

**Приложение 1 к РПД**  
**Языки и технологии программирования**  
**09.03.01 Информатика и вычислительная техника**  
**Направленность (профиль)**  
**Технологии разработки мобильных приложений**  
**Форма обучения – очная**  
**Год набора – 2022**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
**ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	09.03.01 Информатика и вычислительная техника
3.	Направленность (профиль)	Технологии разработки мобильных приложений
4.	Дисциплина (модуль)	Б1.О.14.02 Языки и технологии программирования
5.	Форма обучения	очная
6.	Год набора	2022

## **I. Методические рекомендации**

### **1.1 Методические рекомендации по организации работы студентов во время проведения лекционных занятий**

- В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации для практического занятия и указания для выполнения самостоятельной работы.
- В ходе лекционных занятий обучающемуся необходимо вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание изучаемой дисциплины, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве.
- Желательно оставить в рабочих конспектах поля, на которых делать пометки, подчеркивающие особую важность тех или иных теоретических положений. Рекомендуется активно задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

### **1.2 Методические указания к выполнению лабораторных работ**

- Лабораторные работы сочетают элементы теоретического исследования и практической работы. Выполняя лабораторные работы, обучающиеся лучше усваивают учебный материал, так как многие определения, казавшиеся отвлеченными, становятся вполне конкретными, происходит соприкосновение теории с практикой, что в целом содействует пониманию сложных вопросов науки и становлению обучающихся как будущих специалистов.
- Выполнение лабораторных работ направлено на:
  - обобщение, систематизацию, углубление теоретических знаний по конкретным темам учебной дисциплины;
  - формирование умений применять полученные знания в практической деятельности;
  - развитие аналитических, проектировочных, конструктивных умений;
  - выработку самостоятельности, ответственности и творческой инициативы.
- Лабораторные занятия как вид учебной деятельности должны проводиться в специально оборудованных лабораториях, где выполняются лабораторные работы (задания).
- Форма организации обучающихся для проведения лабораторного занятия – фронтальная, групповая и индивидуальная – определяется преподавателем, исходя из темы, цели, порядка выполнения работы.
- В ходе лабораторных занятий реализуется интерактивная форма взаимодействия – в виде самостоятельных заданий на лабораторных занятиях.

Результаты выполнения лабораторного работы оформляются обучающимися в виде отчета, форма и содержание которого определяются требованиями соответствующей работы.

### **1.3 Методические рекомендации по организации самостоятельной работы обучающихся**

- Самостоятельная работа – планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).
- Самостоятельная работа студентов (далее – СРС) в ВУЗе является важным видом учебной и научной деятельности студента. СРС играет значительную роль в рейтинговой технологии обучения. Обучение в ВУЗе включает в себя две, практически одинаковые по объему и взаимовлиянию части – процесса обучения и процесса самообучения. Поэтому СРС должна стать эффективной и целенаправленной работой студента.
- К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в практических занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом СРС играет решающую роль в ходе всего учебного процесса.
- В процессе самостоятельной работы студент приобретает навыки самоорганизации, самоконтроля, самоуправления, саморефлексии и становится активным самостоятельным субъектом учебной деятельности.
- Формы самостоятельной работы студентов разнообразны. Они включают в себя:
  - изучение учебной, научной и методической литературы, материалов периодических изданий с привлечением электронных средств официальной, статистической, периодической и научной информации;
  - подготовку докладов и рефератов, написание курсовых и выпускных квалификационных работ;
  - участие в работе студенческих конференций, комплексных научных исследованиях.
- Самостоятельная работа приобщает студентов к научному творчеству, поиску и решению актуальных современных проблем.
- Основной формой самостоятельной работы студента является изучение конспекта лекций, их дополнение, рекомендованной литературы, активное участие на практических и лабораторных занятиях.

### **1.4 Методические рекомендации по решению тестовых заданий**

- Тестовая система предусматривает вопросы/задания, на которые обучающийся должен дать один или несколько вариантов правильного ответа из предложенного списка ответов. При поиске ответа необходимо проявлять внимательность.
- При отсутствии какого-либо одного ответа на вопрос, предусматривающий множественный выбор, весь ответ считается неправильным.
- Ответы правильные выделяются в тесте подчеркиванием или любым другим допустимым символом.

### **1.5 Методические рекомендации по подготовке презентации**

#### **Алгоритм создания презентации:**

- 1 этап – определение цели презентации
- 2 этап – подробное раскрытие информации,
- 3 этап – основные тезисы, выводы.

### **Следует использовать 10-15 слайдов. При этом:**

- первый слайд – титульный, предназначен для размещения названия презентации, имени докладчика и его контактной информации;
- на втором слайде необходимо разместить содержание презентации, а также краткое описание основных вопросов;
- оставшиеся слайды имеют информативный характер.

Обычно подача информации осуществляется по плану: тезис – аргументация – вывод.

### **Требования к оформлению и представлению презентации:**

- Читательность (видимость из самых дальних уголков помещения и с различных устройств), текст должен быть набран 24-30-ым шрифтом.
- Тщательно структурированная информация.
- Наличие коротких и лаконичных заголовков, маркированных и нумерованных списков.
- Каждому положению (идее) надо отвести отдельный абзац.
- Главную идею надо выложить в первой строке абзаца.
- Использовать табличные формы представления информации (диаграммы, схемы) для иллюстрации важнейших фактов, что даст возможность подать материал компактно и наглядно.
- Графика должна органично дополнять текст.
- Выступление с презентацией длится не более 10 минут.

## **1.6 Методические рекомендации по подготовке доклада (эссе)**

### **Алгоритм создания доклада:**

- 1 этап – определение темы доклада
- 2 этап – определение цели доклада
- 3 этап – подробное раскрытие информации
- 4 этап – формулирование основных тезисов и выводов.

### **Требования к теме доклада (эссе):**

Темы докладов формулируются таким образом, чтобы расширить знания студента о конкретном программном продукте или компьютерном устройстве, а также дать представление о возможности и использования в профессиональной деятельности, например:

1. Назначение и возможности редакторов трехмерной графики.
2. Сравнительный анализ возможностей текстовых процессоров пакетов MS Office и LibreOffice.
3. Обзор возможностей настольной издательской системы MS Publisher на примере создания информационного буклета
4. Сублимационная печать. Назначение, преимущества и недостатки.

### **Требования к оформлению доклада:**

1. Объем доклада – 5 страниц (без титульного листа и списка источников).
2. Титульный лист должен быть оформлен по образцу (имеется файл с образцом).
3. Основной текст работы оформлен в соответствии с требованиями, указанными ниже.
4. В случае использования в тексте таблиц и/или рисунков на каждый объект должна быть ссылка в тексте работы. Например, «... основные виды программных средств представлены ниже (см. Таблица 1)» или «... схему передачи информации можно увидеть на рис. 1».
5. Количество источников должно быть не менее трех, на все должны быть ссылки внутри текста.
6. Список используемых источников должен быть оформлен в соответствии с требованиями, указанными ниже.

## 1.7 Методические рекомендации по подготовке к сдаче зачета

- Зачет осуществляется в рамках завершения изучения дисциплины и позволяет определить качество усвоения изученного материала, а также степень сформированности компетенций.
- Студенты обязаны сдавать зачет в строгом соответствии с утвержденными учебными планами, разработанными согласно образовательным стандартам высшего образования.
- По данной дисциплине зачет принимается по практической части курса. Задания, предлагаемые на зачете, соответствуют оценочным средствам для промежуточной аттестации обучающихся по предмету.
- Студент обязан не только представить правильно выполненные задания, но и защитить свое решение.
- Преподавателю предоставляется право задавать студентам вопросы по решению заданий практической части с целью выявления глубины понимания изученного материала и степени самостоятельности выполнения заданий.
- При явке на зачет студенты обязаны иметь при себе зачетную книжку, которую они предъявляют преподавателю в начале зачета.
- Рекомендуется при подготовке к зачету опираться на следующий план:
  1. Повторить теоретическую часть курса.
  2. После работы над теорией необходимо ответить на вопросы для самоконтроля.
  3. Повторить методы, способы и приемы решения задач по всем темам курса, опираясь на задания, которые решались на практических занятиях и предлагались для самостоятельной работы.
  4. Решить типовые задания по данной теме.

## II. Планы практических занятий (лабораторных работ)

### Раздел 1. Парадигмы, языки и технологии программирования

Лабораторные работы по разделу не предусмотрены

### Раздел 2. Разработка программ и реализация основных алгоритмических конструкций средствами выбранного языка программирования и среды разработки (ЛР1-ЛР4).

#### Лабораторная работа №1. Создание первого проекта на языке C++ в среде Qt. Решение СЛАУ методом Крамера

Цель: написать программу, реализующую метод Крамера для решения системы линейных алгебраических уравнений (СЛАУ) размером  $(2 \times 2)$ .

План:

1. Познакомиться с особенностями создания, сборки и запуска проекта на языке C++ в среде Qt.
2. Реализовать анализ и решение СЛАУ размером  $(2 \times 2)$  методом Крамера с полным исследованием (учёт всех возможных вариантов значений параметров).

#### Лабораторная работа №2. Использование основных синтаксических конструкций языка C++: ввод/вывод и управляющие инструкции

Цель: написать программу, использующую инструкции ввода/вывода данных, инструкции условия и цикла.

План:

1. Подобрать идентификаторы переменных и констант. Для каждого идентификатора указать в комментариях его смысловое назначение в терминах условия задачи.
2. Для каждого использованного типа указать название на русском языке, количество байт для хранения, диапазон принимаемых значений.
3. Составить и описать три теста: для произвольного вводимого значения(ий) параметра(ов); для одного из параметров значение, предполагающее альтернативную обработку по сравнению с предыдущим или в некотором смысле граничное (пояснить, почему так можно толковать); тест с недопустимым значением одного из параметров.
4. Выполнить программную реализацию решения задачи.

Вопросы для самоконтроля (защита ЛР1-ЛР2):

1. Для чего используется пространство имён?
2. Почему заголовочный файл *iostream* указан в угловых скобках и без расширения *.h*?
3. Сколько различных *Makifile* сгенерировала утилита *qmake*?
4. Почему параметры функции в макроподстановке следует заключать в круглые скобки?
5. Сформулируйте отличия макроопределения (макроподстановки) для функции и встраиваемой (*inline*) функции.
6. Почему встраиваемые функции рекомендуется определять в заголовочном файле?
7. Обязательно ли объявлять каждую функцию в заголовочном файле?
8. Каковы достоинства и недостатки макроопределения для функции в сравнении с обычной функцией?
9. Каковы достоинства и недостатки встраиваемой функции в сравнении с обычной функцией?
10. Какая комбинация клавиш используется в текстовом редакторе *Qt* для активации автодополнения исходного кода?
11. Какая комбинация клавиш используется в текстовом редакторе *Qt* для комментирования/раскомментирования блока исходного кода?
12. С какого значения по умолчанию индексируются массивы в языке C++?

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.

Литература: [1-6, вводные главы]

**Лабораторная работа №3. Использование основных синтаксических конструкций языка C++.** Решение СЛАУ размером (3×3) методами Крамера и Гаусса-Жордана. Вычисление определителей малых порядков

Цель: написать программу, реализующую метод Крамера, Гаусса-Жордана для решения СЛАУ размером (3×3) или отдельные этапы решения в соответствии с этими методами.

План:

1. Составить алгоритм решения задачи по варианту.
2. Проанализировать возможные ситуации в решении, включая те, которые не противоречат типам входных данных, но не могут быть обработаны корректно (исключительные ситуации), о чём следует сообщить пользователю (продумайте текст выводимого сообщения). Составить и записать тесты для каждой ситуации (всего не менее трёх тестов).
3. Выполнить программную реализацию решения задачи.
4. Оценить условное время выполнения (временную сложность) программы, используя профиль для теста, который не содержит исключительную ситуацию, и аналитический профиль.
5. Оценить объёмную сложность программы.

Вопросы для самоконтроля (защита ЛР3):

1. Что такое исключительная ситуация в алгоритме и в программе?
2. Сравните два способа обработки исключительных ситуаций: недопущение (с помощью условного оператора) и постобработка (с помощью *try-throw-catch*).
3. Что такое временная (вычислительная) сложность алгоритма (программы)?
4. Что такое объёмная сложность программы?
5. Что такое профиль и аналитический профиль теста?

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.

Литература: [1], [2], [4, с. 3-231], [6, с. 5-98].

## Лабораторная работа №4. Средства отладки Qt. Способы передачи параметров в функцию. Работа с текстовыми файлами. Вычисление значения детерминанта конечного порядка по определению

Цель: написать программу, реализующую вычисление определителя произвольного конечного порядка по определению, используя основные синтаксические конструкции языка C++.

План:

1. Познакомиться с отладчиком Qt.
2. Допisać программу для вычисления определителя квадратной матрицы, используя имеющийся листинг.
3. Сравнить результат расчётов с результатом, полученным с помощью других программных средств, например, MS Excel (функция МОПРЕД).
4. Организовать считывание трёх определителей разных порядков из текстового файла;
5. Получить значения целевых переменных (по вариантам) на заданной итерации с помощью отладчика;
6. Записать для каждого определителя лог промежуточных вычислений в текстовый файл.

Вопросы для самоконтроля (защита ЛР4):

1. Что такое отладчик?
2. Что такое механизм точек останова?
3. Какие варианты перехода от одной инструкции к другой поддерживает отладчик?
4. Каким образом можно узнать значения переменных и выражений до выполнения заданной инструкции программы?
5. Для чего используются статические переменные?
6. Для чего используется оператор «,» в инструкции цикла с параметром?
7. Перечислите все присваивающие арифметические операторы.
8. Приведите пример использования операторов `break` и `continue`.
9. Смоделируйте и запишите ситуации, в которых могут быть использованы одна или несколько перечисленных ниже сигнатур некоторой функции `f`. Постарайтесь для каждой сигнатуры предложить хотя бы одну ситуацию, в которой именно такая сигнатура является рекомендуемой (укажите почему), или обосновать, почему она не может быть приоритетной ни в каком случае:
  - 1) `void f(int a, int b);`
  - 2) `void f(const int a, const int b);`
  - 3) `void f(int* a, int* b);`
  - 4) `void f(const int* a, const int* b);`
  - 5) `void f(int& a, int& b);`
  - 6) `void f(const int& a, const int& b);`
10. Какие из перечисленных выше вариантов содержат способ передачи аргументов:
  - 1) по значению,
  - 2) по ссылке,
  - 3) по указателю (адресу)?
11. Какие из перечисленных выше вариантов подойдут для записи функции `swap` (обмен значений двух переменных – параметров функции – делается внутри функции; эти же переменные с обновлёнными значениями должны быть доступны для чтения и изменения также и вне функции)?
12. Запишите тело функции `swap` и примеры её вызова в каждом из случаев предыдущего пункта, считая, что она применяется к элементам целочисленного одномерного массива с изменяемыми значениями.

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.

Литература: [3, с. 759-792], [4, с. 72-89].

**Раздел 3. Обработка отдельных типов данных средствами выбранного языка**

## программирования и среды разработки (ЛР5-ЛР8).

### Лабораторная работа №5. Способы реализации массивов и простейших списков

Цель: научиться использовать различные варианты реализации массивов и простейших списков.

План:

1. Создать одномерный пятиэлементный массив на стеке, с использованием указателя и в куче средствами STL (vector, list), заполнив его случайными значениями.
2. Проанализировать способ хранения массива в каждом случае – адреса элементов.
3. Создать двумерный целочисленный массив размером (4×4) двумя способами: как статический массив и как указатель на указатели `int** a = new int*[4]`.
4. Проанализировать способ хранения матриц в каждом случае – адреса элементов.
5. Транспонировать каждую из матриц и вывести значения на консоль или в файл.
6. Проанализировать возможность передачи матрицы в функцию с параметром:
  - 1) `int a[][4]`;
  - 2) `int* a`;
  - 3) `int** a`.

### Лабораторная работа №6. Обработка строк

Цель: научиться выполнять основные действия со строками.

План:

1. Создать входной текстовый файл содержит некоторый текст на русском языке, в котором границы предложений и строк не согласованы. Словом будем считать последовательность символов, отделённую от других пробелами, знаками пунктуации или символом ‘\n’. Словом также может быть целое число без знака или со знаком «-» без отрыва от цифр. Предложением назовём совокупность слов, заканчивающуюся точкой. Опираясь на задание по выбору, подготовьте осмысленный текст, состоящий не менее чем из 20 предложений и общей длиной не менее 500 символов. Каждое предложение должно содержать не более 255 символов.
2. Написать программу, которая форматирует текст так, чтобы каждая строка содержала ровно одно предложение.
3. Если в тексте есть лишние пробелы – перед первым словом, между словами и перед точкой в предложениях, удалить их (допишите соответствующий код в программу).
4. Удалить программно повторяющиеся рядом стоящие слова (при каждом повторе текст содержит не более двух одинаковых рядом стоящих слов).
5. Полученный текст вывести в файл, который будет входным для новой программы – задания по выбору.

Вопросы для самоконтроля (ЛР5-ЛР6):

1. Каковы особенности размещения программы (исполняемого/объектного кода) и данных в памяти?
2. Можно ли с помощью арифметики указателей выйти за пределы области данных, выделенной операционной системой? Можно ли оказаться в области, где хранится сама программа? Тот же вопрос для областей других программ и ячеек, которые занимает операционная система.
3. Охарактеризуйте разделы памяти: статическая область, стек, куча?
4. Какой размер является типовым для каждого раздела?
5. Что такое утечка памяти?
6. Что такое порча памяти?
7. Что означает разрушение созданного в памяти объекта?
8. Что такое автоматическая сборка мусора применительно к памяти?
9. Есть ли в C++14 автоматическая сборка мусора?

10. Приведите примеры языков, в которых механизм автоматической сборки мусора реализован. Каковы его достоинства и недостатки? Что такое проблема кольцевых ссылок?
11. В чём отличие абстрактного типа данных от структуры данных?
12. Что такое базовый и пользовательский типы данных?
13. Сформулируйте сходства и отличия типов стандартной библиотеки: array, vector, list.
14. Перечислите не менее 10 функций, которые доступны для объекта класса string (или QString).

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.  
Литература: [3, с. 333-451, 603-607], [4, с. 38-46, 71-76], [5, с. 232-420].

### **Лабораторная работа №7. Создание оконного приложения**

Цель: научиться создавать приложения с графическим интерфейсом.

План: необходимо создать интерфейс для взаимодействия с пользователем при решении задачи по выбору из лабораторной работы №6. Функциональная спецификация будет следующая:

- 1) загрузка исходного текста из текстового файла (выбор файла с помощью стандартного диалога) с отображением в многострочном текстовом поле с поддержкой полос прокрутки;
- 2) преобразование текста, как в предыдущей лабораторной и отображение нового текста в другом текстовом поле с аналогичными свойствами;
- 3) сохранение нового текста в текстовый файл с указанием файла с помощью стандартного диалога.

Нефункциональное требование – возможность выполнения чтения, преобразования и записи текста тремя способами: из главного меню, при помощи панели инструментов и кнопок на форме.

Вопросы для самоконтроля (защита ЛР7):

1. Какой смысл спецификаторов доступа public, private, protected?
2. Какой макрос отвечает за информирование компилятора метаобъектов о том, что заголовочный файл содержит описание класса.
3. Что такое очередь событий?
4. Что такое функциональная спецификация программы?
5. Что такое сигнал? Что такое слот?
6. Как механизм сигналов и слотов обеспечивает безопасность типов?
7. Приведите примеры менеджеров компоновки виджетов Qt. Для чего они нужны?
8. Какие действия необходимо выполнить в режиме Дизайн для создания пунктов главного меню и размещения кнопок на панели инструментов?
9. Что можно считать ресурсами проекта QWidget и как организован доступ к ним?
10. В файл какого типа преобразуется файл ресурсов с помощью компилятора ресурсов?
11. Выясните, очищается ли автоматически память, занимаемая созданными локально в функциях open, process и save объектами классов QFile, QString, QTextStream при выходе из этих функций.
12. Для чего используется класс QDebug? Приведите пример и прокомментируйте.

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.  
Литература: [1, с. 607-676], [3, с. 53-71, 119-130].

### **Лабораторная работа №8. Подключение сторонних библиотек (на примере OpenGL).**

Цель: научиться подключать и использовать API сторонних библиотек (на примере OpenGL).

План:

1. Исследовать готовый проект MyOpenGL:



- 1.1. Описать наблюдаемую трёхмерную сцену при запуске и взаимодействии с программой.
- 1.2. Указать, какие элементы управления использованы для GUI и как пользователь взаимодействует с трёхмерной сценой с их помощью.
- 1.3. Сформулировать имеющуюся проблему управления сценой.
- 1.4. Предложить варианты решения (запишите на естественном языке и попробуйте реализовать их в коде).
2. Создать облако точек.
  - 2.1. Добавить в сцену 4 вершины, указав координаты таким образом, чтобы получился квадрат. Указать для вершин цвет.
  - 2.2. Развернуть окно создаваемого приложения во весь экран.
  - 2.3. Добавить ещё несколько вершин и поэкспериментировать со способами их соединения: GL\_LINES, GL\_LINE\_STRIP, GL\_LINE\_LOOP, GL\_TRIANGLE\_STRIP, GL\_POLYGON, GL\_QUADS.
  - 2.4. Добавить возможность пользователю с помощью кликов левой кнопкой мыши добавлять серии точек и выбирать способ их соединения, когда серия закончилась (из контекстного меню окна).
3. Осуществить геометрические преобразования (движения) в трёхмерной сцене
  - 3.1. Выполнить построение пирамиды в трёхмерной системе координат. Добавьте возможность выполнения поворота пирамиды с помощью кликов левой кнопкой мыши.
  - 3.2. Внести в код изменения, чтобы при нажатиях левой кнопки мыши выполнялись действия из предыдущего пункта, а при нажатии правой осуществлялся параллельный перенос пирамиды в точку клика. Сделать анимацию такого сдвига.

Вопросы для самопроверки (защита ЛР8):

1. Является ли OpenGL платформонезависимой библиотекой?
2. Является ли OpenGL объектно-ориентированной?
3. Что такое контекст воспроизведения OpenGL?
4. Перечислите шесть основных классов, которые реализуют обёртку для библиотеки OpenGL в составе Qt.
5. Какие матрицы преобразования координат имеются в OpenGL?
6. Что такое видовое окно (viewport)?

Самостоятельная работа: Изучение литературы, подготовка к защите лабораторной работы.  
Литература: [4, с. 350-365].