

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ КемГУ

Дата и время: 2025-04-23 00:00:00

471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

Ю.С. Гаврилова

ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ

*Методические указания к выполнению курсовой работы по дисциплине «Языки и методы
программирования»*

для обучающихся по направлениям подготовки

01.03.02 Прикладная математика и информатика,

02.03.03 Математическое обеспечение и администрирование информационных систем

Новокузнецк

2020

УДК [378.147.88:004.43](072)

ББК 74.484(2Рос-4Кем)я73+32.973-018.1я73

Г 12

Гаврилова Ю.С.

Г 12 Языки и методы программирования: методические указания к выполнению курсовой работы по дисциплине для студентов факультета информатики, математики и экономики, обучающихся по направлению подготовки 01.03.02 Прикладная математика и информатика, 02.03.03 Математическое обеспечение и администрирование информационных систем / Ю.С. Гаврилова; Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк : НФИ КемГУ, 2020 – 35 с.

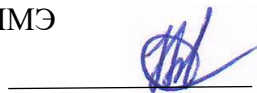
В работе изложены методические указания к выполнению курсовой работы по дисциплине «Языки и методы программирования»: цель, задачи и содержание курсовой работы, структура курсовой работы, требования к ее подготовке и оформлению, процедура защиты, критерии оценивания курсовой работы в балльно-рейтинговой системе, список основной и дополнительной литературы, современных профессиональных баз данных и информационных справочных систем, приложения (титульный лист, задание на курсовую работу, образец введения курсовой работы, примерные темы курсовых работ).

Рекомендовано на заседании
кафедры математики, физики и
математического моделирования
Протокол № 8 от 16.03.2020
Заведующий каф. МФММ



/ Е.В. Решетникова

Утверждено методической комиссией
факультета информатики, математики и
экономики
Протокол № 8 от 18.05.2020
Председатель методической комиссии
ФИМЭ



/ Г.Н. Бойченко

УДК [378.147.88:004.43](072)

ББК 74.484(2Рос-4Кем)я73+32.973-018.1я73

Г 12

© Гаврилова Юлия Сергеевна

© Федеральное государственное бюджетное образовательное учреждение высшего образования «Кемеровский государственный университет», Новокузнецкий институт (филиал), 2020

Текст представлен в авторской редакции

Оглавление

ПРЕДИСЛОВИЕ	4
ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ.....	5
СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ.....	7
ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ.....	20
ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	23
ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК	24
ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	27
КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ	28
РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	31
СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ	33
ПРИЛОЖЕНИЕ 1. Титульный лист	35
ПРИЛОЖЕНИЕ 2. Задание на курсовую работу	36
ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы	37
ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Языки и методы программирования»	39

ПРЕДИСЛОВИЕ

Настоящие методические материалы адресованы студентам, получающим квалификацию бакалавр по направлениям подготовки: 01.03.02 Прикладная математика и информатика, 02.03.03 Математическое обеспечение и администрирование информационных систем и направлены на оказание помощи студентам в написании курсовой работы по дисциплине «Языки и методы программирования».

Курсовая работа по дисциплине «Языки и методы программирования» является важным компонентом профессиональной подготовки будущего специалиста в сфере информационных технологий. Согласно учебным планам основных образовательных программ направлений подготовки 01.03.02 «Прикладная математика и информатика», 02.03.03 «Математическое обеспечение и администрирование информационных систем» курсовая работа выполняется на втором курсе в четвертом семестре.

Целью проектирования курсовой работы является формирование способности студента к самостоятельному анализу и оценке направления программирования, языков программирования и сред разработки с точки зрения их использования для создания программных продуктов, а также формирование навыков применения современных информационных технологий в процессе создания программных продуктов.

В процессе написания курсовой работы осуществляется закрепление, углубление и обобщение знаний, полученных студентами за время обучения, и применение этих знаний к решению конкретных задач.

ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа представляет собой вид учебной работы обучающегося, в которой присутствуют элементы самостоятельного научного исследования.

Курсовая работа – это самостоятельный комплексный проект, при выполнении которого студент решает учебно-исследовательские задачи, носящие творческий характер. Поэтому материал для нее не копируется из интернета, а анализируется, систематизируется, приобретая в результате этого характер оригинальности.

При выполнении курсовой работы обучающийся должен научиться составлять программы высокого качества, легко модифицируемые и простые в обращении с применением современных информационных технологий.

Курсовая работа носит прикладной характер, направлена на практическое применение изученного материала к решению задачи. В рамках дисциплины «Языки и методы программирования» обучающиеся изучили современные информационные технологии: технологии программирования, языки программирования и среды разработки. Некоторые из них можно применить для решения задачи, поставленной в курсовой работе. Однако, по желанию обучающийся может использовать в работе информационные технологии, изученные самостоятельно, при условии, что их применение оправдано (более эффективное решение задачи).

Во время выполнения курсовой работы обучающийся должен продемонстрировать умение составлять алгоритм решения прикладной задачи, а затем реализовывать составленный алгоритм в какой-либо среде разработки.

Алгоритм решения прикладной задачи строится на базе концептуальной модели предметной области. Концептуальная модель задачи необходима для того, чтобы конкретизировать требования к алгоритму, поскольку тема курсовой работы представлена в общем виде. Пример: тема «Разработка приложения «Библиотека» может предполагать разработку

приложения для районной библиотеки, где читатели берут бумажные книги, а может означать разработку некоторой оболочки для хранилища электронных книг. Концептуальная модель как раз и содержит описание выбранной обучающимся предметной области. Построение концептуальной модели предметной области осуществляется студентом самостоятельно, однако обязательно проверяется и корректируется руководителем курсовой работы, поскольку ошибка на данном шаге может привести к ошибкам во всей практической части работы.

Алгоритм решения прикладной задачи может быть основан на применении какого-либо известного паттерна проектирования. Обучающийся должен самостоятельно выбрать паттерн проектирования, соответствующий задаче. Если обучающийся не смог самостоятельно выполнить задание этого этапа, он может воспользоваться помощью преподавателя (с последующим выставлением соответствующего количества баллов при оценивании выполнения работы).

Решение прикладной задачи (программная реализация алгоритма) осуществляется обучающимся самостоятельно.

Оформление пояснительной записки к курсовой работе рекомендуется осуществлять параллельно с процессом решения задачи.

СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

Структура пояснительной записки (текста) курсовой работы должна способствовать раскрытию избранной темы. Структурные элементы: титульный лист, оглавление, введение, основная часть, заключение, библиографический список (использованные источники) и, при необходимости, приложения.

Введение курсовой работы в обязательном порядке должно содержать постановку цели и задач исследования, обоснование его актуальности, теоретической и практической значимости. Пример составления и оформления введения для курсовой работы на тему «Разработка программного приложения для расчета объема фигуры вращения, заданной произвольной кривой» представлен в Приложении 3.

Актуальность исследования – это степень важности курсовой работы в данный момент времени в конкретной стадии развития выбранной отрасли знаний. Она часто связана с неизученностью (недостаточной изученностью) темы или возможностью решения определенной задачи практики и охватывает запросы отдельной отрасли науки, региона, части общества.

В качестве **цели** исследования будет обозначен результат, который будет получен в ходе написания курсовой работы. Целью написания курсовой работы является решение конкретной проблемы, обозначенной в теме работы. Поэтому цель следует формулировать в виде достижения результата.

Все темы курсовой работы по дисциплине «Языки и методы программирования» представляют собой разработку программного приложения, поэтому начать формулировку цели необходимо со слов: «разработать программное приложение» или «разработать программный продукт». Цель исследования формулируется на основе темы курсовой работы, например, если тема курсовой работы «Разработка программного

приложения для расчета объема фигуры вращения, заданной произвольной кривой», то цель данной работы можно сформулировать следующим образом: «разработать программное приложение, моделирующее фигуру вращения произвольной формы, с последующим расчетом их объема и площади фигуры под графиком».

Следует обратить внимание, что в формулировке цели курсовой работы не нужно указывать конкретный язык программирования или среду разработки, используемые в процессе выполнения работы, поскольку оценка возможности применения различных средств разработки должна быть получена во время проведения исследования, а не определена заранее (см. задание на курсовую работу). Также, при формулировании цели курсовой работы следует избегать дословного совпадения с темой работы.

Постановка **задач** исследования по выбранной теме курсовой работы осуществляется согласно поставленной цели. Задачи представляют собой ряд (как правило, четыре или пять) конкретных последовательных этапов достижения поставленной цели.

Курсовая работа должна содержать теоретическую и практическую часть, это должно отражаться и в задачах написания работы. Задачи курсовой работы взаимосвязаны друг с другом, поэтому они располагаются последовательно раскрытию теоретического и практического материала и соответствуют составленному содержанию работы.

Теоретическая и практическая значимость исследования формулируются на основании возможности последующего применения результатов, полученных в курсовой работе.

Основная часть пояснительной записки курсовой работы должна иметь двухчастную структуру, т.е. курсовая работа состоит из двух глав, а каждая глава, в свою очередь, из параграфов.

Рекомендуется придерживаться следующего порядка изложения текста курсовой работы:

1. Средства разработки.

- 1.1 Технологии программирования.
- 1.2 Языки программирования.
- 1.3 Среды разработки.
- 2. Разработка программного приложения.
 - 2.1 Разработка алгоритма.
 - 2.2 Программная реализация алгоритма.
 - 2.3 Результат разработки.

В качестве примера приведем структуру курсовой работы по теме «Разработка программного приложения для расчета объема фигуры вращения, заданной произвольной кривой»:

СОДЕРЖАНИЕ	
Введение.....	3
1 Анализ средств разработки.....	4
1.1. Исследование технологий программирования.....	4
1.2. Обзор языков программирования.....	7
1.3. Выбор среды разработки.....	10
2 Разработка программного приложения.....	13
2.1. Математическая модель расчета площади фигуры под кривой и объема фигуры вращения.....	14
2.2. Разработка алгоритма приложения	18
2.3. Проект программного приложения.....	22
2.4. Программная реализация алгоритма.....	25
2.5. Результат разработки.....	28
Заключение.....	30
Список литературы.....	31

Рассмотрим подробнее вопросы, которые необходимо исследовать в основной части курсовой работы.

Первая глава – это анализ средств разработки, где излагается исследование существующих технологий программирования, языков программирования и сред разработки, оценка возможности применения данных информационных технологий для решения поставленной задачи. Все вышеперечисленное можно отнести к средствам разработки, поэтому следует назвать первую главу «Анализ средств разработки».

В каждом из параграфов рекомендуется рассмотреть соответствующие средства разработки с точки зрения возможности применения их для решения поставленной задачи. Необходимо обратить внимание, что сначала обучающийся выбирает оптимальную для решения задачи технологию программирования, затем язык программирования из тех, которые можно применить для реализации выбранной технологии. Далее происходит выбор среды разработки, позволяющей реализовать приложение на выбранном языке программирования. Под выбором средств разработки подразумевается исследование каждого средства, а также его достоинств и недостатков в контексте решаемой задачи. Не следует рассматривать те средства разработки, использование которых очевидно не является оптимальным в рамках исследуемой темы (рекомендуется выбирать средства разработки из изученных в курсе «Языки и методы программирования»).

Поскольку каждый параграф первой главы представляет собой исследование средств разработки, в конце параграфа должно быть краткое резюме, в котором содержатся сведения о конкретном средстве разработки, которое будет применяться при решении задачи, с указанием его преимуществ для решения поставленной задачи. Пример резюме параграфа 1.2: *В данной работе языком программирования для разработки приложения является язык C++, поскольку помимо возможности реализации объектно-ориентированной парадигмы, он предоставляет наиболее полные и удобные*

средства для работы с библиотеками GL/gl.h и GL/glu.h. Для параграфов 1.1 и 1.3 резюме формулируется аналогичным образом.

Вторая глава имеет практическую направленность, а именно: составление алгоритма решения задачи и его программную реализацию. Название для второй главы можно сформулировать следующим образом: «Разработка программного приложения».

Таким образом, в данной главе обучающемуся необходимо представить концептуальную модель задачи и алгоритм решения, программную реализацию алгоритма и результат разработки (снимки экрана). Алгоритм решения задачи студент составляет самостоятельно, но под руководством преподавателя, а программная реализация составленного алгоритма представляет собой самостоятельную работу студента.

Концептуальная модель – это краткое описание предметной области в виде небольшого реферативного описания: термины и понятия предметной области, субъекты и объекты, способы взаимодействия субъектов, способы использования объектов, закономерности.

Пример: если предметной областью исследования являются процессы регистрации транспортного средства в госавтоинспекции, то необходимо описать, какая информация будет храниться в приложении (регистрационный номер, цвет автомобиля, год выпуска, адрес проживания владельца, наложенные штрафы) и в каком виде; запросы, которые должны будут выполняться во время работы приложения (какая информация будет выводиться в результате работы приложения).

Алгоритм решения прикладной задачи может быть представлен либо в виде блок-схемы, либо в виде нумерованного списка, описывающего последовательность действий. Блок-схема алгоритма должна быть оформлена в соответствии с ГОСТ 19.701-90 (ИСО 5807-85)¹.

Если в процессе решения задачи используются формулы, их необходимо также привести в тексте курсовой работы.

¹ см. раздел «Литература для оформления курсовой работы»

Пример: 1. Пользователь задает область по оси Ox , и по оси Oy , в которой находится график. Размер области построения изображения 660:480 пикселей. Исходя из этого, делается пересчет 1 пикселя для отрисовки по формулам 1-2.

$$L_x = \frac{b_x}{660}, \quad (1)$$

$$L_y = \frac{b_y}{480}. \quad (2)$$

Проект программного приложения также следует представить в тексте курсовой работы. Обычно проект приложения представляют в виде набора схем (диаграмм), отражающих структуру и поведение программного приложения и его частей. Необходимо представить структурную схему разрабатываемого программного приложения, позволяющую определить взаимоотношения между отдельными частями программы и содержание программных сообщений.

Самый простой вид программного продукта – программа в качестве структурных компонентов может включать только подпрограммы и библиотеки ресурсов. Структурными компонентами программного продукта могут служить программы, подсистемы, библиотеки ресурсов и т. п.

Пример: «В соответствии с проведенным выше анализом задачи разрабатываемая программа должна содержать: последовательное задание исходных данных для эксперимента, которые определяют условия для решения задачи; выбор вида решения и вида результатов (в соответствии с ранее определенными функциями). В результате может быть сформирована структурная схема функционирования интерфейса, которая представлена на рисунке 1»



Рисунок 1. Структурная схема функционирования интерфейса

На основе структурной схемы в соответствии с нотацией UML можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Более полное представление о проектируемом программном продукте с точки зрения взаимодействия его компонентов между собой и с внешней средой дает функциональная схема. Функциональная схема или схема данных – схема взаимодействия компонентов ПО с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств. Для изображения функциональных схем необходимо применять стандарт ГОСТ 19.701–90.

Если разработка программного приложения осуществлялась на основе объектно-ориентированной парадигмы программирования, необходимо в соответствии с нотацией UML представить следующие диаграммы: диаграммы взаимодействия, иерархии классов и т.д.

Диаграмма взаимодействия – это диаграмма, на которой представлено взаимодействие, состоящее из множества объектов и отношений между ними, включая и сообщения, которыми они обмениваются. Этот термин применяется к видам диаграмм с акцентом на взаимодействии объектов. Пример диаграммы взаимодействия для предметной области «Библиотека» представлен на рисунке 2.

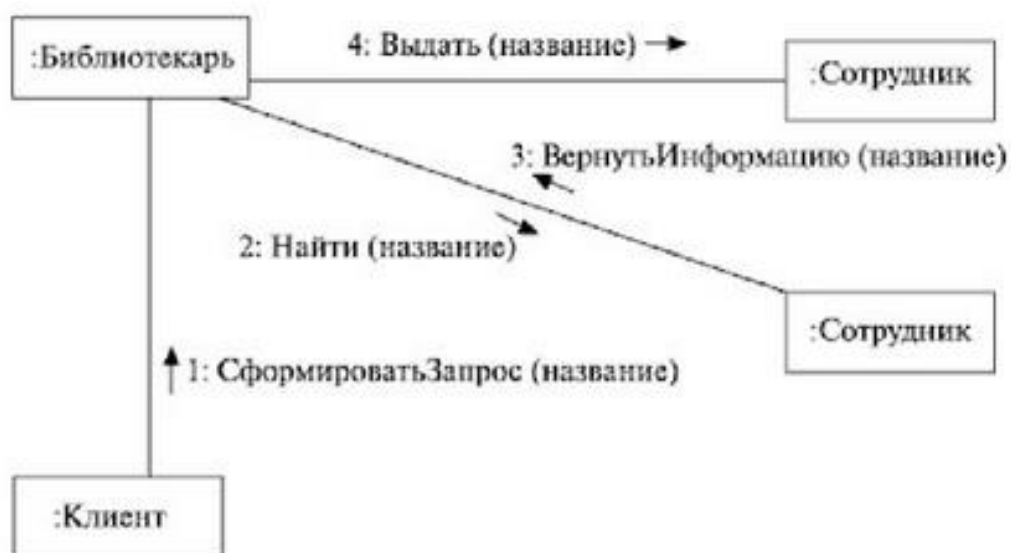


Рисунок 2. UML-диаграмма взаимодействия

Диаграмма иерархии классов – это набор статических, декларативных элементов модели. Информация с диаграммы классов напрямую отображается в исходный код приложения – в большинстве существующих инструментов UML-моделирования возможна кодогенерация для определенного языка программирования. Таким образом, диаграмма классов – конечный результат проектирования и отправная точка процесса разработки. Пример диаграммы взаимодействия для предметной области «Доставка товара» представлен на рисунке 3.

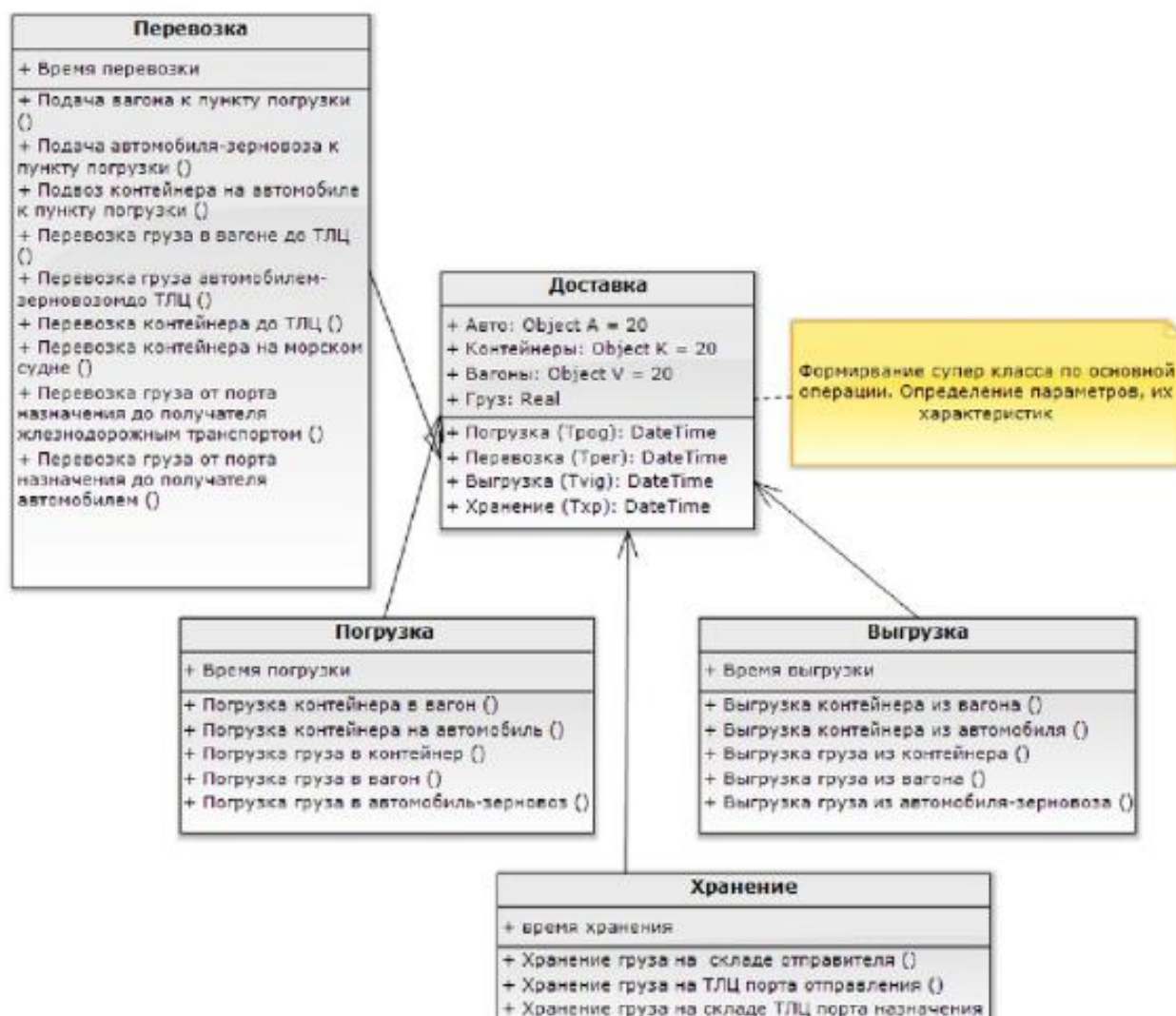


Рисунок 3. UML-диаграмма иерархии классов приложения

Обучающийся может приводить фрагменты кода, если они представляют какую-либо научную или практическую значимость, или если ему необходимо сделать акцент на этой части разработки.

При оформлении листингов следует использовать шрифт Courier New, размер – 12 пт, межстрочный интервал – одинарный. Рекомендуется отделять смысловые блоки пустыми строками, а также визуально обозначать вложенные конструкции с помощью отступов.

Ключевые слова и комментарии рекомендуется выделять с помощью различных начертаний шрифта. Таким же образом в основном тексте работы должны обозначаться имена библиотек, подпрограмм, констант, переменных, структур данных, классов, их поля и методы.

Листинги должны иметь сквозную порядковую нумерацию, например, «Листинг 1». При ссылке на листинг следует писать слово «листинг» с указанием его номера. Название листинга печатается тем же шрифтом, что и основной текст, и размещается над листингом слева, без абзацного отступа через «тире» после номера листинга.

Пример: *Первым шагом является «отрисовка» фигуры. Для этого необходимо создать функцию, которая будет строить непосредственно фигуру вращения в 3D.*

Идея рисования в 3D заключается в том, что, зная координаты точки x и y , мы можем нарисовать ее в 3D плоскости, изменяя лишь координату z . Пример программной реализации алгоритма построения фигуры вращения приведен в листинге 1.

Листинг 1 - программная реализация алгоритма построения фигуры вращения

```
System::Void Render(System::Void){
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();
    glTranslated(0,-5.0f,-30.0f); //отдаление фигуры x, y, z
    glRotatef(rtri,0.0,4.0f,2.0f); // вращение фигуры
    int a=0;
    steep_1 = 10.0/(double)count;
    for (double i = 0; i < 10; i += steep_1)
    {
        glBegin(GL_LINES ); //рисует линии (соединяем каждую точку)
        for (int k = 0; k <= 360; k++)
        {
            glColor3f(1,1,0); //цвет желтый
            glVertex3f(sin(k*3.14/180)*Massive_Y[a]*10/640,i*cos(k*
3.14/180)*Massive_Y[a]*10/640);

        }
        a++;
        glEnd(); // конец рисования
    }
}
```

Однако, необходимо учесть, что стоит рационально подойти к размещению в тексте пояснительной записки фрагментов кода, копировать все строки программы в работу не требуется.

Результат разработки оформляется в виде снимков экрана, сделанных во время функционирования приложения. Выбор конкретных модулей,

которые следует показать, осуществляется обучающимся самостоятельно, однако, стоит соблюсти 2 условия:

- наиболее полно продемонстрировать функциональные возможности приложения;
- не допустить избыточности информации.

При необходимости обучающийся может проконсультироваться с преподавателем по поводу представления графического материала (какие модули и окна программного приложения следует поместить в текст курсовой работы).

Чаще всего необходимо представить следующие результаты:

- главное окно приложения;
- окно ввода данных;
- результат, полученный после запуска программы.

Пример таких изображений представлен на рисунках 3-4.

В случае, если программное приложение предназначено для обработки информации и предполагает вывод данных не только на форму приложения, но и в файл (или любое другое хранилище данных), то результат также необходимо представить в тексте курсовой работы (в виде снимка экрана с открытым файлом).

Если в приложении осуществляется обработка неверного ввода данных, следует представить результат этой обработки (указать значения, которые были введены, и представить снимок экрана после запуска приложения).

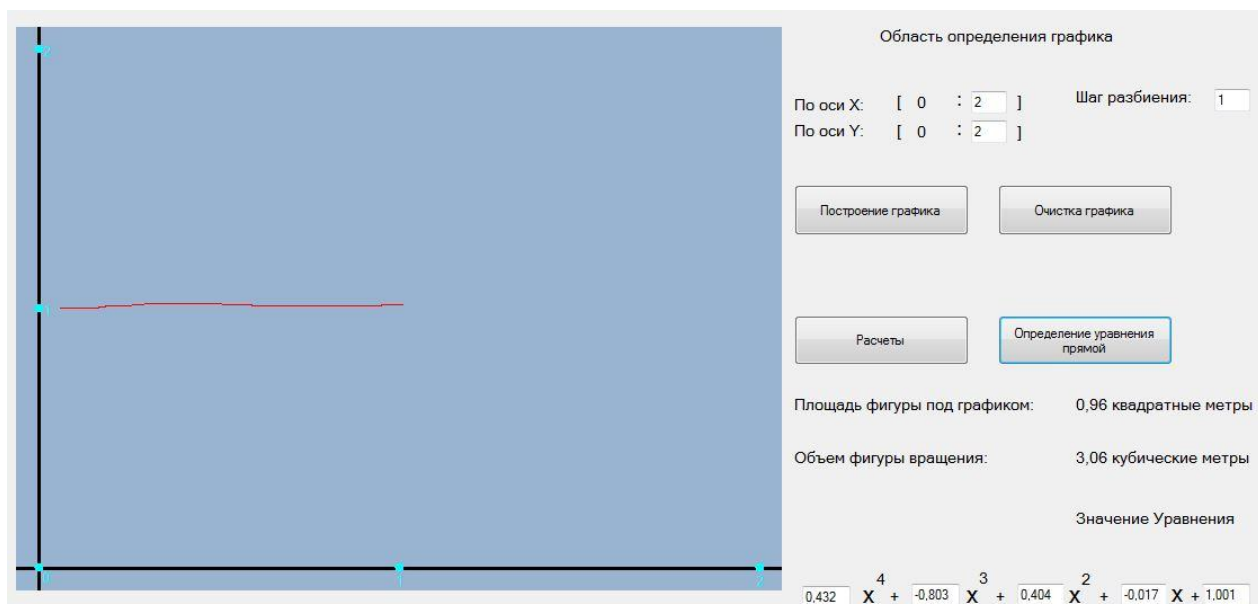


Рисунок 3. Главное окно приложения

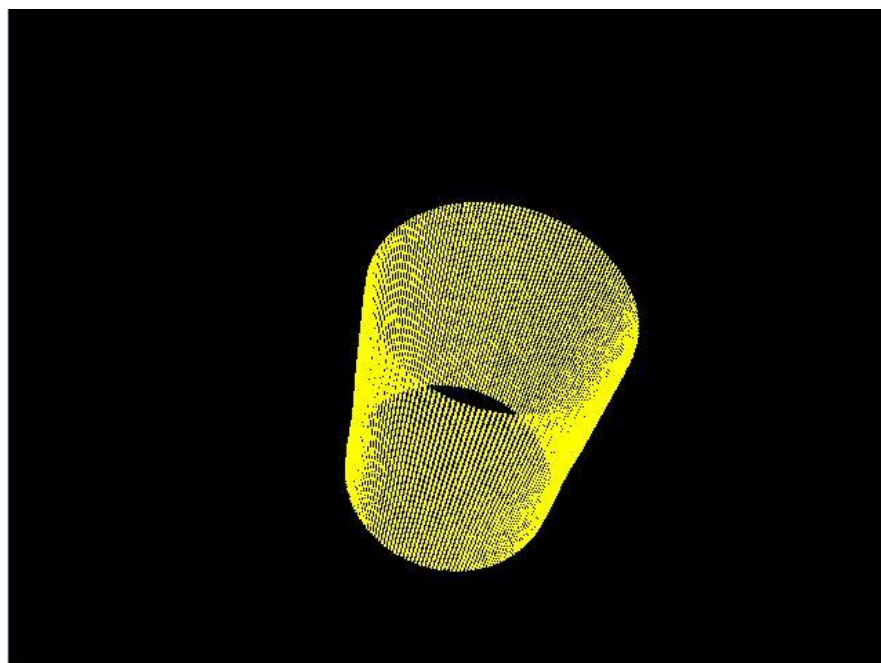


Рисунок 4. Результат запуска приложения

В заключении обобщаются наблюдения, сделанные во время работы и формулируются основные выводы, отражающие наиболее значимые результаты работы, предлагаются рекомендации относительно возможностей использования материалов и результатов работы. Обычно обучающийся формулирует выводы в порядке, соответствующем порядку задач курсовой работы (из введения). Кроме того, обучающийся может указать направление дальнейших исследований в рамках развития данной задачи.

При составлении текста пояснительной записки следует помнить, что:

- необходимо избегать логических ошибок, например, нельзя давать одинаковое название курсовой работе и одному из ее параграфов;
- содержание работы необходимо иллюстрировать наглядными материалами: таблицами, рисунками, схемами, графиками и т.п.

ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ

1. Для направлений подготовки 01.03.02 Прикладная математика и информатика (профиль «Математическое моделирование и информационные технологии»), 02.03.03 Математическое обеспечение и администрирование информационных систем (профиль «Программное и математическое обеспечение информационных технологий») в начале четвертого семестра кафедра предлагает студентам второго курса темы курсовых работ по дисциплине «Языки и методы программирования». Студент может предложить и свою тему, но ее необходимо обосновать, согласовать с руководителем и зарегистрировать на кафедре. Общее руководство и контроль за ходом выполнения курсовой работы осуществляет руководитель курсовой работы.

2. После выбора темы студенту следует встретиться с руководителем курсовой работы для обсуждения задачи. Руководитель выясняет степень подготовленности студента к работе над темой, рекомендует необходимую литературу и дает консультации о порядке выполнения работы. Для самоорганизации студента и своевременного контроля за ходом работы можно составить график ее выполнения.

*График выполнения курсовой работы по дисциплине «Языки и методы
программирования»*

студента (ки) группы _____

шифр группы _____

ФИО _____

№ п/п	Этапы работы	Планируема я дата выполнения	Фактическая дата	Отметка научного руководи теля
1	2	3	4	5
1.	Выбор темы, ее регистрация на кафедре, обсуждение целей и задач работы, графика ее выполнения с руководителем курсовой работы. Подбор библиографических			

	источников по теме, согласование их с руководителем.			
2.	Подготовка чернового варианта теоретической части работы и представление его руководителю работы. Работа над замечаниями руководителя курсовой работы, исправление недочетов и повторное представление.			
3.	Выполнение практической части работы: разработка алгоритма решения задачи. Представление руководителю разработанного алгоритма. Исправление недочетов и повторное представление руководителю.			
4.	Выполнение практической части работы: программная реализация разработанного алгоритма. Представление руководителю завершенной разработки. Исправление недочетов и повторное представление руководителю.			
5.	Подготовка чернового варианта практической части работы и представление его руководителю. Работа над замечаниями руководителя. Написание введения и заключения, оформление библиографического списка, титульного листа работы			
6.	Представление руководителю чернового варианта всей работы. Проверка на наличие в тексте работы некорректных заимствований. Исправление недочетов. Подготовка чистового варианта работы и представление его руководителю.			
7.	Защита курсовой работы			

Руководитель курсовой работы: _____

ФИО

Дата « ____ » _____ 2020__

3. После беседы с руководителем студент приступает к подбору литературы, ее анализу и составлению плана работы. Кроме рекомендуемой руководителем литературы, желательно начать поиск дополнительной

литературы по данной теме. Изучая литературу, полезно делать необходимые выписки (конспектировать), закладки, заметки.

4. Желательно, чтобы в процессе написания курсовой работы студент периодически встречался с руководителем, показывал ему законченные части работы, обсуждал свои идеи и наработки. Это поможет детализировать процесс написания курсовой работы, предостережет от ложных направлений, даст дополнительный импульс к творческому поиску. Руководитель свои рекомендации по выполнению курсовой работы дает лишь после того, как он убедился, что студент достаточно изучил проблему и понял ее сущность.

5. Опыт показывает, что даже при самом добросовестном отношении студента, первый вариант курсовой работы потребует существенных исправлений и доработки. Перечитав и выправив черновик, студент передает его руководителю курсовой работы, который принимает решение либо о допуске работы к защите, либо о доработке и устранении имеющихся недочетов и ошибок.

6. Чтобы студент мог успешно защитить курсовую работу, он должен хорошо владеть ее содержанием, уметь доказывать и обосновывать полученные результаты.

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

1. Объем курсовой работы зависит от темы. Студенту рекомендуется быть максимально кратким, обходиться без повторов, опускать совершенно очевидные выкладки, расчеты и пояснения. Опыт показывает, что объем курсовой работы колеблется в пределах 20-35 страниц машинописного текста.

2. Структурные элементы курсовой работы описаны ранее.

3. Текст курсовой работы оформляется в соответствии с Правилами оформления учебных работ².

4. Текст курсовой работы печатается на одной стороне листа бумаги стандартного размера А4 (210×297 мм). Каждый лист вкладывается в отдельный файл (мультифору), затем все файлы помещаются в скоросшиватель.

5. Курсовая работа должна содержать самостоятельные выводы и авторский текст не менее 70%. Проверка на наличие некорректных зависимостей осуществляется руководителем курсовой работы с помощью системы antiplagiat.ru во время предоставления обучающимся на проверку черновой работы. Обучающемуся рекомендуется самостоятельно заранее проверять свою работу по мере написания текста.

² Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный

ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК

Особое место среди методов исследования занимают методы изучения литературы по проблеме данной работы. Изучение литературы служит средством изучения истории и причин возникновения проблемы, ее современного состояния. Работа с литературой включает несколько методов.

Укажем наиболее важные из них.

1. Составление библиографии, то есть списка литературы, использованной автором при написании курсовой работы.
2. Реферирование - сжатое изложение основного содержания одной или нескольких работ по общей теме.
3. Конспектирование - детальное изложение главных положений и концептуальных идей работы.
4. Аннотирование - краткое, предельно сжатое изложение основного содержания литературных источников.
5. Цитирование - дословная запись высказываний, выражений автора, а также приведение в тексте курсовой работы фактических и статистических данных, содержащихся в литературных источниках.

Еще одно правило работы с литературой - использование библиографического списка в тексте работы: источник, внесенный в список, хотя бы один раз должен быть назван в тексте. И, наоборот, любой источник, на который автор ссылается в тексте работы, должен быть внесен в библиографический список.

При написании курсовой работы наиболее приемлемым является алфавитный способ группировки материалов, включенных в список использованных источников. Использованные источники при этом располагаются в общем алфавитном порядке фамилий авторов или заглавий книг, статей (если автор не указан).

Обращаясь к цитированию, необходимо соблюдать следующие правила:

нельзя вырывать фразы из текста; искажать его произвольными сокращениями; цитату необходимо брать в кавычки и т.п. Следует также обратить внимание на точное указание источников цитат.

Ссылки на использованную литературу в тексте указываются в квадратных скобках порядковым номером по списку источников, например, [23]. Если автор делает ссылку сразу на несколько источников, то в скобках через запятую указываются их номера в списке, например, [23, 25, 38]. Если автор приводит цитату из первоисточника, то в скобках после номера источника обязательно указывается страница, например, [34. С. 216].

Список литературы оформляется в соответствии с требованиями **ГОСТ Р 7.0.100-2018**.

*Примеры библиографического описания информационных источников
по ГОСТ Р 7.0.100-2018*

электронные издания:

1. Трофимов, В.В. Алгоритмизация и программирование : учебник / В.В. Трофимов, Т.А. Павловская. – Москва : Издательство Юрайт, 2019. – 137 с. – ISBN 978-5-534-07834-3. – URL: <https://www.biblio-online.ru/viewer/algoritmizaciya-i-programmirovanie-423824>. - (дата обращения: 22.03.2020). – Текст : электронный.

2. Калентьев А.А. Новые технологии в программировании : учебное пособие / А.А. Калентьев, Д.В. Гарайс, А.Е. Горяинов – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=480503. - (дата обращения: 22.03.2020). – Текст : электронный.

сайты в сети «Интернет»:

CITForum.ru : on-line библиотека свободно доступных материалов по информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.03.2020). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва,

2000 . – URL: <http://www.elibrary.ru> (дата обращения: 22.03.2020). – Режим доступа: для зарегистрир. пользвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 . – URL: <http://window.edu.ru/> (дата обращения: 22.03.2020). –Текст: электронный.

Курсовая работа с неправильно или небрежно оформленным библиографическим списком к защите не допускается.

ЗАЩИТА КУРСОВОЙ РАБОТЫ

1. Курсовая работа должна быть защищена не позднее начала экзаменационной сессии.

2. Руководитель курсовой работы, ознакомившись с окончательным вариантом курсовой работы, принимает решение о допуске к защите курсовой работы.

3. Во время защиты обучающийся кратко (3 - 5 минут) докладывает основные результаты своей работы, сопровождая результат наглядным материалом (презентация), а затем отвечает на вопросы преподавателя.

В докладе обучающийся озвучивает цель и задачи исследования; используемые при разработке алгоритмические и программные решения; основные результаты разработки; выводы по работе.

4. Преподаватель оценивает, насколько успешно достигнута цель и решены задачи, поставленные в курсовой работе; качество доклада и ответов на вопросы; степень самостоятельности студента при выполнении исследования (анализе технологий программирования, языков программирования и средств разработки); полнота проведенной работы (разработка алгоритма решения задачи, программная реализация алгоритма). Отметка за курсовую работу выставляется согласно приведенной в методических указаниях балльно-рейтинговой системе и таблице перевода баллов в буквенный эквивалент.

5. После этого руководитель курсовой работы заносит отметку в зачетную книжку студента и в соответствующую ведомость.

Курсовая работа хранится на кафедре вплоть до завершения студентом учебы в университете.

КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ

Баллы по курсовой работе выставляются обучающемуся за два вида деятельности:

- выполнение исследования и оформление пояснительной записки (80 баллов) (таблица 1);
- защита курсовой работы (20 баллов) (таблица 2).

Таблица 1 – Распределение баллов по этапам выполнения исследования

Этапы выполнения исследования	Оценка в аттестации	Баллы
Этап 1. Анализ средств разработки. 1.1 Анализ технологий программирования. 1.2 Анализ языков программирования. 1.3 Анализ средств разработки.	Баллы за этап 1.1: 8 баллов – проведен анализ технологий программирования, сделаны выводы о применении каждой технологии для решения поставленной задачи, однако для решения задачи выбрана не самая оптимальная; 10 баллов – проведен анализ технологий программирования, сделаны корректные выводы о применении каждой технологии для решения поставленной задачи, но есть некоторые недочеты; 14 баллов – проведен в полном объеме анализ технологий программирования, сделаны корректные выводы о применении каждой технологии для решения поставленной задачи. Баллы за этап 1.2: 1 балл – проведен анализ языков программирования, подходящих для решения поставленной задачи, однако есть существенные недостатки; 2 балла – проведен анализ языков программирования, однако имеются некоторые неточности, или недочеты; 3 балла – в полном объеме проведен анализ языков программирования, сделаны корректные выводы о применении каждого языка программирования для решения поставленной задачи. Баллы за этап 1.3:	10-20

	1 балл – проведен анализ сред разработки, однако рассмотрены не все подходящие среды разработки; 2 балла – проведен анализ сред разработки, однако имеются некоторые неточности, или недочеты; 3 балла – в полном объеме проведен анализ сред разработки, сделаны корректные выводы о применении каждой среды для решения поставленной задачи.	
Этап 2. Разработка приложения 2.1 Разработка алгоритма 2.2 Программная реализация алгоритма 2.3 Результат разработки	Баллы за этап 2.1: 14 баллов – алгоритм составлен с помощью преподавателя; 18 баллов – алгоритм составлен, но наблюдаются некоторые неточности или недочеты; 20 баллов – алгоритм составлен самостоятельно и в полном объеме. Баллы за этап 2.2: 20 баллов – алгоритм реализован не в полном объеме; 25 баллов – алгоритм реализован, однако наблюдаются некоторые неточности или недочеты; 30 баллов – алгоритм реализован в полном объеме. Баллы за этап 2.3: 7 баллов – результат разработки представлен не в полном объеме (условия полноты представленного результата разработки перечислены на стр. 11 данных методических указаний); 9 баллов – результат представлен, однако наблюдаются некоторые неточности или недочеты; 10 баллов – результат представлен в полном объеме.	41-60
Итого за выполнение курсовой работы		51-80

Таблица 2 – Распределение баллов за защиту курсовой работы

Виды и результаты учебной работы	Оценка в аттестации	Баллы
Устное выступление об основных результатах, полученных во время выполнения курсовой работы (5-7 минут)	6 баллов – пороговое значение; 10 баллов – максимальное значение.	6 - 10
Сопровождение устного выступления наглядным	2 балла – пороговое значение; 5 баллов – максимальное значение.	2 - 5

материалом (презентация)		
Ответы на вопросы по теме курсовой работы	2 балла – пороговое значение; 5 баллов – максимальное значение.	2 - 5
Итого по промежуточной аттестации (защита курсовой работы)		10-20

Набранные обучающимся баллы переводятся в оценку и буквенный эквивалент согласно Положению о балльно-рейтинговой системе оценки деятельности студентов КемГУ от 30.12.2016г. (таблица 3).

Таблица 3 - Перевод баллов из 100-балльной шкалы в числовой и буквенный эквивалент

Сумма баллов для дисциплины	Оценка	Буквенный эквивалент
86 - 100	5	отлично
66 - 85	4	хорошо
51 - 65	3	удовлетворительно
0 - 50	2	неудовлетворительно

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная учебная литература

1. Калентьев А.А. Новые технологии в программировании : учебное пособие / А.А. Калентьев, Д.В. Гарайс, А.Е. Горяинов – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9.- URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=480503. - (дата обращения: 22.03.2020). – Текст : электронный.

Дополнительная литература

1. Прыкина, Е.Н. Основы логического программирования в среде Турбо Пролог : учебное пособие / Е.Н. Прыкина. – Кемерово : КемГУКИ, 2006. – 68 с. – ISBN 5-8154-0130-7. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=227891. - (дата обращения: 22.03.2020). – Текст : электронный.
2. Баженов, Р.И. Лабораторный практикум по функциональному программированию: учебно-методическое пособие / Р.И. Баженов. – Москва; Берлин : Директ-Медиа, 2017. – 90 с. – ISBN 978-5-4475-9458-9.- URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=480173. - (дата обращения: 22.03.2020). – Текст : электронный.
3. Язык программирования Си++. Курс лекций : учебное пособие / Фридман А.Л. – Москва : ИНТУИТ.РУ «Интернет-университет Информационных Технологий», 2004. – 262 с. – ISBN 5-9556-0017-5. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=233058. - (дата обращения: 22.03.2020). – Текст : электронный.
4. Сузи, Р.А. Язык программирования Python: учебное пособие / Р.А. Сузи. – 2-е изд., испр. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 327 с. - ISBN 978-5-94774-711-9. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=233288. - (дата обращения: 22.03.2020). – Текст : электронный.
5. Мирошниченко, И.И. Языки и методы программирования: учебное пособие / И.И. Мирошниченко, Е.Г. Веретенникова, Н.Г. Савельева. – Ростов н/Д: Издательско-полиграфический комплекс Рост. гос. экон. ун-та (РИНХ), 2019. – 188 с. - ISBN 978-5-7972-2604-8. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=567706. - (дата обращения: 22.03.2020). – Текст : электронный.
6. Python : сайт. – 2001 - . - URL: <https://www.python.org/> (дата обращения: 22.03.2020). – Текст : электронный.
7. Предварительная спецификация C# 6.0 : сайт. – 2020 - . - URL:

<https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/language-specification/introduction> (дата обращения: 22.03.2020). – Текст : электронный.

8. Руководство по языку F# : сайт. – 2020 - . - URL: <https://docs.microsoft.com/ru-ru/dotnet/fsharp/language-reference/> (дата обращения: 22.03.2020). – Текст : электронный.

9. Visual Studio : сайт. – 2020 - . - URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 22.03.2020). – Текст : электронный.

10. Android Studio : сайт. – 2020 - . - URL: <https://developer.android.com/studio/intro> (дата обращения: 22.03.2020). – Текст : электронный.

11. IntelliJ IDEA : сайт. – 2000 - . - URL: <https://www.jetbrains.com/idea/> (дата обращения: 22.03.2020). – Текст : электронный.

Литература для оформления курсовой работы

1. Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный.

2. ГОСТ 19.701-90 (ИСО 5807-85) Единая система программной документации (ЕСПД). Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения : межгосударственный стандарт : издание официальное : введен впервые : дата введения 1992-01-01 / Москва Стандартинформ, 2010 – 158 с. – Текст: непосредственный.

СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ

CITForum.ru : on-line библиотека свободно доступных материалов по

информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.03.2020). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . – URL: <http://www.elibrary.ru> (дата обращения: 22.03.2020). – Режим доступа: для зарегистрир. пользлвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 - . – URL: <http://window.edu.ru/> (дата обращения: 22.03.2020). –Текст: электронный.

ПРИЛОЖЕНИЕ 1. Титульный лист
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Кемеровский государственный университет»

Факультет информатики, математики и экономики
Кафедра математики, физики и математического моделирования

Иванов Иван Иванович
гр. ПМИ-19-1

РАЗРАБОТКА ГРАФИЧЕСКОГО РЕДАКТОРА

Курсовая работа
по дисциплине «Языки и методы программирования»

по направлению подготовки 01.03.02 Прикладная математика и информатика
направленность (профиль) подготовки «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

Проверил:
канд. техн. наук, доцент
Е.В. Решетникова

Общий балл: _____

Оценка: _____

_____ подпись

«____» _____ 20____ г.

Новокузнецк 20_____

ПРИЛОЖЕНИЕ 2. Задание на курсовую работу

Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Кемеровский государственный университет»

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

УТВЕРЖДАЮ

заведующий кафедрой

_____ Е.В. Решетникова

« ____ » _____

ЗАДАНИЕ

на курсовую работу

студенту гр. ПМИ-19-1 **Иванову Ивану Ивановичу**

1. Тема курсовой работы «**Разработка графического редактора**» утверждена распоряжением декана _____.

2. Исходные данные: печатные и электронные учебные издания, ресурсы сети интернет.

Содержание работы (перечень подлежащих разработке вопросов):

Подобрать и проанализировать научную литературу по теме «**Разработка графического редактора**»; исследовать технологии программирования, языки программирования и среды разработки, оценить возможность их применения для решения поставленной задачи; разработать программное приложение с использованием выбранных информационных технологий.

3. Задание выдано: « ____ » _____ 20 ____ г.

4. Срок сдачи курсовой работы: « ____ » _____ 20 ____ г.

Руководитель канд. техн. наук, доцент _____ Е.В. Решетникова

Студент гр. ПМИ-19-1 _____ И.И. Иванов

ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы

**Тема: Разработка программного приложения для расчета объема
фигуры вращения, заданной произвольной кривой**

Введение

Актуальность исследования

Сегодня в мире информационных технологий широко распространены задачи 3D-моделирования объектов. Трехмерная графика считается незаменимым средством для демонстрации разного рода сложных технических узлов, многоступенчатых производств, архитектурных сооружений. Трехмерные модели наглядно показывают все особенности строения объекта, его мельчайшие элементы, которые скрыты от глаз наблюдателя. Также достаточно важной является задача расчета различных характеристик построенных моделей и дальнейшее их проецирование на оригинал. На основании вышеизложенного можно считать актуальной задачу разработки приложения, которое позволит осуществлять моделирование фигур вращения, заданных произвольной кривой, расчет их объема и площади фигуры под графиком.

Цель исследования: разработать программный продукт, моделирующего фигуры вращения произвольной формы, с последующим расчетом их объема и площади фигуры под графиком.

Задачи исследования:

1. анализ предметной области решения задачи;
2. выбор оптимальной технологии программирования, языка программирования и среды разработки для решения задачи;
3. построение алгоритма решения задачи;
4. разработка программного приложения;
5. тестирование разработанного программного приложения.

Теоретическая значимость исследования заключается в систематизации методов, применяемых при решении задачи расчета объема фигуры вращения, заданной произвольной кривой.

Практическая значимость исследования заключается в том, что выводы и результаты курсовой работы (формулы, алгоритм и программное приложение) могут быть использованы на практике при решении аналогичных задач.

ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Языки и методы программирования»

1. Разработка графического редактора.
2. Разработка редактора UML-диаграмм.
3. Разработка приложения моделирования и визуализации физических законов.
4. Разработка приложения «Автовокзал».
5. Разработка приложения моделирования движения транспортных средств на перекрестке.
6. Разработка приложения автоматизированного раскроя материала.
7. Разработка приложения «Библиотека».
8. Разработка приложения заказа такси.
9. Разработка приложения «Гостиница».
10. Разработка приложения для автостоянки.
11. Разработка приложения регистрации участников соревнования.
12. Разработка приложения для сервисного центра.
13. Разработка приложения «Планировщик».
14. Разработка приложения для автосервиса.
15. Разработка файлового менеджера.
16. Разработка приложения, моделирующего функционирование замкнутой биологической системы.
17. Разработка приложения для автобусного парка.
18. Разработка приложения моделирования планетарной системы.
19. Разработка приложения для вычисления площади произвольного многоугольника.
20. Разработка приложения моделирования муравейника.
21. Разработка компьютерной версии игры «Монополия».
22. Разработка приложения проведения on-line аукционов.
23. Разработка приложения заказа товаров.
24. Разработка приложения «Портфолио студента».

25. Разработка приложения «Туристическое агентство».
26. Разработка приложения моделирования химических процессов.
27. Разработка приложения формирования и отображения 3D-объектов.
28. Разработка приложения «Музей».
29. Разработка приложения «Зоопарк».
30. Разработка приложения моделирования улья.