

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

Л. Г. Высоцкий, Н. Б. Розен

**ВЫСОКОУРОВНЕВЫЕ ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ (ВТП)**

Утверждено редакционно-издательским советом ФГБОУ ВО «КГТУ»
в качестве учебно-методического пособия по выполнению курсовой работы
по дисциплине «Высокоуровневые технологии программирования»
для студентов бакалавриата по направлениям подготовки
09.03.01 Информатика и вычислительная техника,
09.03.03 Прикладная информатика

Калининград
Издательство ФГБОУ ВО «КГТУ»
2023

Рецензент
доцент кафедры прикладной информатики ФГБОУ ВО «Калининградский
государственный технический университет»
С. А. Калинина

Высоцкий Л. Г., Розен Н. Б.

Высокоуровневые технологии программирования (ВТП): учеб.-метод. пособие по выполнению курсовой работы для студ. бакалавриата по напр. подгот. 09.03.01 Информатика и вычислительная техника, 09.03.03 Прикладная информатика / Л. Г. Высоцкий, Н. Б. Розен. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2023. – 33 с.

Учебно-методическое пособие является руководством по выполнению курсовой работы по основам высокоуровневых технологий программирования студентами, обучающимися по направлениям подготовки 09.03.01 Информатика и вычислительная техника, 09.03.03 Прикладная информатика. Курсовая работа предназначена для закрепления теоретического материала и приобретения практических навыков формирования графического пользовательского интерфейса программного приложения, реализующего заданный алгоритм, на основе визуального и событийного программирования.

Табл. 1, список лит. – 5 наименований

Учебно-методическое пособие рассмотрено и одобрено методической комиссией ученого совета Института цифровых технологий 20 сентября 2022 г., протокол № 06

ОГЛАВЛЕНИЕ

ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ	4
УСЛОВИЯ ВЫБОРА ТЕМЫ И ПОРЯДОК РАЗРАБОТКИ КУРСОВОЙ РАБОТЫ.....	5
ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ	7
МЕТОДИКА ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	9
ТЕМЫ И ЗАДАНИЯ ПО КУРСОВОЙ РАБОТЕ	15
ОРГАНИЗАЦИЯ ЗАЩИТЫ КУРСОВОЙ РАБОТЫ.....	22
СПИСОК ИСТОЧНИКОВ.....	25
ПРИЛОЖЕНИЕ 1	26
ПРИЛОЖЕНИЕ 2.....	28
ПРИЛОЖЕНИЕ 3.....	29
ПРИЛОЖЕНИЕ 4.....	30

ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Основной целью выполнения курсовой работы по дисциплине «Высокоуровневые технологии программирования» является формирование определенных умений и навыков, углубление и обобщение знаний, получаемых 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, применять их при решении конкретной задачи, связанной 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 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.

Курсовая работа ориентирована, прежде всего, на освоение студентами практических навыков формирования графического пользовательского интерфейса программным путем средствами модуля tkinter языка Python. Именно с конструирования данного интерфейса начинается создание большинства современных больших программных систем, а далее уже к нему привязывается весь функционал, реализуемый системой в соответствии с требованиями.

В результате выполнения курсовой работы студент должен расширить свои знания в области:

- средств и возможностей языка реализации проекта;
- типовых элементов графического пользовательского интерфейса (GUI);
- 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 РАЗРАБОТКИ КУРСОВОЙ РАБОТЫ

Подавляющее большинство программных приложений в настоящее время представляют собой информационные системы. Общение пользователя с такими системами реализуется через графический интерфейс. Поэтому основная цель данной курсовой работы – повышение умений и знаний студентов в разработке таких интерфейсов на основе возможностей современных систем программирования.

Тема курсовой работы выбирается студентом из предложенного списка вариантов, приведенного в учебно-методическом пособии. Студент получает право на выбор темы после выполнения и защиты хотя бы одной лабораторной работы по данному курсу. После согласования темы курсовой работы с преподавателем она закрепляется за студентом в списке. Приветствуется предложение собственных тем, соответствующих тематике данной курсовой работы, при обязательном согласовании таких тем с преподавателем.

Все темы работы носят индивидуальный характер. Вместе с тем, общим для них является, как указывалось ранее, разработка графического

пользовательского интерфейса, к которому привязывается функционал, реализующий требуемую задачу. Ниже приведены три примерных темы курсовой работы и соответствующие задания к ним:

1. *«Построитель графиков»*. На окне определенного размера строятся графики по введенным данным. Количество рядов от 2 до 5. Количество точек от 5 до 15. Цвет линии и ее тип предварительно выбираются из фиксированных наборов. Существует возможность ввода оглавления осей и названия всей диаграммы.

2. *«Тренажер отработки реакции»*. На тестовом поле на весь экран в случайном месте появляется окружность. Необходимо как можно быстрее щелкнуть мышью по окружности, после чего появляется следующая окружность. Диаметр окружности регулируются (от 15 до 50 пикселей). Учитывается количество попаданий мыши по окружности за определенное время (не менее минуты), которое задается исходно.

3. *«Охранный система»*. Отображаются три кольца защиты некоторого объекта в центре, во внутреннем кольце только одна дверь, в среднем две, во внешнем – три. Каждая дверь открывается только при закрытых остальных. В праздничные и выходные дни для службы безопасности открываются только двери внешнего кольца.

Процесс выполнения курсовой работы предполагает следующую очередность этапов:

- Из предложенного списка вариантов выбрать тему курсовой работы или согласовать с преподавателем собственную тему. Список согласованных тем утверждается на заседании кафедры, и затем формулировка темы не меняется.

- Разработать обобщенную блок-схему работы программного приложения.

- Изучить и проанализировать литературу и другие источники по теме работы.

- Разработать интерфейс системы (видеоформы), обосновывая выбор каждого виджета, и связать компоненты интерфейса с реализуемыми приложением функциями.

- Для каждого отдельного модуля (функции) листинга разработать алгоритм в виде подробной блок-схемы.

- Провести кодирование и отладку программной системы.

- Добавить к проекту отдельное окно со сведениями об авторе, вызываемое командой главного меню.

- Максимально снабдить листинг комментариями.

- 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 иметь на данном курсе, и о наличии у него 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. вместо «я считаю» – «считается». В курсовой работе должны присутствовать: титульный лист (Приложение 1), задание на курсовую работу, содержание (Приложение 2), 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 – не менее 1 страницы. 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 – подробное описание задания (рекомендуется исходно обсудить с руководителем курсовой работы во избежание впоследствии возможных недоразумений) и постановка задачи с представлением основных функций, интерфейса и результатов.

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

Вид программного интерфейса – вид визуальных форм, необходимых при функционировании программы, с указанием вида и расположения используемых виджетов, их функционального назначения, и описание способов обработки ошибок (реакции на ошибки пользователей).

Блок-схемы модулей – подробная графическая схема каждого алгоритма, 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. Поскольку практически все темы курсовой работы предполагают, что результаты функционирования разрабатываемых программных приложений представляются в виде последовательности графических статических или динамических изображений на экране компьютера, результат тестового задания целесообразно представить в виде скриншота одной из таких ситуаций.

Требования программы к техническому и программному обеспечению содержат описание обязательных технических устройств и описание требований к системному программному обеспечению, в том числе необходимых библиотек Python, без которых функционирование программы невозможно.

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 работы задачам.

Список использованных источников представляет собой перечень использованных при написании курсовой работы разнообразных источников. Объем списка должен быть достаточным и обычно составляет не менее 10 единиц. 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 ГОСТ 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 1». 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. 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 литературу и другие источники по теме

Поскольку сердцевина курсовой работы – формирование графического пользовательского интерфейса, студенту рекомендуется тщательно проработать лекционный и практический материал по графическим средствам изучаемого языка программирования (Python'a), проработать дополнительные источники по этой теме, поскольку Python является достаточно динамичным языком, который периодически пополняется новыми средствами обработки как текстовой, так и графической информации.

Реализуемый в языке программирования тип графики предполагает свои методы и алгоритмы ее обработки, поэтому студенту рекомендуется проработать типовые средства работы с этой графикой в данном языке.

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

В конечном итоге любое программное приложение предназначено для реализации определенного функционала, т. е. одной или нескольких функций, объединенных некоторой общей целью. Целесообразно результат данного этапа работы оформить в виде дерева, «корнем» которого является функция, реализующая целевое назначение всего программного приложения, а само дерево – результат декомпозиции данной функции в виде ветвей и «листьев» – подфункций, предполагающих уже реализацию в виде процедур и логических блоков на языке листинга.

Если функции нижнего уровня предназначены для обработки данных, целесообразно четко сформулировать характеристики их входной и выходной информации: допустимый диапазон, точность, форму представления. Эти

характеристики в дальнейшем являются основой для эффективного проектирования пользовательского графического интерфейса, выбора используемых виджетов, обоснования значений их характеристик.

3. Спроектировать графический интерфейс приложения

Сформированный функционал служит основой для выбора виджетов, образующих пользовательский интерфейс. Такой интерфейс предполагает, в общем случае, использование трех видов виджетов: для ввода исходных данных, вывода результатов работы, приложения и активизации требуемых действий путем генерации соответствующих событий.

Если исходные данные представляют собой фиксированный набор дискретных значений относительно небольшого диапазона, для их ввода целесообразно выбирать виджеты типа Listbox (Список), Combobox (Выпадающий список), Radiobutton (Набор радиокнопок), Scale (Линейка прокрутки), Spinbox (Список выбора). При этом Combobox и Spinbox характеризуются минимальной площадью на пространстве формы, что важно при большой загруженности окна интерфейса, а Scale используется для задания только числовых значений. С другой стороны, Scale может ориентироваться как по горизонтали, так и по вертикали, что в некоторых случаях повышает наглядность, придает семантику вводимым данным. Spinbox также может использоваться и для ввода дискретных числовых значений больших диапазонов.

Логические (бинарные) значения обычно задаются посредством Checkbutton'a (флажка).

Если исходные данные организованы в виде иерархических деревьев, для их представления и выбора в программном приложении необходимо организовать меню: главное, контекстное или выпадающее. В этом случае важной является организация структуры такого дерева с точки зрения минимизации времени поиска требуемого для ввода значения. Рекомендуются, чтобы на каждом уровне (ветви) меню было не более десяти значений.

Если же предполагается ввод вещественных значений, становится неизбежным использование текстовых редакторов. В этом случае следует задавать комфортные для пользователя параметры используемых виджетов (тип и размер шрифта, его цвет и цвет фона и т. п.). Обязателен программный контроль значений таких данных.

Надо также учитывать, что у некоторых виджетов, используемых для ввода данных, уже есть встроенные средства для создания рядом поясняющих их назначение надписей (комментариев). У других же виджетов для этих же целей требуется вводить в состав интерфейса дополнительные средства (обычно в виде меток), что усложняет его.

Многие из названных ранее средств для ввода также могут использоваться и для вывода информации. При этом выводимые текстовые данные, т. е. динамические, должны обязательно отделяться от статических (постоянно находящихся на форме надписей и комментариев) местоположением, цветом, размером и т. п. Для вывода же графических изображений, как статических, так и динамических, могут использоваться специализированные контейнеры типа Canvas, хотя статическая графика может выводиться также и поверх других виджетов (кнопок, меток). При этом целесообразно использовать минимальный набор цветовых оттенков для отображения такой информации, поскольку пестрая цветовая гамма быстро утомляет пользователя.

Для генерации событий, запускающих те или иные действия в программном приложении, чаще всего используются различные кнопки или меню. Последние обычно применяются для активации редко используемых действий. Они также часто дублируются так называемыми «горячими клавишами». В качестве таковых следует выбирать сочетание двух или трех клавиш, так как одиночная алфавитно-цифровая клавиша при своем прямом использовании при вводе текста одновременно может запустить и приписанное ей действие как дублера опции меню.

При активации действий через кнопки целесообразно использовать двойной щелчок по ней мыши, так как однократное нажатие может оказаться случайным при оперировании мышью и вызвать не требуемое в данный момент действие.

В конечном итоге спроектированный интерфейс должен обеспечить ввод всех необходимых данных для каждой функции из сформированного ранее списка, вывод результатов работы этих функций и инструментарий для их активизации (запуска). При этом все перечисленные средства должны пространственно достаточно свободно располагаться на форме, чтобы минимизировать проблемы при реконфигурации интерфейса (добавлении новых виджетов, изменении их размеров и положения).

4. Разработать алгоритмы используемых функций

К моменту написания данной курсовой работы студенты представленных здесь направлений должны знать и уметь использовать на практике около десятка базовых алгоритмов обработки информации: упорядочение массива по возрастанию/убыванию, нахождение минимального/максимального значения, нахождение сумма/среднего/произведения и т. д. Разработка алгоритмов программного приложения в курсовой работе должна опираться именно на этот теоретический задел с учетом тех специфических особенностей, которые присущи данной работе.

На этом же этапе необходимо предусмотреть реакцию приложения на различные исключительные ситуации, возникающие в ходе работы: ввод некорректных данных, проблемы с используемыми файлами и т. п. Наиболее эффективно выявлять и управлять такими ошибками с помощью конструкций `try .. except`, которые в настоящее время включены практически во все современные языки программирования. Правильно спроектированный пользовательский интерфейс должен сопровождать подобные ситуации выдачей диагностических сообщений, в которых разъясняется суть проблемы и, возможно, предлагаются меры по её устранению. Если некоторые (все) сообщения могут повторяться в разных местах приложения, целесообразно оформить эти сообщения в виде массива (в процессе работы приложения они будут выводиться по индексам), что устраним дублирование сообщений. Такой подход упрощает и локализацию приложений в других языковых средах, что становится обыденным для промышленного производства программных продуктов в настоящее время.

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, 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 программного обеспечения 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 для рассматриваемых здесь направлений 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 легко

читаемые 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 слабо используемые *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.

6. Разработать контрольный пример

Поскольку основной задачей курсовой работы является разработка пользовательского интерфейса для выполнения некоторой задачи, то контрольный пример, в общем случае, может представлять один или два скриншота с отображением исходных данных задачи и полученным для них результатом.

7. Оформить работу в соответствии с предъявляемыми 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. Страницами считаются 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 к оформлению:

- использовать автоматическую нумерацию страниц;
- содержание должно быть вставлено автоматически;
- желательно использовать колонтитулы;
- сокращения должны быть расшифрованы.

8. Обобщить результаты и сформулировать выводы

В данном разделе пояснительной записки формулируются практические и теоретические результаты выполнения курсовой работы (название созданной программной системы, ее функции, перечень решенных при проектировании практических вопросов), определяются дополнительные опыт, знания и умения, которые были получены студентом в ходе этой работы.

ТЕМЫ И ЗАДАНИЯ ПО КУРСОВОЙ РАБОТЕ

Список вариантов тем курсовых работ, предлагаемых студентам, представлен в таблице 1.

Таблица 1 – Темы курсовых работ

№ п/п	Тема работы
1	2
1	«Построитель графиков». На окне определенного размера строятся графики по введенным данным. Количество рядов от 2 до 5. Количество точек от 5 до 15. Цвет линии и ее тип предварительно выбираются из фиксированных наборов. Возможность ввода оглавления осей и названия всей диаграммы
2	«Построитель круговых диаграмм». На окне определенного размера строится по введенным данным и названиям сегментов круговая диаграмма. Количество значений от 5 до 10. Для них задаются абсолютные значения, а программа определяет количество процентов. На диаграмме указываются названия сегментов. Каждый сегмент закрашивается своим цветом. Возможно выделение отдельного сегмента
3	«Система тестирования сочетаемости цветов» Тестируемое поле занимает не менее 2/3 экрана. Исходно задается цвет фона и цвет текста (не менее 8 значений на каждый цвет). Случайным образом задается место вывода и выводимый текст (слово из 5 случайных символов). Размер шрифта – 12 пунктов. Испытуемый должен повторить текст. Время между выводом на экран и концом ввода фиксируется. Задается количество повторений теста (не менее 20). Определяется и выводится после тестирования среднее время задержки ответа

1	2
4	«Тренажер отработки реакции». Нажатием клавиши «пробел» сверху запускается падающий шарик. Надо щелкнуть мышью в момент совмещения шарика с линией. Скорость падения шарика должна задаваться в определенном диапазоне, а место его появления вверху экрана может генерироваться случайным образом. Линия тоже двигается. Возможны два варианта: движение снизу и сверху. В первом случае также задается скорость перемещения линии, во втором – скорость перемещения линии в два раза меньше заданной скорости перемещения шарика, а начальное положение линии находится на 2/3 высоты экрана. Замеряется и выводится расстояние между линией и центром шарика в момент щелчка мыши. Также рассчитывается среднее значение этой величины по серии экспериментов. Исходно задается число (не менее 10) попыток
5	«Тренажер отработки реакции». На тестовом поле на весь экран в случайном месте появляется окружность. Необходимо как можно быстрее щелкнуть мышью по окружности, после чего появляется следующая окружность. Диаметр окружности регулируются (от 15 до 50 пикселей). Учитывается количество попаданий мыши по окружности за определенное время (не менее минуты), которое задается
6	«Игра «морской бой» с компьютером». Компьютер выбирает ходы случайным образом по правилам игры. Можно добавить элементы «искусственного интеллекта»
7	«Игра в «крестики-нолики». Для двух участников. Поле размером не менее 40x40. Компьютер определяет наличие пяти одинаковых символов в линию
8	«Охранная система». Отображаются три кольца защиты некоторого объекта в центре, во внутреннем кольце только одна дверь, в среднем – две, во внешнем – три. Каждая дверь открывается только при закрытых остальных. В праздничные и выходные дни для службы безопасности открываются только двери внешнего кольца
9	«Электронное табло» Вводится текст, который в дальнейшем непрерывно отображается с различными видеоэффектами (движение по кольцу, двунаправленное движение, увеличение размера текста и т.п., не менее 10 видеоэффектов). Возможность совмещения видеоэффектов и/или программирование последовательности
10	«Игра «Ханойская башня». Количество колец не менее пяти
11	«Визуальная иллюстрация сортировки массива перестановкой». Количество элементов массива в диапазоне 8 ÷ 15. Возможны два варианта выполнения: пошагово по команде пользователя и в автоматическом режиме
12	«Визуальная иллюстрация сортировки массива методом «пузырька». Количество элементов массива в диапазоне 8 ÷ 15. Возможны два варианта выполнения: пошагово по команде пользователя и в автоматическом режиме
13	«Лабиринт». Схема лабиринта насчитывает не менее 20 поворотов, ширина дорожек равна 70 пикселям. На входе шар диаметром 40 пикселей, который можно перемещать по дорожкам с помощью четырех клавиш клавиатуры и фиксации времени прохождения лабиринта

1	2
14	«Бомбардировщик». Вверху экрана слева направо движется стилизованный самолет, скорость которого в определенных пределах может меняться случайным образом при каждом запуске. Внизу справа налево движется стилизованный танк. Необходимо попасть бомбой из самолета по танку
15	«Поиск критического пути». Создается оргграф в виде сети с одной начальной и одной конечной вершинами и не менее 10 промежуточных вершин. Для каждой вершины задается вес и смежность с другими вершинами. Надо найти, просчитать и выделить критический путь в графе
16	«Часы». Компьютерные часы в виде шести семисегментных индикаторов, отображающие часы, минуты и секунды. Пары индикаторов разделены двоеточиями. Часы работают в трех режимах: компьютерное время, секундомер и таймер. Секундомер и таймер можно устанавливать в начальное состояние, запускать и останавливать на паузу
17	«Кафедра». Справочная система по кафедре: план кафедры с указанием номера кабинета, щелчок по кабинету приводит к отображению во всплывающем окне подробной информации о нем (фамилии преподавателей с фотографиями, если это преподавательская, или назначение класса с расписанием занятий)
18	«Калькулятор». Максимально повторяются функции калькулятора ОС Windows
19	«Электроцепь». Графическое моделирование электрической цепи, включающей источник постоянного напряжения (10 В), резистор (5 КОм), далее параллельное включение сопротивления 7 Ком и реостата (в виде компонента Scale), сопротивление которого может регулироваться в диапазоне 4÷8 КОм, амперметр и подключаемый к параллельно реостату вольтметр. Амперметр и вольтметр показывают текущие значения тока и напряжения. Напряжение постоянного источника также предварительно устанавливается стрелками \uparrow , \downarrow с дискретом 0,1 В. Вся графика выполняется средствами языка программирования
20	«Интерпретатор арифметических выражений». Должен включать не менее 7 числовых значений. Выражение может включать четыре арифметические операции и две тригонометрические (синус и косинус). Расчет выполняется пошагово с отображением выражения после каждого упрощения
21	«Управление лифтом». Стилизованное изображение лестничного пролета дома на 9 этажей и шахты лифта. На каждом этаже кнопка вызова лифта. Отдельно пульт лифта, который исходно не виден. Нажатие кнопки некоторого этажа приводит к перемещению лифта на этот этаж, открыванию дверей и визуализации пульта. Нажатие на пульте приводит к закрытию дверей, перемещению на требуемый этаж и открыванию дверей на 15 сек, если лифт не вызван
22	«Звездное небо». Фрагмент звездного неба, на котором размещено не менее 10 созвездий. В отдельном поле перечислены их названия. При выборе названия данное созвездие становится более ярким. При этом звезды созвездия соединяются пунктиром. Предусмотрен возврат в исходное состояние
23	Задача расстановки восьми ферзей

1	2
24	«Фауна». Карта Калининградской области, отдельно фигуры 10 представителей фауны. Выбор одного из представителей приводит к их отображению на карте в местах обитания. Щелчок по карте приводит к показу животных, обитающих в этой местности
25	«Торговые фирмы». Карта Калининграда, отдельно название 10 торговых фирм. Выбор фирмы приводит к отображению на карте магазинов фирмы с указанием их адресов (на каждую фирму не мене 4 магазинов)
26	«Логика». Три логических элемента, расположенных в два уровня. На первом уровне элементы на три и четыре входа, их выходы подаются на вход двухходового элемента. Исходно задаются виды логических функций для элементов (И, ИЛИ, И-НЕ, ИЛИ-НЕ) и логические значения входов. В два такта определяется результат на выходе двухходового элемента
27	«Двоично-десятичный дешифратор». Схематично изображен дешифратор на четыре двоичных входа и десять выходов. На вход подается одна из возможных комбинаций и отображается результат работы схемы
28	«Тренировка внимательности». Матрица 5x6, в которую заносятся случайно пятнадцать пар букв. Исходно они визуальнo заменяются знаками вопроса. При щелчке мыши показывается буква. При щелчке на следующую букву предыдущая буква снова исчезает. При последовательном щелчке по одинаковым буквам они удаляются из матрицы. Фиксируется время игры. Три лучших результата запоминаются
29	«Ректорат». Приводится план части 2-го этажа ГУК, где расположен ректорат. При щелчке по определенному кабинету указывается «хозяин» кабинета, его портрет, телефон, часы приема
30	«Конкур». По всему экрану расставлены в произвольном порядке 15 ворот шириной 40 пикселей и указано направление их прохождения. Слева экрана находится линия старта, а справа – финиша. Исходно колобок диаметром 30 пикселей находится на линии старта. Стрелками ←, ↑, → и ↓ перемещая колобок надо пройти все ворота и достичь финиша. Фиксируется время прохождения и лучшие результаты
31	«Решение задачи коммивояжера» методом "ближайшей свободной вершины" с графической иллюстрацией. Исходно задается количество точек, оно должно быть не менее десяти. Координаты вершины задаются мышью на координатной системе, а далее задается начальная вершина и строится маршрут с определением его длины
32	«Экспертный опрос». Обработка результатов экспертного опроса (лабораторная работа № 2 по Программной инженерии). Количество экспертов – не менее 6, количество задач – не менее 10
33	«Футбольный чемпионат». Для шести футбольных команд приведено расписание игр. Выбор команды приводит к выделению цветом ее игр. Расписание можно менять
34	«Расписание студента». Список из 10 студентов разных групп. При выборе конкретного студента выводится его расписание по дням в виде таблицы. Расписание можно менять

1	2
35	«Нахождение максимина». Визуализация нахождения максимального/минимального элемента в массиве от 10 до 20 элементов. Переключение режима нахождения
36	«Фонотека». Включает не менее 15 песен. При выборе произведения показывается портрет исполнителя и его творческая биография. Звучит сама песня. Можно проигрывать клип песни
37	«Графический редактор». На поле размером 500x500 строится по введенному описанию графический примитив. Начало координат в левом нижнем углу. Вид примитива определяется по охватывающим скобкам описания: круглые – ломаная линия, квадратные – прямоугольник, угловые – окружность, фигурные – трапеция, прямой слеш – произвольный многоугольник (полигон), обратный слеш – ромб, одинарные кавычки – дуга, двойные кавычки – сектор. В редактор введены средства задания толщины линии (1 ÷ 5 пикселей), цвета линий и заливки, очистки экрана
38	«Управление краном». Изображен кран и пульт управления стрелой и перемещением крана. Можно перемещать крюк вверх, вниз, влево, вправо, а кран слева направо и назад по экрану
39	«Тренировка внимательности». Вверху экрана исходно располагается 15 прямоугольников, размеры которых задаются случайным образом: ширина и высота в диапазоне 50 ÷ 100 пикселей. При запуске игры внизу экрана в центре появляется прямоугольник, соответствующий одному из прямоугольников верхнего ряда. Перемещая стрелками ←, ↑, → и ↓ его надо наложить на соответствующий прямоугольник верхнего ряда, после чего они исчезают, и внизу появляется новый прямоугольник. На уровне верхнего ряда прямоугольник можно перемещать только по вертикали. Фиксируется время игры. Три лучших результата запоминаются
40	«Автобусные маршруты». Карта Калининграда и список 10 автобусных маршрутов. При выборе одного из них он отображается на карте с пронумерованными остановками. В отдельном окне выводится список остановок
41	«Библиотека». Изображена книжная полка с корешками 20 книг. По щелчку мыши по корешку появляется в окошке рядом информация о книге: автор(ы), название, год издания, объем. Краткая аннотация книги. Щелчок вне полки убирает информацию
42	«Тренировка внимательности». Матрица 10x10. Ячейки матрицы заполнены в случайном порядке числами в диапазоне от 0 до 99. Размер цифр меняется случайным образом в диапазоне от 10 до 30 пунктов. Также случайно задается цвет цифр и параметры шрифта: толщина и наклон. Заполнение матрицы происходит при запуске игры. Одновременно запускается таймер. Необходимо последовательно мышью пройти все числа от 0 до 99. Три лучших результата фиксируются

1	2
43	«Тренировка внимательности». Матрица 10x10. Ячейки матрицы заполнены в случайном порядке изображениями зверей, рыб, растений, людей, строений и т. п. Заполнение матрицы происходит при запуске игры. Одновременно запускается таймер. В отдельном окошке появляются случайно изображения из матрицы. Необходимо в матрице найти аналогичное изображение и щелкнуть по нему мышью, после чего оно исключается из игры. Фиксируются три лучших результата
44	«Конкур». По всему экрану расставлены в произвольном порядке 15 ворот шириной 40 пикселей и указано направление их прохождения. Слева экрана находится линия старта, а справа – финиша. Исходно колобок диаметром 30 пикселей находится на линии старта. Стрелками ←, ↑, → и ↓ перемещая колобок надо пройти все ворота и достичь финиша. Фиксируется время прохождения и лучшие результаты
45	«Биатлон». Вверху экрана находится стандартная биатлонная мишень (5 окружностей) с диаметром круга 20 пикселей и таким же расстоянием между кругами, внизу ствол ружья, который можно перемещать влево/вправо. Расстояние между мишенью и ружьем – 500 пикселей. Исходно ружье находится вне проекции мишени. Необходимо как можно быстрее поразить все мишени. Если после пяти выстрелов не вся мишень поражена, то следующие выстрелы можно делать только с интервалом в 10 сек. При каждом запуске мишень случайно отклоняется от центрального положения в диапазоне ±20 пикселей. Можно фиксировать лучшие результаты
46	«Тренировка внимательности». При запуске программы отображается таблица 3x3. В шести случайно выбранных ячейках отображаются цифры от 0 до 9. Через заданное время таблица очищается и также в заданное время надо восстановить ее вид
47	«Кратчайший путь». Матрица 10x10. Во все ячейки, кроме левой верхней и правой нижней, заносятся случайные числа в диапазоне 0 ÷ 100. Программно определяется самый краткий по сумме путь из левой верхней ячейки в правую нижнюю. Двигаться можно только вправо и вниз. После этого путь выделяется цветом
48	«Самый длинный путь». Матрица 10x10. Во все ячейки, кроме левой верхней и правой нижней, заносятся случайные числа в диапазоне -100 ÷ 100. Программно определяется самый большой по сумме путь из левой верхней ячейки в правую нижнюю. Двигаться можно только влево, вправо и вниз. После этого путь выделяется цветом
49	«Поиск кратчайшего пути». Создается орграф в виде сети с одной начальной и одной конечной вершинами и не менее 10 промежуточных вершин. Как минимум у шести вершин не менее двух выходных дуг. Для каждой вершины задается вес и смежность с другими вершинами. Надо найти, просчитать и выделить кратчайший путь в графе

1	2
50	«Найди совпадения». Вверху экрана в случайном порядке располагаются семь контуров окружностей, диаметр которых задается тоже случайным образом в диапазоне $50 \div 150$ пикселей. Внизу экрана также случайным образом располагаются семь закрашенных разным цветом окружностей того же диаметра. Надо на время, перетаскивая мышью нижние окружности, совместить их с верхними. Совмещение приводит к самоуничтожению данных окружностей. Три лучших результата запоминаются
51	«Управление автобусом». Исходно слева на экране нарисован автобус. На экране также изображен пульт управления автобусом, включающий средства запуска и останова движения, открывания и закрывания дверей. Пуск возможен только при закрытых дверях, двери можно открывать только после остановки. При достижении правой границы экрана автобус снова появляется слева
52	«Визуальная иллюстрация сортировки массива методом «вставки». Количество элементов массива в диапазоне $8 \div 15$. Возможны два варианта выполнения: пошагово по команде пользователя и в автоматическом режиме
53	«Тренировка внимательности». Матрица 10×10 . Ячейки матрицы заполнены в случайном порядке числами в диапазоне от 0 до 99. Размер цифр меняется случайным образом в диапазоне от 10 до 30 пунктов. Также случайно задается цвет цифр и параметры шрифта: толщина и наклон. Заполнение матрицы происходит при запуске игры. Одновременно запускается таймер. Необходимо последовательно мышью пройти все числа от 99 до 0. Три лучших результата фиксируются
54	«Гонки». Шесть дорожек, по которым двигаются с разной скоростью шесть участников (стилизованных автомобилей, лошадей, собак и т. п.). Движение попиксельное. На каждом шаге у каждого участника задержка меняется случайным образом в пределах 30 %. Во время гонки все время отображается текущее положение каждого участника. Гонка заканчивается при достижении каждым участником правого края экрана. Есть средства старта гонки и возврата в исходное состояние
55	«Шлюз». Схематично изображен шлюз канала с двумя воротами, которыми можно управлять (закрывать и открывать). Одновременно ворота не могут быть в открытом состоянии. Движение судна возможно как слева, так и справа. Заполнение шлюза водой должно быть постепенным. Визуально отображается состояние ворот
56	Игра «2048»

ОРГАНИЗАЦИЯ ЗАЩИТЫ КУРСОВОЙ РАБОТЫ

После создания программного приложения, реализующего задачу курсовой работы, результат демонстрируется преподавателю, руководящему данным процессом, для чего заранее определяется дата и время встречи, или это происходит в заранее установленные часы консультации по курсовой работе. Если преподаватель выражает согласие с представленными результатами, студент переходит к написанию пояснительной записки с указанным выше содержанием. В противном случае он фиксирует все сделанные преподавателем замечания и согласовывает дату и время следующей встречи.

После написания студентом пояснительной записки она, 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 преподавателем устанавливается 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:

а. 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;

б. материал пояснительной записки излагается грамотно, логично, последовательно;

в. оформление отвечает требованиям написания курсовой работы;

г. 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:

а. в работе, выполненной на достаточном теоретическом и практическом уровне, полно и всесторонне решаются вопросы поставленной проблемы, но нет должной степени творчества;

б. студент показал знание теоретического материала по рассматриваемой проблеме, однако умение анализировать, аргументировать свою точку зрения, делать обобщения и выводы вызывают у него затруднения;

в. материал пояснительной записки не всегда излагается логично, последовательно. Имеются недочеты в оформлении курсовой работы;

г. во время защиты студент показал умение кратко, доступно (ясно) представить результаты исследования, однако затруднялся отвечать на поставленные вопросы.

3. Оценку «удовлетворительно» имеют работы:

а. 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;

б. студент испытывает затруднения при необходимости анализировать, аргументировать свою точку зрения, делать обобщения и выводы;

в. материал не всегда излагается логично, последовательно;

г. имеются недочеты в оформлении курсовой работы;

д. 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 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. Буйначев, С. К. Основы программирования на языке Python: учеб. пособие / С. К. Буйначев, Н. Ю. Боклаг. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 91 с.
4. Рогов, А. Ю. Технология программирования: учеб. пособие / А. Ю. Рогов, О. В. Проститенко. – Санкт-Петербург: СПбГТИ(ТУ), 2010. – 112 с.
5. Иванова, Г. С. Технология программирования: учеб.е пособие / Г. С. Иванова. – Москва: Изд-во КноРус, 2016. – 334 с.

ПРИЛОЖЕНИЕ 1

Образец заполнения титульного листа
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО РЫБОЛОВСТВУ

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

Институт цифровых технологий

Кафедра прикладной информатики

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 техн. наук доктор технических наук

канд. техн. наук кандидат технических 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	24
13.1 Требования к информационной и программной совместимости	37
13.2 Требования к составу и параметрам технических средств	39
14. Руководство пользователя	43
.....	
Список использованных источников.....	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 основных видов электронных усилителей, а также критериев, по которым выбирается конкретный тип усилителя. Приводится методика выбора типа усилителя. В следующей главе обосновывается выбор типа и детально проводится проектирование усилителя на основе исходных данных.

Ключевые слова: усилитель, транзистор, коэффициент передачи

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 разработки схем алгоритмов программ.

Рассматриваемый ГОСТ практически полностью соответствует международному стандарту **ISO 5807:1985**.

Элементы блок-схем алгоритмов

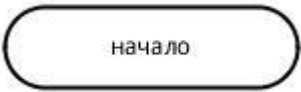
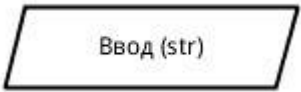
Блок-схема представляет собой совокупность символов, соответствующих этапам работы алгоритма и соединяющих их линий.

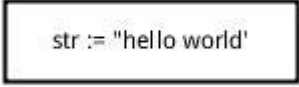
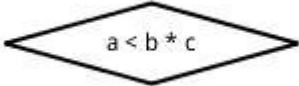
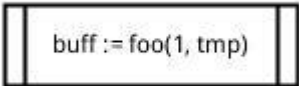
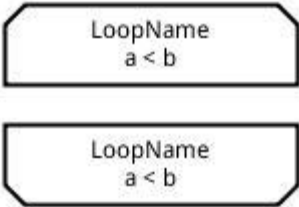
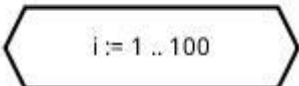

Пунктирная линия используется для соединения символа с комментарием. *Сплошная линия* отражает зависимости по управлению между символами и может снабжаться стрелкой. Стрелку можно не указывать при направлении дуги слева направо и сверху вниз.

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, дисплей 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>

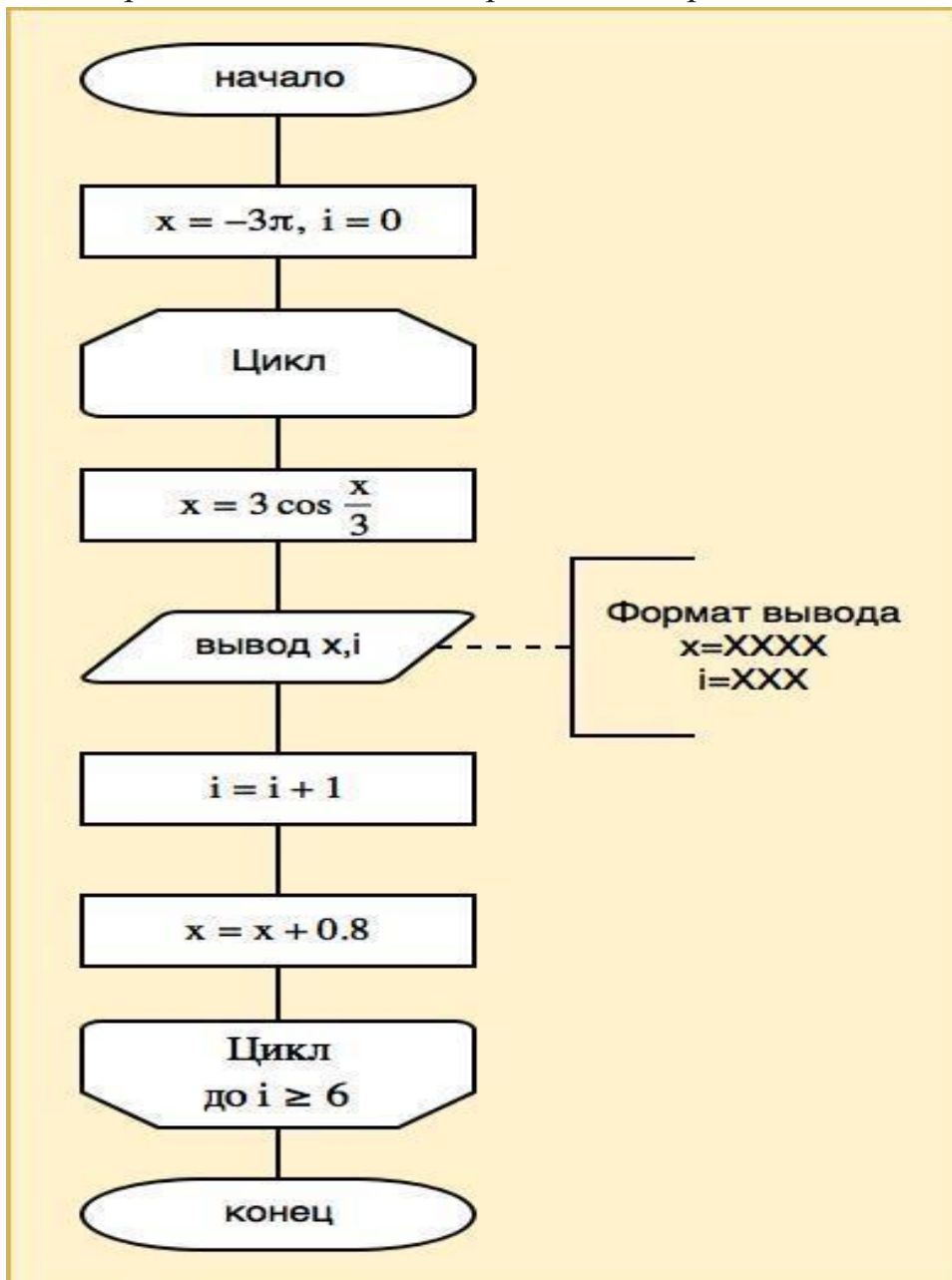
Too(i, str)
выделяет i-тый символ
строки str

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*.

Учебное издание

Леонид Григорьевич Высоцкий
Нина Борисовна Розен

ВЫСОКОУРОВНЕВЫЕ ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ (ВТП)

Редактор Э. С. Круглова

Подписано в печать 04.09.2023 г. Формат 60 × 90 1/16
Уч.-изд. л. 2,3. Печ. л. 2,1. Тираж 27 экз. Заказ № 44

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