

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

Л. Г. Высоцкий

ВЫСОКОУРОВНЕВЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ (ВТП)

Учебно-методическое пособие
по выполнению курсовой работы
для студентов, обучающихся в бакалавриате по направлению подготовки
09.03.03 Прикладная информатика (заочное отделение)

Калининград
Издательство ФГБОУ ВО «КГТУ»
2022

УДК 004.9(075)

Рецензент:

кандидат педагогических наук, доцент кафедры прикладной информатики
ФГБОУ ВО «Калининградский государственный технический
университет» Е. Ю. Заболотнова

Высоцкий, Л. Г.

Высокоуровневые технологии программирования (ВТП): учебно-методическое пособие по выполнению курсовой работы для студентов, обучающихся в бакалавриате по направлению подготовки 09.03.03 Прикладная информатика (заочное отделение) / **Л. Г. Высоцкий.** – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2022. – 42 с.

Учебно-методическое пособие включает подробные указания по выполнению курсовой работы по дисциплине «Высокоуровневые технологии программирования». Рассмотрены порядок выполнения и содержание каждого этапа разработки программного приложения. Приведен список рекомендуемых тем для курсовой работы. В пособие также включены подробные указания по оформлению пояснительной записки.

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 рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией ИЦТ 20 сентября 2022 г., протокол № 6

УДК 004.9(075)

© Федеральное государственное бюджетное
образовательное учреждение
высшего образования
«Калининградский государственный
технический университет», 2022 г.
© Высоцкий Л. Г., 2022 г.

Оглавление

ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ	4
УСЛОВИЯ ВЫБОРА ТЕМЫ И ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	6
ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ	7
МЕТОДИКА ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	10
ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ	20
ОРГАНИЗАЦИЯ ЗАЩИТЫ КУРСОВОЙ РАБОТЫ.....	28
Список использованных источников	32
ПРИЛОЖЕНИЕ 1.....	33
ПРИЛОЖЕНИЕ 2	35
ПРИЛОЖЕНИЕ 3	36
ПРИЛОЖЕНИЕ 4.....	37

ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Основной целью выполнения курсовой работы по дисциплине «Высокоуровневые технологии программирования» является закрепление, углубление и обобщение знаний, полученных студентами во время изучения дисциплины «Высокоуровневые технологии программирования», а также определение связи данной дисциплины с будущей специальностью.

Курсовая работа позволяет:

- систематизировать, закрепить и расширить теоретические знания и практические навыки, полученные на лекциях и лабораторных работах курса, применить их при решении конкретных задач, связанных с будущей специальностью;
- развить навыки самостоятельной работы;
- повысить уровень подготовки по специальным разделам (алгоритмизация, кодирование, отладка) дисциплины «Высокоуровневые технологии программирования».

Курсовая работа должна подготовить студентов к самостоятельному применению высокоуровневых технологий программирования и в дальнейшем использовать их при подготовке выпускной квалификационной работы. Дополнительными задачами данного вида отчетности студентов являются:

- приобретение навыков оформления печатной работы в соответствии с предъявленными требованиями для использования этих навыков при подготовке будущей квалификационной работы;
- использование средств электронной презентации для доклада при защите работы.

Данная курсовая работа ориентирована на освоение студентами навыков решения частной, но часто встречающейся на практике задачи – формирование графиков различных зависимостей на пространстве определенного размера визуальными средствами современных языков программирования. При этом графики

должны формироваться как элемент более широкого программного компонента – графического пользовательского интерфейса (GUI), через который идет взаимодействие пользователя с программным приложением. В задании на курсовую работу студентам предлагается для этого использовать средства модуля `tkinter` языка `Python`. Но студент может выбрать для формирования графиков и средства других систем программирования при условии, что будет решена требуемая задача.

В результате выполнения курсовой работы студент должен расширить свои знания:

- в области средств и возможностей языка реализации проекта;
- типовых элементов графического пользовательского интерфейса;
- стандартов оформления технической документации.

Повысить свои умения в:

- систематизации и конкретизации теоретических знаний в вопросах создания программных систем;
- поиске и анализе информации, необходимой по теме курсовой работы;
- использовании средств и функциональных возможностей языка реализации проекта;
- разработке и тестировании программных приложений, включающих графический пользовательский интерфейс;
- разработке структуры сложного программного приложения;
- формулировании выводов и предложений.

Приобрести навыки:

- самостоятельного решения вопросов, возникающих в ходе создания сложной программной системы;
- системного мышления через определение целей и постановку задачи курсовой работы;

- практического использования теоретического материала по программированию, освоенного за предыдущее время обучения;
- оформления программной документации в соответствии со стандартами;
- самостоятельного освоения дополнительного технического материала по теме работы.

УСЛОВИЯ ВЫБОРА ТЕМЫ И ПОРЯДОК ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

В настоящее время общение пользователей с программными системами в подавляющем большинстве реализуется через графический интерфейс, что резко расширяет информационные возможности человеко-машинного диалога, например, позволяя формировать широкий круг различных графиков и диаграмм, отображающих анализируемые системой зависимости. Основная цель данной курсовой работы – повышение умений и знаний студентов в разработке таких интерфейсов в процессе создания графиков различной точности по заданным математическим функциям.

Тема курсовой работы выбирается студентом из предложенного списка вариантов математических функций и диапазона их аргументов, приведенного в данном учебно-методическом пособии.

Все темы работы носят индивидуальный характер. Вместе с тем общим для них является, как указывалось ранее, разработка графического пользовательского интерфейса, в составе которого формируются три варианта одного и того же графика с различной точностью.

Процесс выполнения курсовой работы предполагает следующую очередность этапов:

- Из предложенного списка выбрать свой вариант курсовой работы.
- Разработать обобщенную блок-схему работы программного приложения.
- Изучить и проанализировать литературу и другие источники по теме работы,

- Разработать интерфейс системы (видеоформы), включая в него виджеты, обеспечивающие выполнение требуемой задачи по построению графиков.
- Разработать алгоритм в виде подробной блок-схемы для построения каждого вида графика.
- Провести кодирование и отладку ПС.
- Максимально снабдить листинг комментариями;
- Провести проверку корректности построения графиков средствами пакета Microsoft Excel;
- Оформить пояснительную записку по курсовой работе в соответствии со стандартами на техническую документацию по программированию, включающую требуемые разделы.

ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

По объему курсовая работа должна занимать не менее 20—25 страниц печатного текста. Содержание курсовой работы должно свидетельствовать о достаточно высокой теоретической подготовке студента, которую он должен получить на данной дисциплине, и о наличии у автора необходимых знаний по теме работы. Текст пояснительной записки должен быть написан грамотно, хорошим литературным и профессиональным языком, Работа должна иметь правильно составленную библиографию, логичную структуру, обеспечивающую наиболее эффективно решение проблемы. Пояснительная записка, как и любая техническая документация, пишется в обезличенной форме, т. е., например, вместо «я считаю» - «считается». В курсовой работе должны присутствовать: титульный лист (Приложение 1), задание на курсовую работу, содержание, аннотация (реферат), основная часть, заключение, список использованных источников, список приложений.

Титульный лист оформляется в соответствии с примером, приведенным в Приложении 1.

Содержание – Содержание включает наименование всех разделов и подразделов, пунктов, списка литературы и приложений с указанием номеров страниц. Аннотация и содержание в оглавлении не указываются. Подразделы и пункты указываются с отступами вправо. Названия всех разделов должны точно соответствовать тексту курсовой работы. Обязательно указываются страницы, с которых начинаются все пункты. Образец листа содержания приведен в Приложении 2.

Аннотация (реферат) – сокращенное описание работы, оформляется в соответствии с примером, приведенным в Приложении 3.

Обоснование выбора языка программирования – часть, в которой необходимо обосновать выбор языка программирования. Предполагается, что студент будет использовать в качестве языка реализации курсовой работы язык Python, но допускается по согласованию с руководителем работы использование других языков, поддерживающих ООП и GUI.

Постановка задачи – полное описание задания.

Описание модулей программы и их взаимосвязь - структурная схема состава программных модулей с кратким описанием функций. Описание должно однозначно показывать, что какой(ие) модуль(ли) предназначен(ы) для реализации каждой функции приложения.

Структура программного интерфейса – структура визуальной формы (скриншота), используемой при функционировании программы, с указанием вида и расположения задействованных виджетов, их функционального назначения.

Блок-схемы модулей – графическая схема алгоритма построения каждого варианта графика, выполненная в соответствии с требованиями ЕСКД. Общие требования представлены в Приложении 4.

Код (листинг) на выбранном языке программирования с подробными комментариями.

Вид результатов выполнения программы при выполнении тестового задания, т.е. скриншоты всех трех графиков, а также вид контрольного графика, построенного средствами пакета Microsoft Excel.

Заключение содержит четко сформулированные итоги проделанной работы и основные выводы, к которым пришел автор. Заключение должно быть кратким, обстоятельным и соответствовать цели и поставленным в начале задачам.

Список использованных источников представляет собой перечень использованных при написании курсовой работы разнообразных источников. Работа с литературой не ограничивается чтением учебников и учебных пособий. В настоящее время Интернет-ресурсы дают возможность познакомиться с самыми новыми тенденциями и процессами. В тексте работы должны быть обязательно приведены ссылки на использованные источники с указанием даты последнего обращения.

Список использованной литературы должен соответствовать современным требованиям по ГОСТу 2019 г.

Приложения к курсовой работе оформляются на отдельных листах, причем каждое приложение должно иметь свой тематический заголовок и номер, который пишется в правом верхнем углу, например: «Приложение I». В приложениях представлены формы, которые содержат большой цифровой и фактический материал, как правило, первичный, но дополняющий основной текст работы. Например, громоздкие подробные таблицы, текст листинга на языке реализации курсовой работы. Каждое приложение должно начинаться с новой страницы и иметь заголовок. Приложения нумеруются арабскими цифрами порядковой нумерации. Номер приложения размещается в правом верхнем углу над заголовком после слова «Приложение» (например: Приложение 1). На все Приложения в основной части курсовой работы должны быть ссылки.

МЕТОДИКА ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа является одним из видов научно-исследовательской деятельности и должна соответствовать современному состоянию отечественной и зарубежной науки, а также быть направленной на достижение определенной цели.

При работе над ней следует:

1. Изучить и проанализировать теоретические основы данных построений, литературу и другие источники по теме работы.

Поскольку цель курсовой работы – формирование графика некоторой математической функции на канве заданного размера, то студенту необходимо обязательно изучить теоретические основы таких построений, которые изложены ниже.

Канва формы модуля `tkinter` языка `Python` является эффективным средством создания графиков, иллюстрирующих математические зависимости. Исходным для построения является массив значений, преобразуемый в график с помощью метода канвы `create.polyline(<массив точек>)`, который подробно рассмотрен в лабораторных работах по дисциплине «Высокоуровневые технологии программирования».

Поскольку график всегда строится на пространстве определенного размера, то обязательным является его масштабирование на основе определенных заранее коэффициентов масштабирования как по оси X - m_x так и по Y - m_y . Предполагая, что размер пространства графика по вертикали равен H_y пикселей, коэффициент m_y определяется как

$$m_y = H_y / R_y,$$

где R_y - размах значений исходной функции, определяемый как

$$R_y = Y_{max} - Y_{min},$$

Y_{max} - максимальное значение функции;

Y_{min} - минимальное значение функции.

При этом возможны три варианта расположения графика относительно оси X (рис. 1).

В первом случае (а) $\mathcal{R}_y = \mathcal{Y}_{max}$, т. е. принимается, что $\mathcal{Y}_{min} = 0$. Соответственно, ось X отображается в самом низу графика.

Во втором случае (б) $\mathcal{R} = |\mathcal{Y}_{min}|$, т. е. принимается, что $\mathcal{Y}_{max} = 0$. Соответственно, ось X отображается на самом верху графика.

Для третьего случая основная проблема - правильное расположение оси X , для чего требуется знание m_y .

Соответственно, m_x определяется как

$$m_x = \mathcal{H}_x / \mathcal{R}_x$$

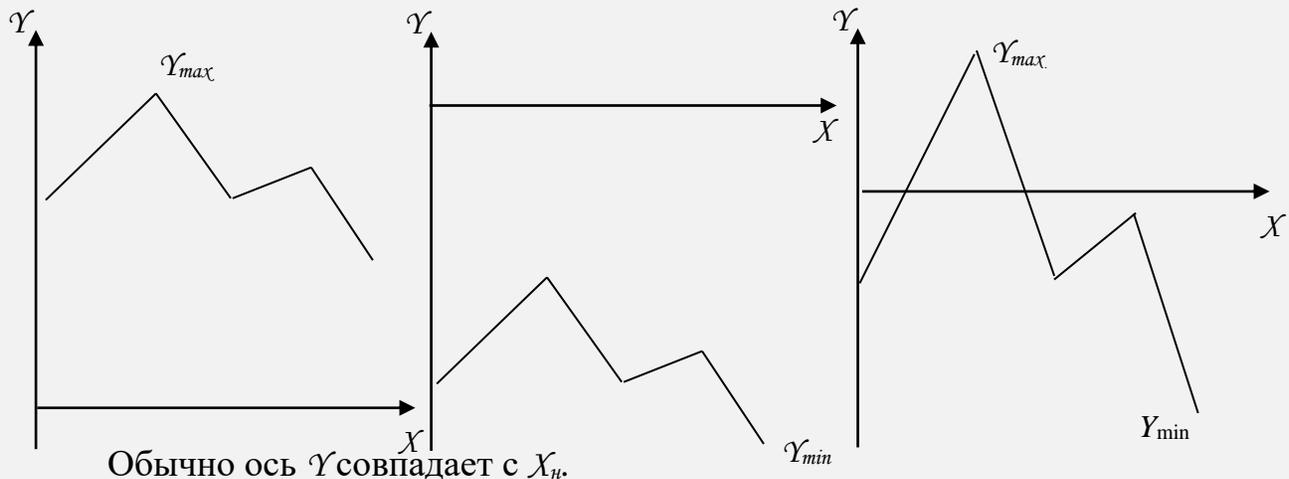
где \mathcal{H}_x - размер пространства графика по горизонтали в пикселях;

\mathcal{R}_x - размах значений аргумента, определяемый как

$$\mathcal{R}_x = X_k - X_n.$$

X_k - конечное значение аргумента;

X_n - начальное значение аргумента.



Обычно ось \mathcal{Y} совпадает с X_n .

а)

б)

в)

В дальнейшем при создании графика предполагается, что координаты его точек x_m относительно начала координат всей формы (0_Γ) определяются как (рис. 2).

$$x_m = m_x^* (x - X_n) + \Delta x$$

где x - очередное значение аргумента, а Δx - смещение начала локальной системы координат (координат графика с началом 0_l) относительно правой стороны формы в пикселях.

Координаты y_m находятся по более сложному алгоритму, поскольку, как отмечалось ранее, положительным для оси Y является направление вниз. Поэтому (рис. 2) координата конкретной точки находится как

$$y_m = \Delta y - m_y * y,$$

где y - очередное значение функции, а Δy - смещение начала локальной системы координат относительно верхней границы формы в пикселях

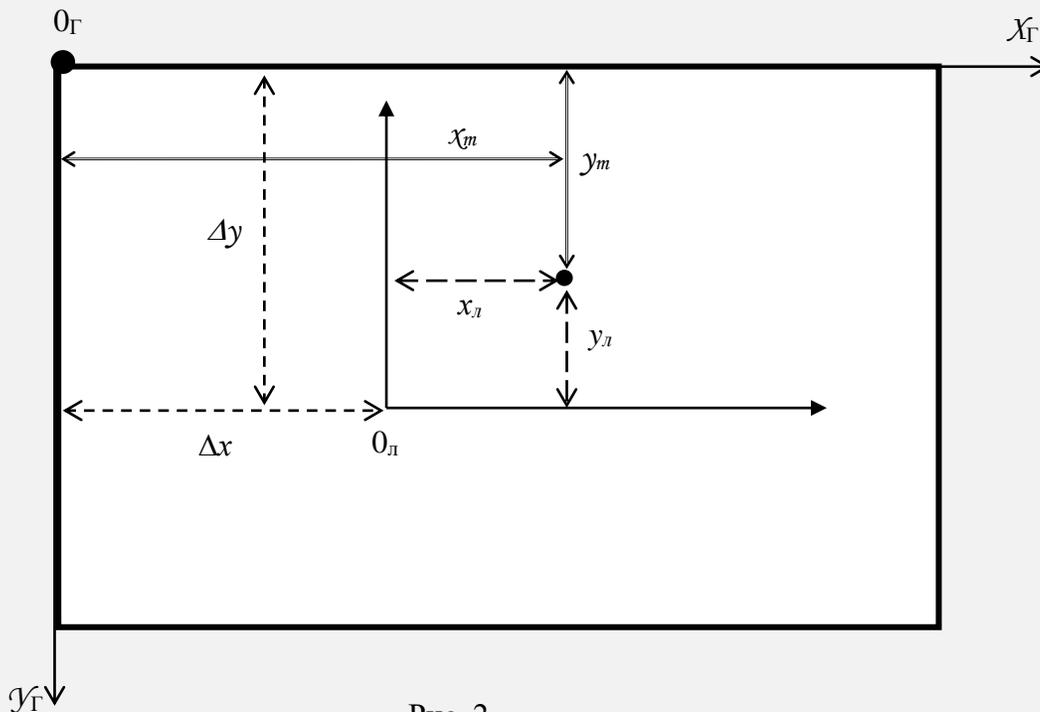


Рис. 2

Значение Δy зависит от варианта графика (см. рис. 1):

- в первом случае должно обеспечиваться условие $\Delta y \geq H_y$;
- во втором $\Delta y \geq 0$;
- для третьего случая $\Delta y \geq m_y * Y_{max}$.

Разметка координатных осей надписями и числовыми значениями может быть выполнена методом

```
<имя канвы>.create_text(<координаты опорной точки>, text="<отображаемый  
текст>", [<параметры шрифта>])
```

Рисование графика методом `create.polyline(<массив точек>)` с достаточно большим шагом аргумента приводит к созданию ломаной кривой, не отражающей истинного вида требуемой функции. Для повышения качества графика можно уменьшать шаг аргумента, сведя его в пределе к расчету координаты Y для каждого отдельного пикселя кривой, т. е. к

$$\Delta x = (X_k - X_n) / H_x.$$

Данный подход может привести к большому объему вычислений. Если не требуется абсолютная точность в отображении функции, то можно уменьшить объем таких вычислений путем сглаживания графика на основе использования параметра `smooth = 1` у полилинии, т. е. в первом варианте графика этот параметр находится в нулевом значении (значение по умолчанию).

Студентам также рекомендуется тщательно проработать лекционный и практический материал по графическим средствам используемого языка программирования (Python'a), а также дополнительные источники по этой теме, поскольку Python является достаточно динамичным языком, который периодически пополняется новыми средствами обработки как текстовой, так и графической информации. Необходимо, кроме того, вспомнить материал курса высшей математики, посвященный построению графиков математических функций.

2. Сформировать перечень функций, которые должны реализовываться в курсовой работе.

Исходя из общей процедуры построения графиков, изложенной выше, в ходе выполнения курсовой работы студенту необходимо реализовать следующие функции:

1. Определить размах указанной вариантом функции на заданном диапазоне аргумента и вариант графика в соответствии с рис. 1;

2. Определить коэффициенты масштабирования m_x и m_y для координат графика;
3. Построить координатные системы для каждого из трех видов графиков;
4. Построить 1-й график в виде ломаной линии;
5. Рассчитать коэффициент масштабирования m_x для 2-го графика с расчетом координаты Y для каждого отдельного пикселя кривой;
6. Построить 2-й график;
7. Продублировать 1-й график со сглаживания полилинии на основе использования параметра `smooth = 1`;
8. Проверить корректность построения графиков средствами пакета Microsoft Excel.

Поскольку все три графика представляют одну и ту же математическую функцию, то можно использовать один и тот же алгоритм для их построения, изменяя только Δx для 2-го графика.

3. Спроектировать графический интерфейс приложения.

Как указывалось выше, в данной работе требуется построить три варианта одного и того же графика. Для этого предлагается использовать виджет Notebook (многостраничный блокнот, рис. 3). Это требует предварительной загрузки в программу построения графиков расширения `ttk`, в котором находится этот виджет.

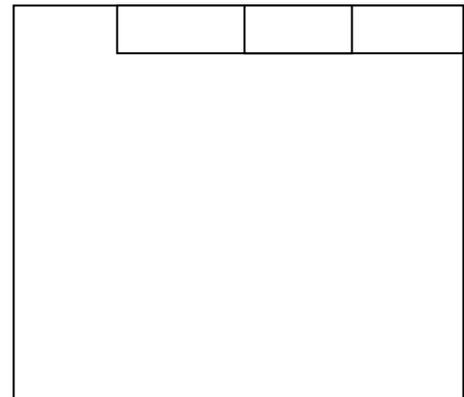


Рис. 3

Ниже приведен листинг для создания блокнота из двух страниц и размещения на одной из страниц канвы, как базы для формирования графика:

```

from tkinter import *
from tkinter import ttk

root = Tk()
root.title("Графики")

```

```
root.geometry('600x450')
```

```
tab_control = ttk.Notebook(root) #создается элемент управления  
вкладкой
```

```
tab1 = Frame(tab_control) # создается вкладка1
```

```
tab2 = Frame(tab_control) #создается вкладка2
```

```
tab_control.add(tab1, text='Первая') #текст ярлыка вкладки1
```

```
tab_control.add(tab2, text='Вторая') #текст ярлыка вкладки2
```

```
tab_control.pack(expand=1, fill='both')
```

```
c1 = Canvas(tab1, width=400, height=200, bg="white")
```

```
c1.place(x = 20, y = 20)
```

```
c2 = Canvas(tab2, width=400, height=200, bg="white")
```

```
c2.place(x = 20, y = 20)
```

```
root.mainloop()
```

Листание страниц блокнота при просмотре графиков осуществляется щелчками мыши по их ярлыкам.

Соответственно, блокнот должен располагаться на Windows-окне, созданном средствами языка Python или другого, обладающего всеми средствами для решения задачи курсовой работы.

4. Разработать алгоритм используемой функции.

Представляется целесообразным формировать графики одним из двух возможных алгоритмов:

1. В цикле на основе смены значений аргументов рассчитываются значения искомой функции и формируется <массив точек> полилинии, которая на втором шаге строится методом `create.polyline(<массив точек>)`;
2. В цикле для каждого значения аргумента рассчитываются значение искомой функции и строится отрезок линии от предыдущей точки к рассчитанной. После этого рассчитанная точка становится предыдущей.

Первый вариант алгоритма можно использовать для построения всех трех графиков. Но если количество точек во 2-м графике значительное, то его целесообразно строить по 2-му алгоритму.

5. Выполнить разработку программного кода.

Предполагается, что языком реализации курсовой работы будет Python, поскольку именно он изучается в курсе «Высокоуровневые технологии программирования», но студент может выбрать другой язык, обосновав его преимущество для реализации данной работы.

При формировании листинга рекомендуется соблюдать следующие принципы, выработанные поколениями программистов и пользователей и которые на профессиональном сленге программной инженерии звучат так:

1. KISS (Keep It Simple Stupid – будь проще, дурачок), т.е. предполагается отсутствие хитрых трюков и необычных конструкций в листинге, поскольку подобная свобода самовыражения, приветствуемая в искусстве, в технологической сфере может привести к непредсказуемым (а значит, возможно, и к катастрофическим) последствиям.
2. EIBTI (Explicit Is Better Than Implicit – явное лучше неявного), поскольку, в противном случае, процесс сопровождения ПО может превратиться в непрекращающийся кошмар выявления неявного специалистами по сопровождению.
3. Третий принцип еще не успел оформиться в соответствующий девиз с сокращением, но звучит он так: код читается чаще, чем пишется. Поэтому, выбирая между кодом, который проще пишется, и кодом, который проще читается, всегда надо выбирать второй вариант.

Студенты рассматриваемого здесь направления должны уже начиная с первых лет обучения вырабатывать свой *стиль программирования* - набор приемов или методов, которые используют опытные программисты, чтобы получить правильные, эффективные, удобные для применения и легкочитаемые программы. Обычно свой стиль - это результат здравого смысла, исходящего из опыта,

накапливаемого годами работы. Поэтому так важно начинать этот процесс как можно раньше.

Для повышения читабельности рекомендуется в листинге активно использовать неотображаемые символы (пробелы, отступы, пропуски строк) и комментарии. Главная посылка – хорошее визуальное форматирование показывает логическую структуру программы. Исследование показало, что оптимальное число пустых строк в программе лежит в диапазоне от 8 до 16 % ее объема;

Практика программирования также показала, что при большом размере листинга существует тенденция ухудшения структуры кода при внесении в него новой функциональности, исправления ошибок и пр. Появляется *избыточность*, образуются неиспользуемые или слабоиспользуемые *фрагменты*, структура становится запутанной и трудной для понимания. *Рефакторинг* – это регулярная деятельность, которой должны заниматься студенты в процессе работы над листингом, по переписыванию кода, но не с целью добавления новой функциональности, а для улучшения его структуры. Поэтому рекомендуется после получения работоспособного листинга еще раз просмотреть код именно с этой целью.

6. Разработать контрольный пример.

Результат выполнения разработанного программного приложения, т. е. три графика, сравнение их с результатом работы пакета Microsoft Excel и является одновременно контрольным примером данного задания.

7. Оформить работу в соответствии с предъявленными требованиями.

Ниже перечислены основные правила оформления программной документации:

Текстовые документы оформляются на листах формата А4, графический материал допускается представлять на листах формата А3.

Поля на листе: левое – не менее 30, правое – не менее 10, верхнее – не менее 15, нижнее – не менее 20 мм.

Нумерация страниц сквозная. Номера проставляются сверху справа арабской цифрой. Первой страницей считается титульный лист, на котором номер страницы не проставляется. В счет страниц входят и листы с рисунками и приложениями.

Названия разделов пишутся прописными буквами в середине строки. Расстояние между заголовком и текстом – два интервала.

Названия подразделов и пунктов начинаются с абзацного отступа, вразрядку, с прописной буквы и без точки в конце.

Расстояние между последней строкой текста предыдущего раздела и последующим заголовком при расположении их на одной странице – три интервала.

Разделы и подразделы нумеруются арабскими цифрами с точкой. Ссылки на пункты, разделы и подразделы указываются в виде «в разд. 4», «в п. 3.3.4».

Текст разделов печатается через 1,5 интервала. Высота шрифта №12. Шрифт - Times New Roman. Выравнивание абзацев – по ширине, отступ первой строки – 1,27. Расстояние между абзацами отсутствует.

Шрифт основных заголовков – 14, полужирный, подзаголовков – 13,5.

Перечисления отображаются арабскими цифрами со скобкой, например: 2), 3) и т. д. Допускается выделять перечисления дефисом перед пунктом текста или символом, его заменяющим.

Все иллюстрации именуется рисунками. Все рисунки, таблицы и формулы нумеруются арабскими цифрами последовательно или в пределах раздела. В приложениях – в пределах приложения. Каждый рисунок имеет подпись – название снизу рисунка, например:

Рис. 23. Главное окно программного комплекса.

На все рисунки, таблицы и формулы в документе должны быть ссылки в виде: «рис. 23» или «блок-схема данного модуля приведена на рис. 31». Если позволяет место, рисунки и таблицы должны размещаться сразу после абзаца, в

котором они упоминаются в первый раз, или как можно ближе к этому абзацу на следующей странице. Если рисунок занимает более одной страницы, то на всех страницах, кроме первой, проставляется номер рисунка и слово «Продолжение». Рисунки следует размещать так, чтобы их можно было рассматривать без поворота страницы. В противном случае рисунки надо размещать так, чтобы для просмотра страницу надо было повернуть по часовой стрелке.

Схемы алгоритмов выполняются в соответствии с ЕСПД (Приложение 4).

Номер таблицы размещается в правом верхнем углу или перед заголовком таблицы, если он есть. Ссылки на таблицы в тексте документа указываются в виде слов «табл.» и номера таблицы.

Номера формул ставятся с правой стороны в круглых скобках. Ссылка на номер формулы дается в скобках, например: «радиус окружности определяется по формуле (7)».

Каждое приложение должно начинаться с новой страницы с указанием в правом углу слова «ПРИЛОЖЕНИЕ» и иметь тематический заголовок. При наличии более одного приложения все они нумеруются арабскими цифрами. Все рисунки и таблицы в приложении нумеруются арабскими цифрами в пределах каждого приложения с добавлением буквы «П», например:

Рис. П. 9

Дополнительные требования:

- использовать автоматическую нумерацию страниц;
- содержание должно быть вставлено автоматически;
- желательно использовать колонтитулы;
- сокращения должны быть расшифрованы;
- курсовую работу необходимо представлять на кафедру в сброшюрованном виде и регистрировать ее на кафедре.
- обобщить результаты и сформулировать выводы.

В данном разделе пояснительной записки формулируются практические и теоретические результаты выполнения курсовой работы (название созданного программного приложения, его функции, перечень решенных при проектировании практических вопросов, сравнение с результатом проверки средствами пакета Microsoft Excel), определяются дополнительные опыт, знания и умения, которые были получены студентом в ходе этой работы.

ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ

Как указывалось ранее, график строится в трех вариантах на разных вкладках компонента Notebook:

- с заданным вариантом шагом Δx по оси X;
- с расчетом координат каждого пикселя графика;
- дублирование 1-го графика с использованием сглаживания полилинии.

Индивидуальные задания для построения представлены ниже (первое число – номер группы, второе – номер студента в списке группы).

1-1. На координатном пространстве размером 300x300 пикселей отобразить график заданной функции:

$$y = x^2 \sin(\sqrt{x}); 0 \leq x \leq 2; \Delta x = 0.1;$$

1-2. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = (8 + x) / (\sin(x) + 4); 0.1 \leq x \leq 1.8; \Delta x = 0.2.$$

1-3. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \cos^2(x); 5 \leq x \leq 10; \Delta x = 0.4.$$

1-4. На координатном пространстве размером 600x300 пикселей отобразить график заданной функции:

$$y = \cos(x^2); -5 \leq x \leq -2; \Delta x = 0.3.$$

1-5. На координатном пространстве размером 400x200 пикселей отобразить график заданной функции:

$$y = x^3 \cos(\sqrt{x}); 0.2 \leq x \leq 0.9; \Delta x = 0.1.$$

1-6. На координатном пространстве размером 500x400 пикселей отобразить график заданной функции:

$$y = (1 + x^2) / (2 + \cos(\sqrt{x})); 0.1 \leq x \leq 2.9; \Delta x = 0.2.$$

1-7. На координатном пространстве размером 600x400 пикселей отобразить график заданной функции:

$$y = x^2 / (\cos(x) - 6); -0.1 \leq x \leq 0.9; \Delta x = 0.1.$$

1-8. На координатном пространстве размером 500x500 пикселей отобразить график заданной функции:

$$y = \ln(x + 2); 0.5 \leq x \leq 5; \Delta x = 0.4.$$

1-9. На координатном пространстве размером 400x400 пикселей отобразить график заданной функции:

$$y = e^x; -1 \leq x \leq 1; \Delta x = 0.2.$$

1-10. На координатном пространстве размером 600x600 пикселей отобразить график заданной функции:

$$y = e^x / x; 2 \leq x \leq 7; \Delta x = 0.5.$$

1-11. На координатном пространстве размером 300x400 пикселей отобразить график заданной функции:

$$y = \sqrt{x^2 + 3}; 0 \leq x \leq 51; \Delta x = 3.$$

1-12. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = e^x * \cos(x); 0 \leq x \leq 0.9; \Delta x = 0.05.$$

1-13. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \sqrt{1+x^2}; -0.5 \leq x \leq 5; \Delta x = 0.3.$$

1-14. На координатном пространстве размером 300x600 пикселей отобразить график заданной функции:

$$y = -\sqrt{x/(3+x^3)}; 1 \leq x \leq 35; \Delta x = 3.$$

1-15. На координатном пространстве размером 400x500 пикселей отобразить график заданной функции:

$$y = x^3 * (1 + \ln(x)); 2 \leq x \leq 4; \Delta x = 0,4.$$

1-16. На координатном пространстве размером 400x600 пикселей отобразить график заданной функции:

$$y = \sin(1/x); 0.9 \leq x \leq 2.8; \Delta x = 0.2.$$

1-17. На координатном пространстве размером 500x500 пикселей отобразить график заданной функции:

$$y = \sin(1/x) * \cos(x^2); 0.8 \leq x \leq 4.4; \Delta x = 0.2.$$

1-18. На координатном пространстве размером 300x300 пикселей отобразить график заданной функции:

$$y = (x-2)^2; 0.5 \leq x \leq 1.5; \Delta x = 0.2.$$

1-19. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = -(x+1)^2; 0.1 \leq x \leq 0.8; \Delta x = 0.02.$$

1-20. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \frac{5}{x+4}; 5 \leq x \leq 15; \Delta x = 1.$$

1-21. На координатном пространстве размером 400x200 пикселей отобразить график заданной функции:

$$y = (3x^2 + 2x - 1); -0.7 \leq x \leq 1.5; \Delta x = 0.1.$$

1-22. На координатном пространстве размером 500x400 пикселей отобразить график заданной функции:

$$y = 3 - \frac{2}{x^2}; 0.8 \leq x \leq 2.6; \Delta x = 0.3.$$

1-23. На координатном пространстве размером 600x400 пикселей отобразить график заданной функции:

$$y = x/(1+x); 0.1 \leq x \leq 3.5; \Delta x = 0.3.$$

1-24. На координатном пространстве размером 500x500 пикселей отобразить график заданной функции:

$$y = \ln(x^2 + 2); 0.5 \leq x \leq 5; \Delta x = 0.3.$$

1-25. На координатном пространстве размером 400x400 пикселей отобразить график заданной функции:

$$y = e^x + \sin(x); -1 \leq x \leq 1; \Delta x = 0.1.$$

1-26. На координатном пространстве размером 600x600 пикселей отобразить график заданной функции:

$$y = (e^x + 3)/x; 2 \leq x \leq 5; \Delta x = 0.2.$$

1-27. На координатном пространстве размером 700x700 пикселей отобразить график заданной функции:

$$y = e^x + 5; 1 \leq x \leq 4; \Delta x = 0.2.$$

1-28. На координатном пространстве размером 600x400 пикселей отобразить график заданной функции:

$$y = x^2/(1+x); 0.1 \leq x \leq 3.5; \Delta x = 0.4.$$

1-29. На координатном пространстве размером 600x500 пикселей отобразить график заданной функции:

$$y = \ln(x^2 + 2); 0.5 \leq x \leq 6; \Delta x = 0.4.$$

1-30. На координатном пространстве размером 800x800 пикселей отобразить график заданной функции:

$$y = \ln(x^3 + 2); 5 \leq x \leq 15; \Delta x = 0.4.$$

1-31. На координатном пространстве размером 600x600 пикселей отобразить график заданной функции:

$$y = \cos(x^2) / \delta^3; -5 \leq x \leq -2; \Delta x = 0.3$$

1-32. На координатном пространстве размером 600x400 пикселей отобразить график заданной функции:

$$y = x^2 / (1 + x); -5 \leq x \leq -0.2; \Delta x = 0.3.$$

1-33. На координатном пространстве размером 600x500 пикселей отобразить график заданной функции:

$$y = x \ln(x^2 + 2); -5 \leq x \leq 11; \Delta x = 1.$$

1-34. На координатном пространстве размером 600x600 пикселей отобразить график заданной функции:

$$y = 12 \cos(x^2) \sin(x); -9 \leq x \leq -1; \Delta x = 0.5.$$

1-35. На координатном пространстве размером 600x500 пикселей отобразить график заданной функции:

$$y = x \ln(x^2 + 2)e^x; -9 \leq x \leq 1; \Delta x = 0.5.$$

2-1. На координатном пространстве размером 300x400 пикселей отобразить график заданной функции:

$$y = 4 / \sqrt{x^2 + 3}; 0 \leq x \leq 5; \Delta x = 0.2.$$

2-2. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = e^{x+2} * \cos(x-1); 0 \leq x \leq 0.9; \Delta x = 0.1.$$

2-3. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \frac{x^2 + 9}{x^2 - 26}; -0.5 \leq x \leq 5; \Delta x = 0.6.$$

2-4. На координатном пространстве размером 300x600 пикселей отобразить график заданной функции:

$$y = \frac{x}{x^2 + 25}; -5 \leq x \leq 5; \Delta x = 0.8.$$

2-5. На координатном пространстве размером 400x500 пикселей отобразить график заданной функции:

$$y = 2\tilde{\sigma}\sin(\tilde{\sigma}^2); -0.5 \leq x \leq 0.7; \Delta x = 0.1.$$

2-6. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = 3\tilde{\sigma} - \sin(x^2); 0.1 \leq x \leq 0.8; \Delta x = 0.1.$$

2-7. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \cos^2(x - 5); 5 \leq x \leq 10; \Delta x = 0.5.$$

2-8. На координатном пространстве размером 500x500 пикселей отобразить график заданной функции:

$$y = \frac{x - 4}{x^2 + 26}; -5 \leq x \leq 5; \Delta x = 0.8.$$

2-9. На координатном пространстве размером 500x600 пикселей отобразить график заданной функции:

$$y = \frac{x}{x^2 - 35}; -5 \leq x \leq 5; \Delta x = 0.7.$$

2-10. На координатном пространстве размером 500x400 пикселей отобразить график заданной функции:

$$y = x^2 \sin(4\sqrt{x}); 0 \leq x \leq 2; \Delta x = 0.1.$$

2-11. На координатном пространстве размером 400x800 пикселей отобразить график заданной функции:

$$y = (8 + x^2)/(\sin(x) + 4); 0.1 \leq x \leq 1.8; \Delta x = 0.2.$$

2-12. На координатном пространстве размером 500x700 пикселей отобразить график заданной функции:

$$y = \cos^2(x)\sqrt{\delta}; 5 \leq x \leq 10; \Delta x = 0.4.$$

2-13. На координатном пространстве размером 600x600 пикселей отобразить график заданной функции:

$$y = \cos(x^2)\delta^3; -5 \leq x \leq -2; \Delta x = 0.3.$$

2-14. На координатном пространстве размером 400x600 пикселей отобразить график заданной функции:

$$y = x^{(3+\delta)} \cos(\sqrt{x}); 0.2 \leq x \leq 0.9; \Delta x = 0.1.$$

2-15. На координатном пространстве размером 500x800 пикселей отобразить график заданной функции:

$$y = (1 + x^2)/(2^\delta + \cos(\sqrt{x})); 0.1 \leq x \leq 2.9; \Delta x = 0.2.$$

2-16. На координатном пространстве размером 400x700 пикселей отобразить график заданной функции:

$$y = x^2 / (\cos(x) + 6\delta); 0.1 \leq x \leq 2.9; \Delta x = 0.2.$$

2-17. На координатном пространстве размером 800x800 пикселей отобразить график заданной функции:

$$y = \ln(x^2 + 2); 0.5 \leq x \leq 5; \Delta x = 0.4.$$

2-18. На координатном пространстве размером 700x700 пикселей отобразить график заданной функции:

$$y = e^x + 5; -1 \leq x \leq 1; \Delta x = 0.2.$$

2-19. На координатном пространстве размером 800x600 пикселей отобразить график заданной функции:

$$y = (e^x + 4)/x; 2 \leq x \leq 7; \Delta x = 0.5.$$

2-20. На координатном пространстве размером 600x300 пикселей отобразить график заданной функции:

$$y = 4 * \cos(x^2); -5 \leq x \leq 0; \Delta x = 0.4.$$

2-21. На координатном пространстве размером 600x300 пикселей отобразить график заданной функции:

$$y = 4 * x^2 - 12; -5 \leq x \leq 0; \Delta x = 0.5.$$

2-22. На координатном пространстве размером 800x600 пикселей отобразить график заданной функции:

$$y = (\delta^2 + 4) / x; 2 \leq x \leq 7; \Delta x = 0.5.$$

2-23. На координатном пространстве размером 300x600 пикселей отобразить график заданной функции:

$$y = \frac{x}{x^2 + 5}; 5 \leq x \leq 15; \Delta x = 0.8.$$

2-24. На координатном пространстве размером 300x400 пикселей отобразить график заданной функции:

$$y = \frac{5}{x + 6}; 5 \leq x \leq 15; \Delta x = 0.8.$$

2-25. На координатном пространстве размером 400x400 пикселей отобразить график заданной функции:

$$y = (2x^2 + 2x - 1); -0.7 \leq x \leq 1.5; \Delta x = 0.15.$$

2-26. На координатном пространстве размером 400x600 пикселей отобразить график заданной функции:

$$y = 3 - \frac{2}{x^2}; 0.8 \leq x \leq 4.6; \Delta x = 0.4.$$

2-27. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = (8 + x) / (\sin(x) + 4); 0.1 \leq x \leq 1.8; \Delta x = 0.2.$$

2-28. На координатном пространстве размером 600x400 пикселей отобразить график заданной функции:

$$y = x^2 / (\cos(x) - 6); 1 \leq x \leq 15; \Delta x = 1.$$

2-29. На координатном пространстве размером 400x400 пикселей отобразить график заданной функции:

$$y = e^x - x^3; -1 \leq x \leq 1.8; \Delta x = 0.2.$$

2-30. На координатном пространстве размером 400x300 пикселей отобразить график заданной функции:

$$y = e^x * \cos(x); 0 \leq x \leq 1.8; \Delta x = 0.1.$$

2-31. На координатном пространстве размером 500x300 пикселей отобразить график заданной функции:

$$y = \sqrt{1+x^2}; -5 \leq x \leq 5; \Delta x = 0.5.$$

2-32. На координатном пространстве размером 400x500 пикселей отобразить график заданной функции:

$$y = x^3 * (1 + \ln(|x+5|)); -4 \leq x \leq 3; \Delta x = 0,4.$$

2-33. На координатном пространстве размером 500x500 пикселей отобразить график заданной функции:

$$y = \sin(1/x) * \cos(x^2); 0.8 \leq x \leq 4.4; \Delta x = 0.2.$$

2-34. На координатном пространстве размером 300x300 пикселей отобразить график заданной функции:

$$y = (x-2)^2; 0.9 \leq x \leq 4.1; \Delta x = 0.2.$$

После этого производится проверка правильности построения графика средствами пакет Microsoft Excel.

ОРГАНИЗАЦИЯ ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

После создания программного приложения, реализующего задачу курсовой работы, результат демонстрируется студентом преподавателю, руководящему данным процессом, для чего заранее определяются дата и время встречи или это происходит в заранее установленные часы консультации по курсовой работе. Если преподаватель выражает согласие с представленными результатами, то студент переходит к написанию пояснительной записки с указанным выше содер-

жанием. В противном случае студент фиксирует все сделанные преподавателем замечания и согласовывает с ним дату и время следующей встречи.

После написания студентом пояснительной записки последняя с подписью студента и текущей датой передается руководителю работы для просмотра, в ходе которого проверяется как собственно содержимое, общая структура которого представлена ранее в данных методических указаниях, так и правила оформления технической документации. При выявлении недочетов, влияющих на качество курсовой работы, записка возвращается студенту с подробным объяснением сути претензий и назначением срока для их устранения.

Пояснительная записка без замечаний подписывается руководителем курсовой работы с визой «Допущена к защите». Все подписи сопровождаются датами. После этого устанавливается преподавателем единолично или по согласованию со студентом дата защиты курсовой работы.

Защита принимается комиссией из двух-трех человек, назначенной заведующим кафедрой. В неё обязательно входит руководитель разработки, ведущий лектор (если он не является руководителем работы) или специалист по данной проблематике, который может быть как работником кафедры, так и представителем сторонней организации, являющийся экспертом по данной проблематике.

В ходе защиты студент демонстрирует работу программного приложения и отвечает на вопросы как по листингу, так и по другим разделам пояснительной записки. На основании защиты определяется окончательная оценка курсовой работы, которая выставляется в ведомость и зачетную книжку, а также проставляется на титульном листе пояснительной записки с датой и подписью всеми членами комиссии.

Курсовая работа оценивается по следующим параметрам:

1. соответствие содержания работы ее теме;

2. тщательность проработки материала (постановки задачи, обоснования модульной структуры программы, выбора алгоритмов);
3. самостоятельность при выполнении работы;
4. стиль изложения работы, ее логика и качество формулировок в ее выводах;
5. соответствие оформления курсовой работы требованиям;
6. соблюдение правил орфографии и пунктуации;
7. качество оформления графического материала;
8. ответы на вопросы в процессе защиты.

Нормы оценки курсовой работы:

1. *Оценку «отлично» получают работы, в которых:*
 - a. используется материал, освоенный студентом дополнительно в русле самостоятельного расширения знаний по изучаемой дисциплине, т. е. очевидна самостоятельность студента в выполнении работы, предлагаются обоснованные методы решения поставленной задачи, дается аргументированный критический анализ используемого теоретического и практического материала на базе глубоких знаний технической информации по данной теме.
 - b. материал пояснительной записки излагается грамотно, логично, последовательно.
 - c. оформление отвечает требованиям написания курсовой работы.
 - d. во время защиты студент показал умение кратко, доступно (ясно) представить результаты работы, адекватно ответить на поставленные вопросы, обосновать (защитить) выбранные методы решения проблемы.
2. *Оценка «хорошо» ставится, если:*
 - a. в работе, выполненной на достаточном теоретическом и практическом уровне, полно и всесторонне решаются вопросы поставленной проблемы, но нет должной степени творчества.

- b. студент показал знание теоретического материала по рассматриваемой проблеме, однако умение анализировать, аргументировать свою точку зрения, делать обобщения и выводы вызывает у него затруднения.
- c. материал пояснительной записки не всегда излагается логично, последовательно. Имеются недочеты в оформлении курсовой работы.
- d. во время защиты студент показал умение кратко, доступно (ясно) представить результаты исследования, однако затруднялся отвечать на поставленные вопросы.

3. *Оценку «удовлетворительно» имеют работы:*

- a. правильно, но не в полной мере, используются теоретические и практические знания по изучаемой дисциплине, не проявилось умение логически стройного их использования, самостоятельного анализа источников, предлагаемые решения не содержат новизны, встречаются отдельные ошибки в решении поставленной в ходе проектирования задачи.
 - b. студент испытывает затруднения при необходимости анализировать, аргументировать свою точку зрения, делать обобщение и выводы.
 - c. материал не всегда излагается логично, последовательно.
 - d. имеются недочеты в оформлении курсовой работы.
 - e. во время защиты студент затрудняется в представлении результатов работы и ответах на поставленные вопросы.
4. *Оценку «неудовлетворительно» студенты получают в случае* когда не могут ответить на вопросы преподавателей во время защиты, не владеют материалом работы, не в состоянии дать объяснения выводам и практическим результатам по решаемой проблеме.

Список использованных источников

1. Положение об организации методической работы института [Электронный ресурс] / ФГБОУ «КГТУ». – 2021. – Электрон. дан. – Режим доступа: https://klgtu.ru/upload/sveden/document/polozhenie_org_met_rab_instituta.pdf (дата обращения: 27.02.2022).
2. Стандарты Единая Система Программной Документации (ЕСПД) [Электронный ресурс] / ВШЭ. Факультет компьютерных наук. Центр практик и проектной работы. – 2018-2019. – Электрон. дан. – Режим доступа: <https://www.hse.ru/data/2018/11/28/1144395257/ГОСТы%20и%20ЕСПД%202018-2019.pdf> (дата обращения: 27.02.2022).
3. Основы программирования на языке Python : учебное пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург : Изд-во Урал. ун-та, 2014. – 91, [1] с. – Электрон. дан. – Режим доступа: https://elar.urfu.ru/bitstream/10995/28769/1/978-5-7996-1198-9_2014.pdf (дата обращения: 27.02.2022).
4. Технология программирования: учебное пособие / А.Ю. Рогов, О.В. Проститенко – Санкт-Петербург: СПбГТИ(ТУ). 2010. – 112 с. – Электрон. дан. – Режим доступа: http://sa.technolog.edu.ru/repository/titp_book.pdf (дата обращения: 27.02.2022).
5. Технология программирования: учеб. пособие / Г. С. Иванова – Москва: Изд-во: КноРус, 2016. – 334 с. - Электрон. дан. – Режим доступа: <https://litgu.ru/knigi/programming/2806-tehnologiya-programmirovaniya.html> (дата обращения: 27.02.2022).

ПРИЛОЖЕНИЕ I

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО РЫБОЛОВСТВУ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Калининградский государственный технический университет»

Институт цифровых технологий

Кафедра _____
наименование кафедры

Курсовая работа
допущена к защите
Руководитель: _____
(уч. степень, звание, должность*)
_____ И.О. Фамилия
«__» _____ 202__ г.

Курсовая работа защищена
с оценкой _____
Руководитель: _____
(уч. степень, звание, должность)
_____ И.О. Фамилия
«__» _____ 202__ г.

ТЕМА КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине
«Наименование дисциплины»
КР.ХХ¹.ХХ.ХХ.ХХ².Х³.Х⁴

Работу выполнил:
студент гр. ____
_____ И.О. Фамилия
«__» _____ 20__ г.

Калининград
202__

ПОЯСНЕНИЯ

Обозначения в шифре

КП.ХХ¹.ХХ.ХХ.ХХ².Х³.ХХ⁴.ПЗ

КР – курсовая работа.

КП – курсовой проект.

ХХ¹ – номер кафедры.

ХХ.ХХ.ХХ²– шифр направления подготовки

Х³ – последняя цифра года, когда выполнена работа (например, 2022 год, будет цифра 2).

ХХ⁴ – номер варианта курсовой работы(проекта).

ПЗ – пояснительная записка

*Ученую степень и звание следует сокращать в соответствии с рекомендациями Министерства науки РФ, например:

Сокращение

Полное написание

Учёные степени

д-р биол. наук

доктор биологических наук

д-р с.-х. наук

доктор сельскохозяйственных наук

д-р техн. наук

доктор технических наук

канд. с.-х. наук

кандидат сельскохозяйственных наук

канд. техн. наук

кандидат технических наук

канд. хим. наук

кандидат химических наук

Учёные звания

доц.

доцент

проф.

профессор

Пример рубрикации

Содержание

1. Цели и задачи курсовой работы	2
2. Обоснование выбора языка Python для написания курсовой работы	3
3. Техническое задание	4
3.1 Обоснование для разработки	6
3.2 Описание задания	9
.....	
13. Руководство системного программиста	34
13.1 Требования к информационной и программной совместимости	37
13.2 Требования к составу и параметрам технических средств	39
14. Руководство пользователя	43
.....	
Список использованной литературы	47

Пример аннотации

Аннотация

Представленная курсовая работа состоит из введения, трех глав, заключения, списка литературных источников и приложений.

В первой части работы рассматривается подробная характеристика главных источников права, а также, источников права, используемых в различных правовых семьях. Дана сводная классификация. В следующей главе детально анализируется каждый из видов правовых источников и делается вывод на основании изложенного материала.

Ключевые слова: источники права, правовые семьи.

Работа состоит из 31 страницы, содержит 43 литературных источника, 3 таблицы и 8 приложений.

Правила оформления блок-схем

Схема - это абстракция какого-либо процесса или системы, наглядно отображающая наиболее значимые части. Соглашения выработаны для изображения схем-алгоритмов и закреплены ГОСТами и международными стандартами.

На территории Российской Федерации действует единая система программной документации (ЕСПД), частью которой является Государственный стандарт — **ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем» [1].**

В приложении приведена только часть по разработки схем алгоритмов программ. Рассматриваемый ГОСТ практически полностью соответствует международному стандарту **ISO 5807:1985.**

Элементы блок-схем алгоритмов

Блок-схема представляет собой совокупность символов, соответствующих этапам работы алгоритма и соединяющих их линий.

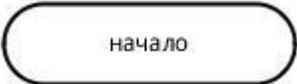
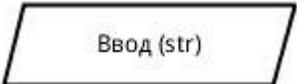
Пунктирная линия используется для соединения символа с комментарием.

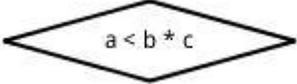
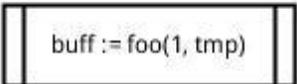
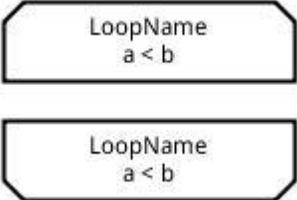
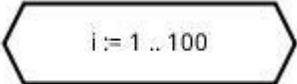
Сплошная линия отражает зависимости по управлению между символами и может снабжаться стрелкой. Стрелку можно не указывать при направлении дуги слева направо и сверху вниз.

Линии должны подходить к символу слева, либо сверху, а исходить снизу, либо справа.

Есть и другие типы линий, используемые, например, для изображения блок-схем параллельных алгоритмов

Далее рассмотрены лишь основные символы, которых вполне достаточно

 <p>Терминатор начала и конца работы функции</p>	<p>Терминатором начинается и заканчивается любая функция. Тип возвращаемого значения и аргументов функции обычно указывается в комментариях к блоку терминатора.</p>
 <p>Операции ввода и вывода данных</p>	<p>В ГОСТ определено множество символов ввода/вывода, например вывод на магнитные ленты, дисплеи и т.п. Если источник данных не принципиален, обычно используется символ параллелограмма. Подробности ввода/вывода могут быть указаны в комментариях.</p>

 <p>Выполнение операций над данными</p>	<p>В блоке операций обычно размещают одно или несколько (ГОСТ не запрещает) операций присваивания, не требующих вызова внешних функций.</p>
 <p>Блок, иллюстрирующий ветвление алгоритма</p>	<p>Блок в виде ромба имеет один вход и несколько подписанных выходов. В случае если блок имеет 2 выхода (соответствует оператору ветвления), на них подписывается результат сравнения — «да/нет». Если из блока выходит большее число линий (оператор выбора), внутри него записывается имя переменной, а на выходящих дугах — значения этой переменной.</p>
 <p>Вызов внешней процедуры</p>	<p>Вызов внешних процедур и функций помещается в прямоугольник с дополнительными вертикальными линиями.</p>
 <p>Начало и конец цикла</p>	<p>Символы начала и конца цикла содержат имя и условие. Условие может отсутствовать в одном из символов пары. Расположение условия, определяет тип оператора, соответствующего символам на языке высокого уровня — оператор с предусловием (while) или постусловием (do ... while).</p>
 <p>Подготовка данных</p>	<p>Символ «подготовка данных» в произвольной форме (в ГОСТ нет ни пояснений, ни примеров), задает входные значения. Используется обычно для задания циклов со счетчиком.</p>
 <p>Соединитель</p>	<p>В случае если блок-схема не уместится на лист, используется символ соединителя, отражающий переход потока управления между листами. Символ может использоваться и на одном листе, если по каким-либо причинам тянуть линию неудобно.</p>

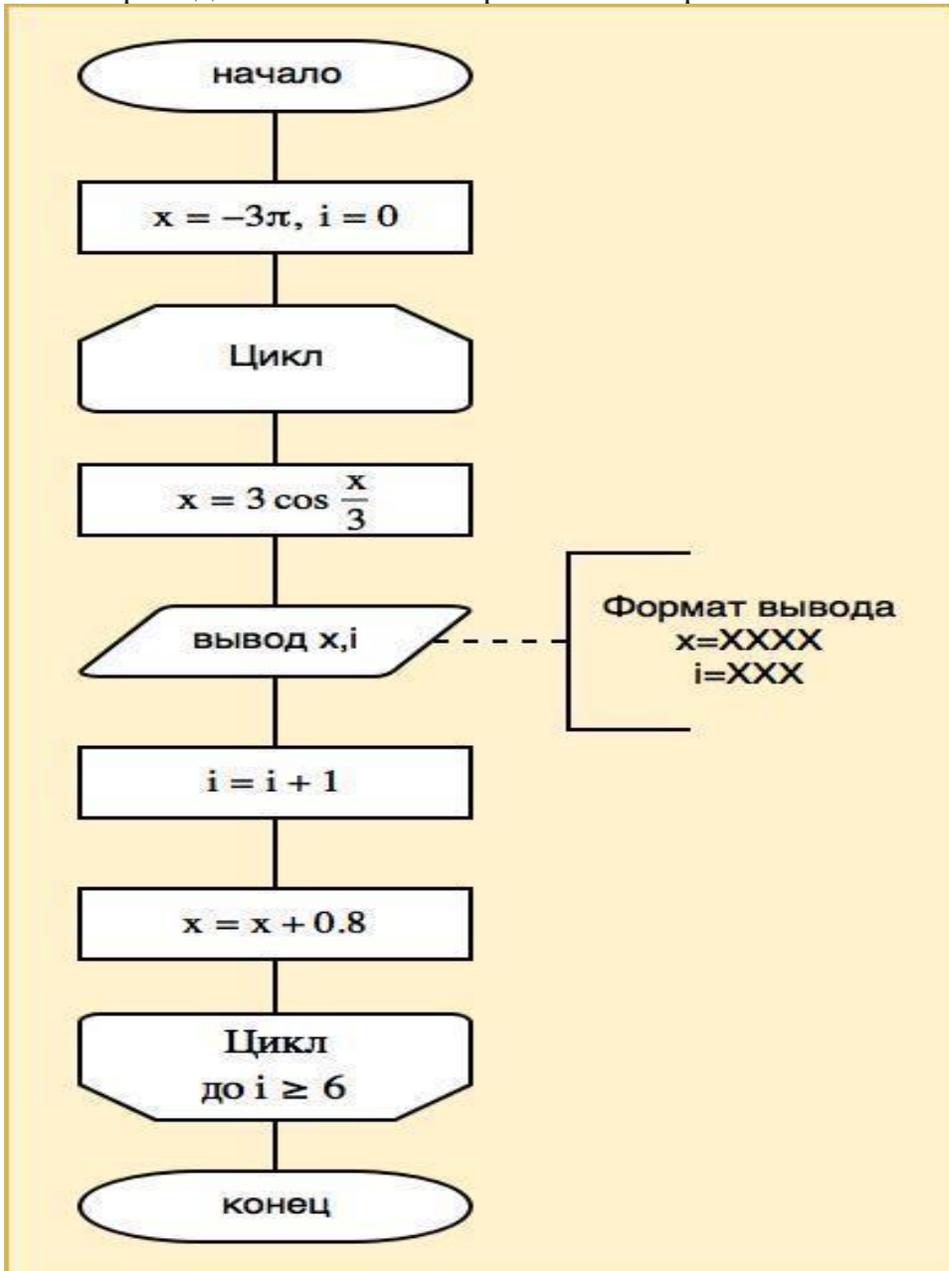
foo(i, str)
выделяет i-тый символ
строки str

Комментарий

Комментарий может быть соединен как с одним блоком, так и группой. Группа блоков выделяется на схеме пунктирной линией.

Примеры блок-схем

Ниже приведена блок-схема простого алгоритма



Часто для оформления блок-схем используются возможности *Microsoft Word*, но это малоудобно, например в *MS Word* нет стандартного блока для терминатора начала и конца алгоритма (прямоугольник со скругленными краями, а не овал). Более удобными являются утилиты *MS Visio* и *yEd*.

Локальный электронный методический материал

Леонид Григорьевич Высоцкий

ВЫСОКОУРОВНЕВЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ (ВТП)

Редактор Г.А. Смирнова

Уч.-изд. л. 2,1. Печ. л. 2,6.

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