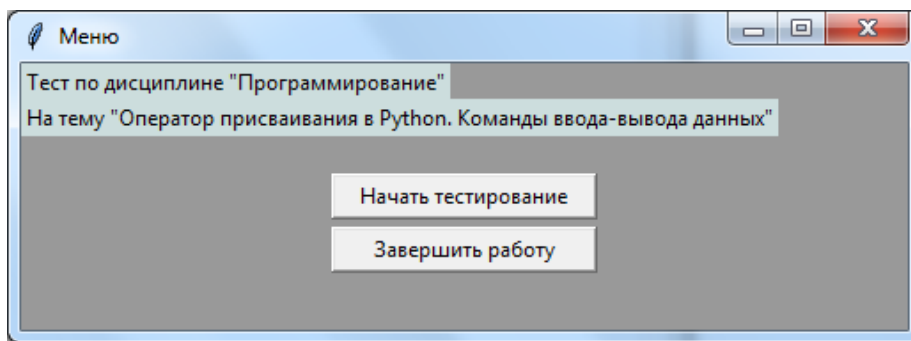


Рисунок 2. Пример работы Программы 2

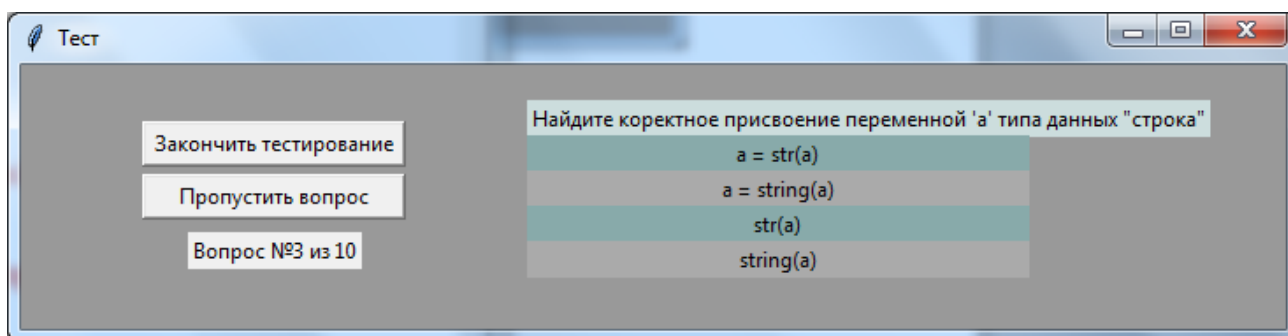


Рисунок 2. Продолжение

Порядок оформления отчета и защита ознакомительной практики для студентов очной формы обучения

Результаты практики оформляются в виде отчета, который имеет следующую структуру:

- Титульный лист.
- Индивидуальное задание на практику.
- Введение.
- Основная часть: постановка задачи.
- Описание классов, используемых при разработке программы.
- Макет GUI с описанием используемых виджетов.
- Описание событий и связанных с ними методов.
- Текст программы.
- Содержимое тестового файла с вопросами и правильными ответами.
 - Скриншоты выполнения программы.
 - Заключение.
 - Список использованных источников.

В Приложении 1 приведен пример оформления титульного листа отчета, а в Приложении 2 – индивидуальное задание на ознакомительную практику, в которое нужно занести сведения о студенте и соответствующие даты.

Промежуточная аттестация по практике проводится руководителем (руководителями) практики от университета в сроки, предусмотренные календарным учебным графиком и приказом о направлении на практику. Промежуточная аттестация по практике проходит в форме собеседования обучающегося с руководителем практики (защита результатов прохождения практики). Результаты промежуточной аттестации обучающегося оцениваются дифференцированно в зависимости от соответствия представленной программы и отчета требованиям индивидуального задания.

При выставлении оценки учитывается структура и дизайн графического интерфейса программы тестирования, содержание вопросов и правильных ответов на них, а также порядок оформления отчета.

Задание на ознакомительную практику для студентов заочной формы обучения

Задание 1. Тема: Графика в Python

Согласно своему варианту написать программу на языке Python, формирующую указанное изображение с использованием графических примитивов (минимум пять простейших фигур). Цветовую гамму и размер выбрать самостоятельно, допускается собственный рисунок.

Таблица 1. Перечень заданий по вариантам

№ варианта	Задание
0	Домик
1	Цветок
2	Машина
3	Велосипед
4	Пирамидка
5	Кукла
6	Кубик
7	Снеговик
8	Светофор
9	Бабочка

Примеры:

Задача 1. Нарисовать флаг и шарик с помощью графических примитивов:

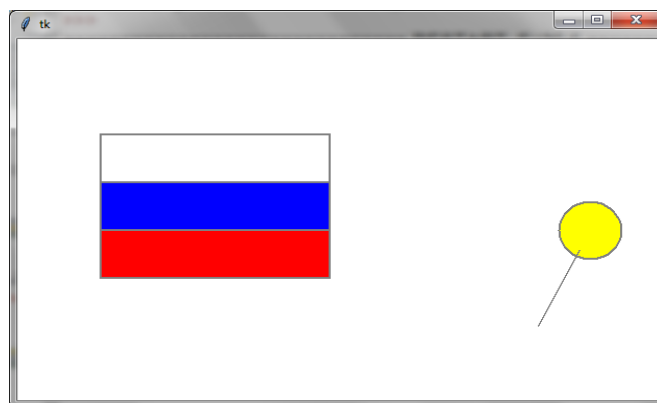


Рисунок 3. Результат работы программы задания 1

Методические рекомендации к задаче 1:

Рекомендуется для размещения объектов на холсте предварительно представить их координаты на черновике, с учетом того, что значение координаты y возрастает при перемещении вниз. Порядок изображения и расположения объектов зависит только от вас. Обратите внимание, что заливка используется только для замкнутых фигур, таких как круг или многоугольник.

Пример программы:

```
# Подключаем графический модуль
from graphimport *
# Устанавливаем ширину и высоту рабочей области окна
# Устанавливаем координаты области рисования
windowSize (800, 600)
canvasSize (700, 500)
# Устанавливаем толщину и цвет пера
penColor ("grey")
penSize (2)
# Устанавливаем цвета заливки и
# рисуем прямоугольники с координатами противоположных углов
brushColor ("white")
rectangle (80, 100, 300, 150)
brushColor ("blue")
rectangle (80, 150, 300, 200)
brushColor ("red")
rectangle (80, 200, 300, 250)
# Устанавливаем цвет заливки и
# рисуем окружность с центром в точке (x , y) установленного
радиуса
brushColor ("yellow")
circle (550, 200, 30)
# Рисуем линию между точками с координатами
Line (500, 300, 540, 220)
# Запускаем основной цикл обработки сообщений
run()
```

Модуль `graph` – это набор функций, который представляет собой «обёртку» для создания графических программ на языке Python на основе виджета Canvas библиотеки Tkinter. Скачать модуль `graph` и файлы для его установки можно в ЭОИС в разделе данной практической работы или в Интернете на официальном сайте языка Python.

В таблицах 2 и 3 приведены функции организации окон и команды рисования геометрических фигур.

Таблица 2. Функции для работы с окнами

Работа с окном		
Действие	Функция	Пример
Объект главного окна	mainWindow ()	
Ширина и высота рабочей области окна	windowSize (width, height)	windowSize (800, 600)
Объект области рисования (холст)	canvas ()	
Начало координат (x, y) области рисования	canvasPos (x, y)	
Ширина и высота области рисования	canvasSize (width, height)	canvasSize (700, 500)
Диапазоны математической декартовой системы координат	viewCoords (x1, x2, y1, y2)	
Запускает основной цикл обработки сообщений; последняя строчка	run ()	
Закрывает графическое окно	close ()	
True, если точка с координатами (x, y) находится в пределах области рисования	pointInView (x,y)	print (pointInView (980, 80))
True, если окружность с центром в точке (x, y) радиуса r находится в пределах области рисования	circleInView (x, y,r)	

Таблица 3. Команды для создания графических примитивов

Команды для рисования геометрических фигур		
Действие	Функция	Пример
Толщина пера	penSize (width)	penSize (5)
Цвет пера (см. Таблицу HTML цветов)	penColor (color) penColor (r, g, b)	penColor ("blue") penColor (178, 34, 34)

Команды для рисования геометрических фигур		
Действие	Функция	Пример
Цвет заливки (см. Таблицу HTML цветов)	brushColor (color) brushColor (r, g, b)	
Точка цвета с координатами (x, y)	point (x, y) point (x, y, color)	point (100, 100, 'red')
Перейти в точку, заданную координатами (x, y) или кортежем pos= (x, y) из этих координат	moveTo (pos) moveTo (x, y)	pos = (120,45) moveTo (pos)
Линия из текущего положения в точку, с координатами (x, y) или кортежем pos= (x, y) из этих координат	lineTo (pos) lineTo (x, y)	
Линия между точками (x1, y1) и (x2, y2)	line (x1, y1, x2, y2)	
Ломаная линия, составленная из точек, представленных списком кортежей координат	polyline (p)	d=[(80,30), (130,30), (130,160), (80,80)] polyline (d)
Многоугольник (замкнутая ломаная линия)	polygon (points)	polygon(d)
Прямоугольник с координатами противоположащих углов (x1, y1) и (x2, y2)	rectangle (x1, y1, x2, y2)	rectangle (10, 20, 60, 50)
Окружность с центром в точке (x, y) радиусом r	circle (x, y, r)	circle (200, 100, 50)
Овал (вписанный в прямоугольник с точками (x1, y1) и (x2, y2))	oval (x1, y1, x2, y2)	Oval (100,50,400,200)
Текст с точки (x, y)	label ("Текст", x, y)	label ("Учимся рисовать", 50, 50, font='20')
Рисунок из файла (загружаются рисунки) формата GIF)	image (x, y, fileName)	image (50, 50, 'c:/LP.gif')

Таблица 4. Таблица HTML цветов

Формат Color	HEX – шестнадцатеричный формат	RGB – формат Красный, Зеленый, Голубой
Black	#000000	0, 0, 0
Gray	#808080	128, 128, 128
Silver	#C0C0C0	192, 192, 192
White	#FFFFFF	255, 255, 255
Fuchsia	#FF00FF	255, 0, 255
Purple	#800080	128, 0, 128
Red	#FF0000	255, 0, 0
Maroon	#800000	128, 0, 0
Yellow	#FFFF00	255, 255, 0
Olive	#808000	128, 128, 0
Lime	#00FF00	0, 255, 0
Green	#008000	0, 128, 0
Aqua	#00FFFF	0, 255, 255
Teal	#008080	0, 128, 128
Blue	#0000FF	0, 0, 255
Navy	#000080	0, 0, 128

Задание 2. Тема: Анимация в Python

Используя результаты предыдущей работы, напишите на языке Python программу анимации для вашего графического объекта. Перемещение управляется клавишами стрелками. Траекторию и завершение движения определите по варианту.

Таблица 5. Перечень заданий по вариантам

№ варианта	Задания
0.	Движение от верхнего правого угла поля к нижнему левому, заканчивается при достижении левого края поля
1.	Движение по полю налево и направо с разной скоростью, заканчивается при достижении края поля
2.	Движение по полю вверх и вниз с разной скоростью, заканчивается при достижении края поля

№ варианта	Задания
3.	Движение от левого нижнего угла поля к правому верхнему, заканчивается при нажатии клавиши ESC
4.	Движение по полю в разных направлениях, заканчивается только при достижении нижнего края поля
5.	Движение от верхнего правого угла поля к нижнему левому с разной скоростью, заканчивается при нажатии клавиши ESC
6.	Движение по полю снизу вверх, заканчивается при достижении края поля
7.	Движение только по диагоналям поля, заканчивается при нажатии клавиши ESC
8.	Движение в любых направлениях, заканчивается при нажатии клавиши ESC
9.	Движение от левого верхнего угла поля к правому нижнему, заканчивается при достижении правого края поля

Примеры:

Пример 2. Организовать «полёт» шарика:

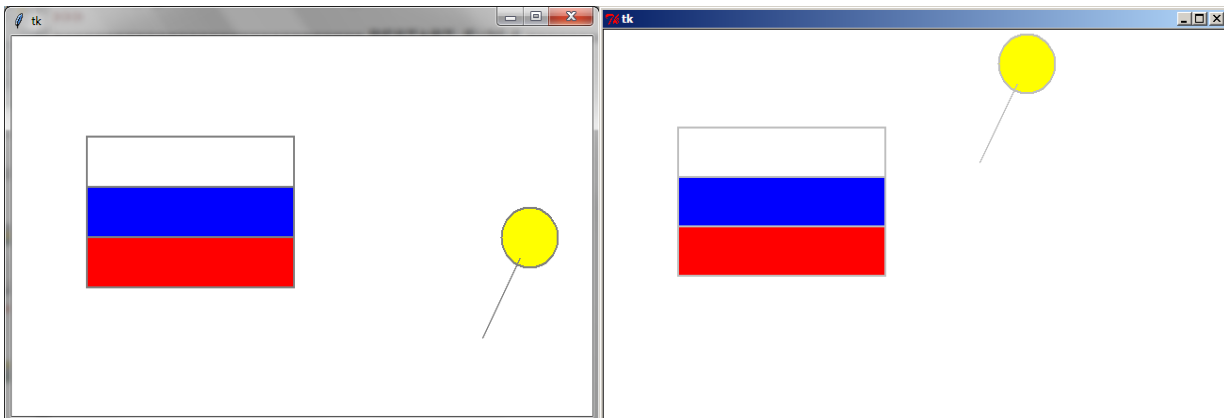


Рисунок 4. Результат работы программы задания 2

Создание анимации

Анимация (англ. animation) – оживление изображения на экране.

Технология создания анимации:

- рисуем объект в точке (x, y);
- выполняем задержку на несколько миллисекунд;
- стираем объект;
- изменяем координаты (x, y);
- переходим к шагу 1.

Фрагменты рисунка нужно определить как объекты, которые потом можно перерисовывать: obj = функция для рисования объекта.

Таблица 6. Функции для работы с графическими объектами

Функции для работы с графическими объектами		
Действие	Функция	Пример
Определение объекта	obj = функция для рисования объекта	shar = circle (x, y, 30)
Функция возвращает x-координату левого верхнего угла прямоугольника	xCoord(obj)	x1=xcoord(shar)
Функция возвращает y-координату левого верхнего угла прямоугольника	yCoord(obj)	y1=ycoord(square)
Установить для объекта obj новые координаты противоположных углов прямоугольника, (x1,y1) и (x2,y2), в который вписано изображение объекта	changeCoords (obj, pos)	pos= [(10,100), (50,100)] changeCoords (linia, pos)
Установить для объекта obj новый цвет контура color	changePenColor (obj, color)	changePenColor (h, "red")
Установить для объекта obj новый цвет заливки color	changtFillColor (obj, color)	changtFillColor (h, "black")
Установить для объекта obj новые свойства	changeProperty (obj, ...)	changeProperty (linia, fill="green")
Переместить левый верхний угол объекта obj в точку с координатами (x, y)	moveObjecTo (obj, x, y)	moveObjecTo (linia, 10, 10)
Переместить объект obj на вектор (dx, dy)	moveObjectBy (obj, dx, dy)	moveObjectBy (linia, 10, 10)
Удалить объект по ссылке	deleteObject (obj)	deleteObject (l)

Обработчики событий области рисования

Чтобы организовать срабатывание (вызов) функции через определенные промежутки времени, применяют «обработчик события». Событие – это изменение состояния программы или некоторое действие пользователя.

`onTimer (fn, time)` – установить функцию `fn`, которая будет вызываться по таймеру каждые `time` миллисекунд.

`onKey (key)` `onKey (fn)` `onKey (key, fn)` – установить функцию `fn` как обработчик нажатия клавиши с символьным обозначением `key`.

Если функция не указана, обработчик нажатия этой клавиши отключается; если не указана клавиша, устанавливается один обработчик на все клавиши; функция `fn` должна принимать один параметр – блок данных о событии.

Для того чтобы выяснить коды и символьные обозначения клавиш, можно использовать такую программу:

<pre>import tkinter as tk def _keyPress (event): print ("Key Press Event:") print (" event.char:", event.char) print (" event.keysym:", event.keysym) print (" event.keycode:", event.keycode) print (" event.keysym_num:", event.keysym_num) root = tk.Tk () root.bind ("<KeyPress>", __keyPress) tk.mainloop ()</pre>	<pre>>>> Key Press Event: event.char: ı event.keysym: odiaeresis event.keycode: 87 event.keysym_num: 246 Key Press Event: event.char: event.keysym: Alt_L event.keycode: 18 event.keysym_num: 65513 Key Press Event: event.char: event.keysym: Shift_L event.keycode: 16 event.keysym_num: 65505 Key Press Event: event.char: w event.keysym: w event.keycode: 87 event.keysym_num: 119</pre>
---	--

Методические рекомендации к задаче 2:

Рисунки определяются как объекты, с которыми работает программа.

- В программе используются три функции:

- 1) изменения координат объектов при фиксации определенных клавиш,
- 2) перемещения объектов,
- 3) проверки выхода объектов за границы рисунка.

- В программе применяют «обработчик события» (обработчик нажатия клавиши) и функцию, которая будет вызываться по таймеру каждые `time` миллисекунд.

Пример программы:

```
from graph import *
canvasSize (700, 500)
windowSize (800, 600)
# Функция определения шагов перемещения (dx и dy)
# при нажатии на клавиши – стрелки или клавишу пробел
# Функция закрывает окно при нажатии на клавишу ESC
defkeyPressed (event):
    global dx, dy
    if event.keycode == VK_LEFT:
        dx = -5; dy = 0
    elif event.keycode == VK_RIGHT:
        dx = 5; dy = 0
    elif event.keycode == VK_UP:
        dx = 0; dy = -5
    elif event.keycode == VK_DOWN:
        dx = 0; dy = 5
    elif event.keycode == VK_SPACE:
        dx = dy = 0
    elif event.keycode == VK_ESCAPE:
        close()
# Функция перемещения объектов
defupdate():
    moveObjectBy(shar, dx, dy)
    moveObjectBy(linia, dx, dy)
# Функция проверки выхода фигуры за край поля
def end():
    if 0>min (coords(shar)) or 600<max(coords(shar)):
        print ('Шарик улетел')
    close()
#Основная программа
penColor("grey")
penSize(2)
brushColor("white")
rectangle(80, 100, 300, 150)
brushColor("blue")
rectangle(80, 150, 300, 200)
brushColor("red")
rectangle(80, 200, 300, 250)
# Определяем начальное положение фигур
x = 550; y = 200
# Задаем нулевые значения шагов изменения координат
dx = 0; dy = 0
brushColor("yellow")
# Определяем объекты для анимации
shar = circle(x, y, 30)
linia = line(x-50, y+100, x-10, y+20)
# Выполняем функцию как обработчик нажатия клавиш keyPressed
onKey (keyPressed)
# Определяем функции update и end ,
# которые будут вызываться по таймеру каждые 50 миллисекунд
onTimer (update, 50)
onTimer (end, 50)
run()
```

Создание объекта для последующей анимации

```
from graph import *  
Brush Color ("red")  
obj=circle(60,60,30)  
brush Color ("green")  
obj0 = rectangle (40,120,160,480)  
obj1 = line (300,200,100,700)  
run()
```

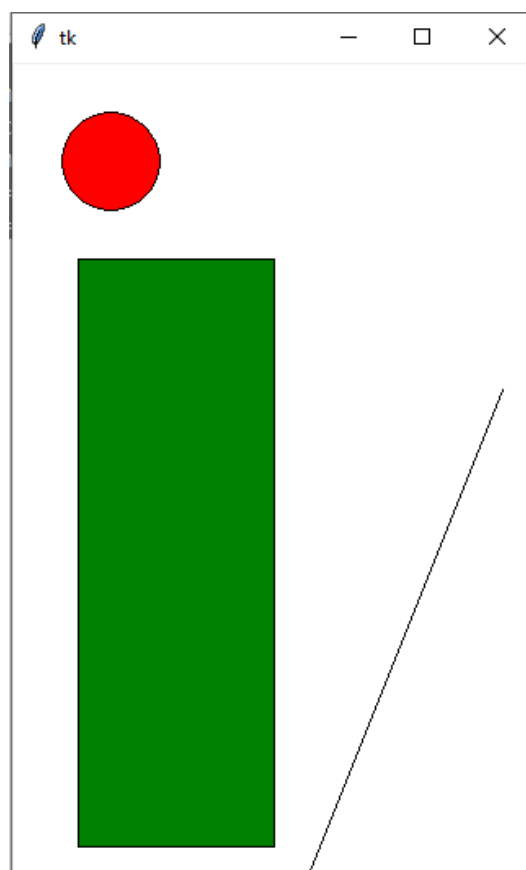


Рисунок 5. Результат работы программы создания объектов для анимации

Ниже приведены примеры фрагментов программы с элементами анимации:

Перемещение объекта на координаты:

```
from graph import *  
windowSize(400,400)  
brushColor("red")  
obj=circle (60,60,30)  
moveObjectTo(obj,200,200)
```

Изменение цвета заливки объекта:

```
from graph import *  
windowSize(400,400)  
brushColor("red")  
obj=circle(60,60,30)  
changeFillColor (obj,"blue")
```

Изменения цвета контура объекта:

```
from graph import *
penSize(2)
windowSize(400,400)
brushColor("red")
obj=circle(60,60,30)
changePenColor(obj,"blue")
```

Перемещение объекта по диагоналям:

```
from graph import *
def keyPressed(event):
    global dx,dy
    if event.keycode == VK_LEFT: dx=-5; dy=-5
    elif event.keycode == VK_RIGHT: dx=5; dy=5
    elif event.keycode == VK_UP: dx=5; dy=-5
    elif event.keycode == VK_DOWN: dx=-5; dy=5
def update():
    moveObjectBy(obj,dx,dy)
dx=0
dy=0
windowSize(400,400)
brushColor("red")
obj=circle(60,60,30)
onKey(keyPressed)
onTimer(update, 30)
```

Перемещение объекта по горизонтали и вертикали:

```
from graph import *
def keyPressed(event):
    global dx,dy
    if event.keycode == VK_LEFT: dx=-5; dy=0
    elif event.keycode == VK_RIGHT: dx=5; dy=0
    elif event.keycode == VK_UP: dx=0; dy=-5
    elif event.keycode == VK_DOWN: dx=0; dy=5
def update():
    moveObjectBy(obj,dx,dy)
dx=0
dy=0
windowSize(400,400)
brushColor("red")
obj=circle(60,60,30)
onKey(keyPressed)
onTimer(update, 30)
```

Порядок оформления отчета и защита ознакомительной практики для студентов заочной формы обучения

Ниже приведен макет отчета по ознакомительной практике для студентов заочной формы обучения:

Введение

Ознакомительная практика по получению первичных профессиональных умений и навыков входит в состав вариативной части образовательной программы бакалавриата и проводится после теоретического обучения и экзаменационной сессии второго семестра.

Целью ознакомительной практики студентов является закрепление и расширение теоретических знаний и практических навыков, получаемых студентом во время учебного процесса. После первого курса практика нацелена на освоение основ программирования на алгоритмическом языке Python, в частности работа с графическим пользовательским интерфейсом на примере индивидуального задания.

Задание 1

Тема: Графика в Python

Контрольные вопросы:

1. Какие средства языка предназначены для изображения простейших фигур?
2. Как задать цвет и толщину линий?
3. Варианты задания цвета объекта?
4. Для каких фигур можно выполнить заливку и как задать цвет заливки?
5. Как представлены координаты точек на холсте?
6. Как определить область рисования?

Задание 1. Согласно своему варианту написать программу на языке Python, формирующую указанное изображение с использованием графических примитивов (простейших фигур). **Допускается собственный рисунок.** Цветовую гамму и размер выбрать самостоятельно.

Изображение согласно варианту _____

Ответы на вопросы:

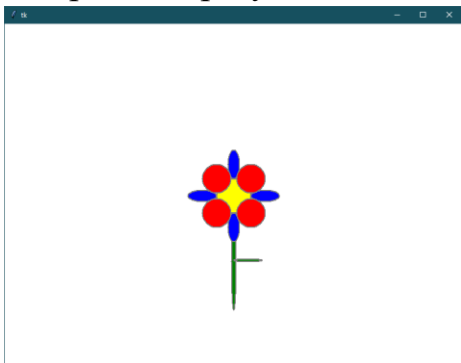
.....

Программа:

#Выполнил студент группы 2_-ЗИЭ

.....

Скриншот результата:



Задание 2

Тема: Анимация в Python

Контрольные вопросы:

1. Что понимают под объектом при создании анимации?
2. Какие средства языка предназначены для перемещения объектов?
3. Что такое «обработчики событий области рисования»?
4. Какие координаты объекта определяет функция coords (obj)?
5. Как задать шаг изменения координат при перемещении объекта?
6. Как завершить работу программы?

Задание 2. Используя результаты предыдущей работы, напишите на языке Python программу анимации для вашего графического объекта. Перемещение управляется клавишами-стрелками. Траекторию и завершение движения определите по варианту: _____

Ответы на вопросы:

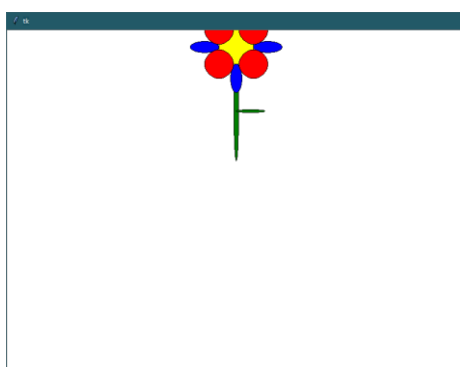
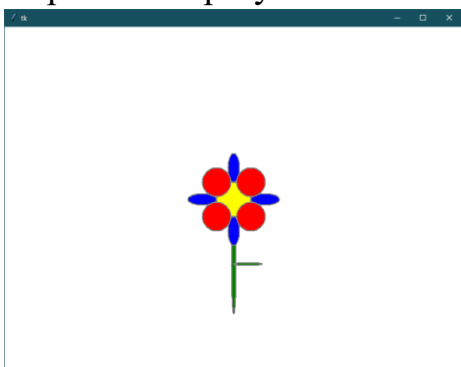
.....

Программа:

#Выполнил студент группы 2_-ЗИЭ

.....

Скриншоты результатов:



Заключение

Результат ознакомительной практики:

Получение навыков практического применения, полученных в процессе обучения знаний элементов языка Python

Промежуточная аттестация по практике проводится руководителем (руководителями) практики от университета в сроки, предусмотренные календарным учебным графиком и приказом о направлении на практику. Промежуточная аттестация по практике проходит в форме собеседования обучающегося с руководителем практики (защиты результатов прохождения практики). Результаты промежуточной аттестации обучающегося оцениваются дифференцированно в зависимости от соответствия представленной программы и отчета требованиям индивидуального задания.

Список литературы

1. Хахаев, И. А. Практикум по алгоритмизации и программированию на Python: курс : учебное пособие : [16+] / И. А. Хахаев. – 2-е изд., исправ. – Москва : Национальный Открытый Университет «ИНТУИТ», 2016. – 179 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=429256> (дата обращения: 30.03.2022). – Библиогр. в кн. – Текст : электронный.

2. Буйначев, С. К. Основы программирования на языке Python: учебное пособие / С. К. Буйначев, Н. Ю. Боклаг ; Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. – Екатеринбург : Издательство Уральского университета, 2014. – 92 с. : табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=275962> (дата обращения: 30.03.2022). – Библиогр. в кн. – ISBN 978-5-7996-1198-9. – Текст : электронный.

3. Сузи, Р. А. Язык программирования Python: учебное пособие : [16+] / Р. А. Сузи. – 2-е изд., испр. – Москва : Интернет-Университет Информационных Технологий (ИНТУИТ) : Бинوم. Лаборатория знаний, 2007. – 327 с. – (Основы информационных технологий). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=233288> (дата обращения: 30.03.2022). – ISBN 978-5-9556-0109-0. – Текст : электронный.

4. Лекция 14. Разработка приложений с графическим интерфейсом пользователя Библиотека Tkinter – Текст: [Электронный ресурс]; Режим доступа: <http://python.inr.ru/s1914.pdf> (дата обращения: 24.04.2022).

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО РЫБОЛОВСТВУ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Калининградский государственный технический университет»

Институт цифровых технологий

Кафедра прикладной информатики

ОТЧЕТ

по учебной ознакомительной практике

База практики: ФГБОУ ВО КГТУ кафедра прикладной информатики
(наименование предприятия)

Выполнил студент группы

_____ ,

Форма обучения _____, курс

Подпись

(И.О. Фамилия)

Руководитель практики от предприятия _____

(при наличии)

Подпись

(И.О. Фамилия)

Руководитель практики от кафедры _____

Сокращенное наименование кафедры

Подпись

(И.О. Фамилия)

Отчет защищен с оценкой _____

Дата защиты отчета _____

Калининград

202 _

Федеральное агентство по рыболовству
 Федеральное государственное бюджетное образовательное учреждение
 высшего образования
 «Калининградский государственный технический университет»
 Кафедра прикладной информатики

УТВЕРЖДАЮ

Зав. кафедрой / /

« » 20 г.

Индивидуальное задание

на ознакомительную практику по получению первичных профессиональных
 умений и навыков, в том числе первичных умений и навыков научно-
 исследовательской деятельности

(вид, тип практики)

студента

(Ф.И.О. полностью)

(группа)

Направление 09.03.03 Прикладная информатика
 подготовки
 (специальность)

Место прохождения
 практики:

ФГБОУ ВО КГТУ, кафедра ПИ
 236022 Калининград, Советский проспект 1

За время прохождения

практики: с « » июня 20 г.
 по « » июля 20 г.

студент должен выполнить следующие виды работ (заданий):

№ п/п	Содержание практики (наименование работ/заданий)	Рабочий график практики
1	Получение индивидуальных заданий студентами	
2	Выполнение индивидуального задания: разработка программы для тестирования по заданной теме на основе GUI на алгоритмическом языке Python	
3	Подготовка отчета по учебной практике, защита практики	

Планируемые результаты практики

Код и наименование компетенции. Индикаторы достижения компетенции	Результаты обучения, соотнесенные с компетенциями/индикаторами достижения компетенции
<p>ПКС-8: Способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем: ПКС-8.3: Формирование первичных профессиональных умений и навыков в осуществлении и обосновании выбора проектных решений</p>	<p>Должен знать:</p> <ul style="list-style-type: none"> - стандартные критерии выбора проектных решений; - базовые синтаксические конструкции языка С; - структуру типовой программы на языке С; - специфические особенности языка С в реализации подпрограмм, организации ввода/вывода, работе с типами данных; - структуру обзоров научной литературы и электронных информационных ресурсов.
<p>ПКС-9: Способен программировать приложения и создавать программные прототипы решения прикладных задач: ПКС-9.4: Формирование первичных профессиональных умений и навыков программирования приложений и создания программных прототипов решения прикладных задач</p>	<p>Должен уметь:</p> <ul style="list-style-type: none"> - оперировать механизмами выбора проектных решений; - конструировать программы на языке С, реализующие типовые алгоритмы обработки информации; - работать с механизмами подпрограмм на языке С;
<p>ПКС-12: Способен готовить обзоры научной литературы и электронных информационно-образовательных ресурсов для профессиональной деятельности: ПКС-12.3: Формирование первичных профессиональных умений и навыков подготовки обзоров научной литературы и электронных информационных ресурсов</p>	<ul style="list-style-type: none"> - использовать известные механизмы подготовки обзоров научной литературы и электронных информационных ресурсов. <p>Должен владеть:</p> <ul style="list-style-type: none"> - стандартами в области информационных технологий по осуществлению и обоснованию выбора проектных решений; - механизмами поиска учебной литературы по изучаемым языкам программирования; - типовыми механизмами отладки программ на языке С; - стандартными механизмами формирования агрегатных вычислений в сложных отчетах; - методиками и рекомендациями по подготовке обзоров научной литературы и электронных информационных ресурсов. <p>Должен приобрести опыт:</p> <ul style="list-style-type: none"> - на примере практических задач по осуществлению и обоснованию выбора проектных решений; - создания на языке С типовых программ обработки информации; - самостоятельного освоения новых языков программирования; - подготовки обзоров научной литературы и электронных информационных ресурсов

Руководитель
практики от
университета

(подпись)

(Фамилия И.О., должность)

Руководитель
практики от
профильной
организации

(подпись)

(Фамилия И.О., должность)

Практикант

(подпись)

(телефон, E-mail)

« _____ » _____ ИЮЛЯ 20 _____ Г.

Локальный электронный методический материал

Елена Юрьевна Заболотнова
Светлана Александровна Калинина

ОЗНАКОМИТЕЛЬНАЯ ПРАКТИКА

Редактор М. А. Дмитриева

Уч.-изд. л. 0,8. Печ. л. 1,7.

Издательство Федерального государственного бюджетного образовательного
учреждения высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1.