

Министерство науки и высшего образования Российской Федерации

Томский государственный университет
систем управления и радиоэлектроники

И. Л. Артемов

Программирование

Методические указания к лабораторным работам и самостоятельным занятиям студентов
всех форм обучения, обучающихся по направлению подготовки
«Мехатроника и робототехника»

Томск
2024

УДК 621.01
ББК 334
©А86

Рецензент:

Антипин М. А., доцент каф. управления инновациями ТУСУР, канд. физ.-мат. наук.

Артемов Игорь Леонидович

Программирование: методические указания к лабораторным работам и самостоятельным занятиям / Артемов И. Л. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2024. – 17 с.

Методические указания к лабораторным работам и самостоятельным занятиям по дисциплине «Программирование» разработаны для студентов бакалавриата, обучающихся по направлению подготовки 15.03.06 «Мехатроника и робототехника». Методические указания содержат необходимые разъяснения по форме организации лабораторных работ и ориентированы на достижение результатов образовательной деятельности в соответствии с требованиями федерального государственного образовательного стандарта высшего образования.

Одобрено на заседании каф. управления инновациями, протокол № 4 от 23.11.2023

УДК 621.01
ББК 334

© Артемов И. Л. 2024
© Томск.гос. ун-т систем упр. и
радиоэлектроники, 2024

Оглавление

Оглавление.....	3
Введение	4
Материально-техническое обеспечение лабораторных работ	5
Задания для лабораторных работ	6
Прием результатов выполнения лабораторных работ	15
Заключение	16
Список используемых источников.....	17

Введение

Лабораторные работы по дисциплине «Программирование», предусмотренные настоящими указаниями, выполняются студентами индивидуально во время аудиторных занятий. Все консультации осуществляются преподавателем.

Студент может использовать ресурсы сети Интернет для составления программ. Во время проведения лабораторных работ в аудитории студентам запрещается передавать друг другу файлы и другие материалы, являющиеся результатом выполнения заданий.

Студент имеет право просить консультации у преподавателя, если он в текущий момент не распределяет задания, не принимает выполненные работы и не консультирует другого студента. Преподаватель, давая консультацию студенту, указывает раздел технической документации или методической литературы, в которой имеется ответ на вопрос студента. Если необходимые сведения в документации и литературе отсутствуют, то преподаватель должен дать устные пояснения или продемонстрировать практические действия, приводящие к требуемому результату, с последующим повторением студентом.

Консультации, выдача лабораторных работ и прием результатов выполнения осуществляется только во время аудиторных занятий. Задания выполняются последовательно. Правильное выполнение некоторых заданий возможно только, если студент корректно выполнил предыдущие задания.

При сдаче лабораторных работ преподаватель задает вопросы студенту по работе программы, используемым языковым конструкциям и возможностям, алгоритмам работы программы, входным и выходным данным.

Форма представления результатов лабораторных работ - составленные программы.

Материально-техническое обеспечение лабораторных работ

Учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения лабораторных работ, учебная аудитория для проведения занятий семинарского типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации.

634034, Томская область, г. Томск, Вершинина улица, д. 74, 414 ауд.

Описание имеющегося оборудования:

- Проектор NEC v260x;
- Экран проекторный;
- Доска маркерная;
- Компьютер (13 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- Microsoft Windows 10
- Microsoft Visual Studio 2022

Размещение и освещенность рабочих мест в учебной аудитории должно удовлетворять действующим требованиям санитарных правил и норм (СанПиН).

Задания для лабораторных работ

Тема занятия 1 – Отладка линейного алгоритма по вычислению математического выражения.

Цель занятия: познакомиться со средой разработки, выполнять простейшую отладку приложения.

Задания для студентов:

Вариант 1

$$f(x) = \sqrt[3]{x + \sin x} + \frac{\sqrt[5]{50x - \cos x}}{(x + \sin x)x^3} \qquad g(x) = 2^{\frac{\sin \frac{1}{x^2+x+1}}{x^2+x+1}} + \frac{x^2 + \sqrt{x}}{|x|\sin x}$$

$$f(x) = \frac{1}{x \sin x} + \sqrt[3]{\frac{x + \cos x}{\sin x + \frac{1}{\sin x + 1}}} \qquad g(x) = \frac{x \cos x}{x + 1} - \frac{\sin x}{x^{4.5} + 2x + x^2}$$

Вариант 2

$$f(x) = \frac{2^{\sqrt[3]{x^2}}}{\sin x \cdot e^x} + \frac{1}{x^3} + \frac{1}{\frac{\sin x + \cos x}{x \cos x}} \qquad g(x) = \frac{(x+1)^2}{(x+2)^{2.5}} + \sqrt[5]{(x+2)^3 + x \sin x}$$

$$f(x) = \frac{e^{7x} - |\cos x + e^x|}{1 + \sin x + \frac{2}{x^2 + x^5}} + (1 + \cos x)^7 \qquad g(x) = \operatorname{tg} x + \frac{\cos x}{\sqrt[4]{1 + \sqrt{x + \sin \frac{1}{x+1}}}}$$

Вариант 3

$$f(x) = \frac{\ln^{3.5}(x + \sqrt{x}) + \ln(\sin x + e^x)}{\sqrt[3]{\frac{x}{e^x \sin x}}} \qquad g(x) = \ln(\sin x + \sqrt{x+1}) + \frac{\sqrt{x-1}}{x \sin x}$$

$$f(x) = \frac{\sqrt[3]{x + \sin x} + \sqrt{x^2 \sin x}}{|x + \cos^2 x|x} \qquad g(x) = \frac{\sin x + \sqrt{x - \cos x}}{\sin x \cdot \sqrt[3]{40x - \sin x}}$$

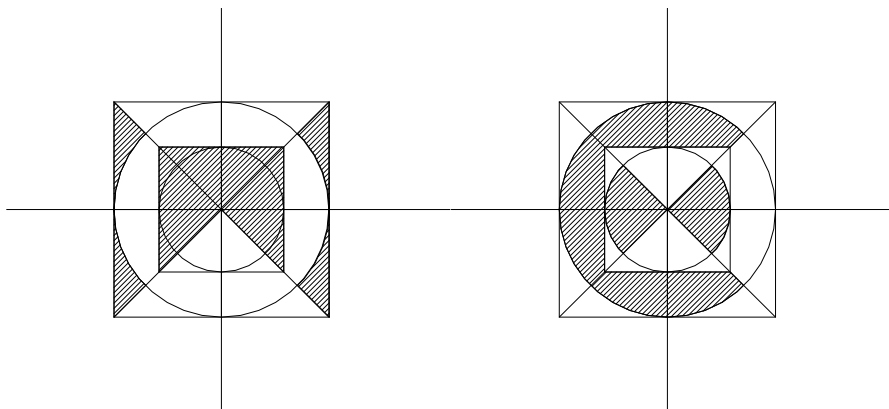
Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 2 – Программирование логических выражений.

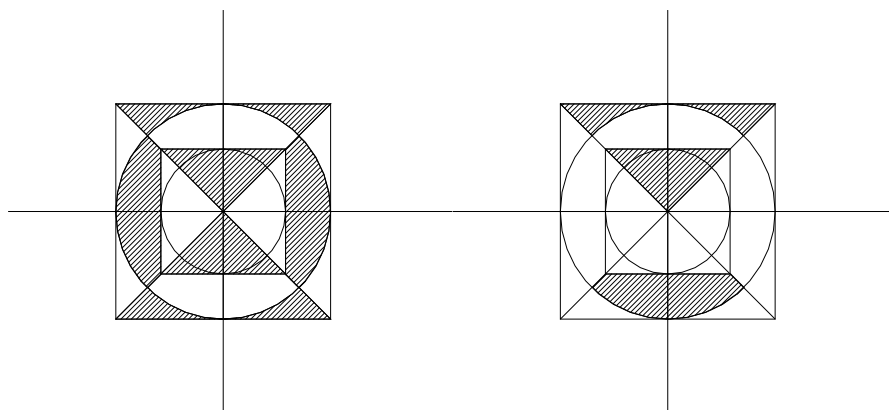
Цель занятия: изучение логических операций языка C++.

Задания для студентов. Записать условие попадания в область.

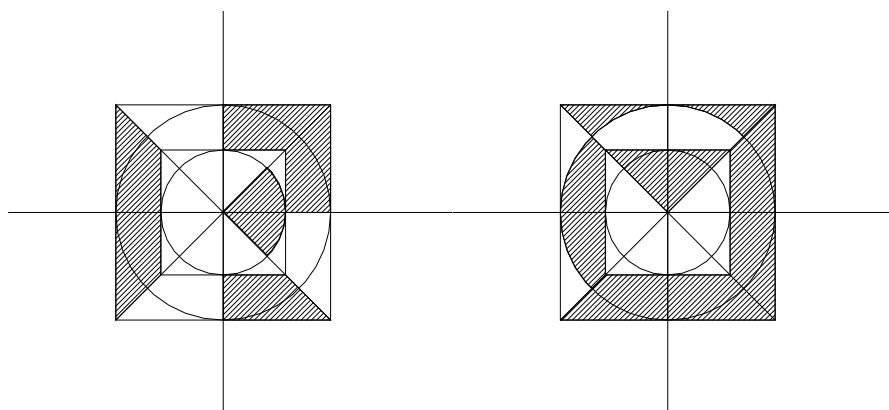
Вариант 1



Вариант 2



Вариант 3



Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

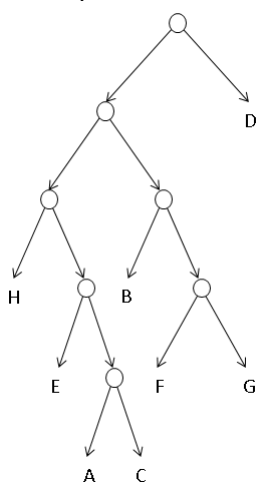


Тема занятия 3 – Программы с вложенными операторами if и switch. Попадание в интервалы.

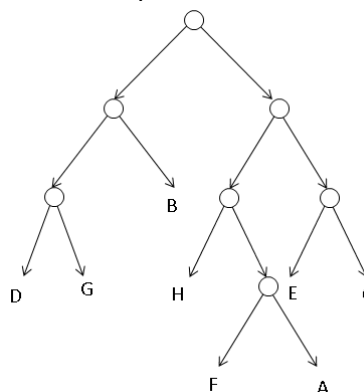
Цель занятия. Закрепить полученные знания по использованию условного оператора. Задано восемь интервалов на числовой оси. Написать программу для определения попадания точки в заданный интервал. Начертить схему алгоритма попадания в интервал. Решить задачу путем деления интервала пополам. Подготовиться к составлению программы по заданному преподавателем алгоритму.

Задания для студентов:

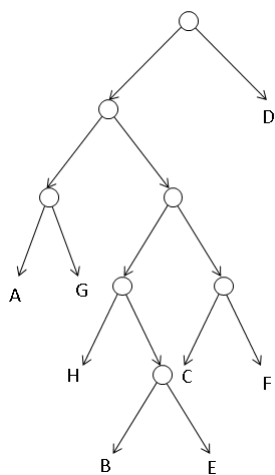
Вариант № 1



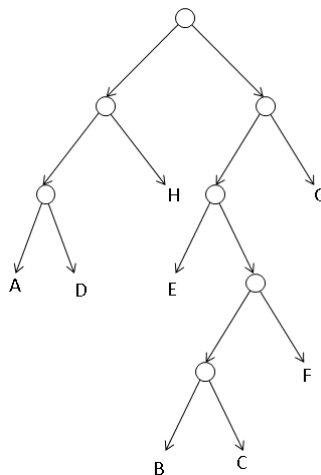
Вариант № 2



Вариант № 3



Вариант № 4



Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для

пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 4 – Обработка последовательностей чисел. Создание генератора табличных данных.

Цель занятия: Обработка последовательностей данных на C++.

Задания для студентов:

1. Дан файл, содержащий обыкновенные дроби, которые записаны каждая в отдельной строке. Числитель и знаменатель состоят из цифр от 1 до 9. Количество чисел в файле порядка нескольких гигабайт. Перемножьте все дроби и дайте ответ в виде несократимой дроби. Дроби могут быть распределены по любому закону. Требуется написать программу, которая будет максимально быстрой и использовать минимальное количество памяти.
2. Сгенерировать данные для тестирования вышеприведенной программы. Проанализировать предоставленные алгоритмы другими студентами и предложить свой вариант данных (если возможно) для «выведения из строя» работы программы.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 5 – Сортировка Шелла. Перестановка элементов..

Цель занятия: программирование алгоритма сортировки Шелла.

Задания для студентов:

1. Реализовать сортировку методом Шелла.
2. Реализовать перемешивание букв в слове из восьми разных букв. Оценить среднее количество попыток. Вычислить теоретическую вероятность и проверить с помощью численного эксперимента.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 6 – Обработка массивов с помощью указателей.

Цель занятия: разбор и понимание работы с указателями.

Задания для студентов:

1. Составить программу, которая показывает связь массивов и указателей.
2. Использовать отладчик для демонстрации и понимания работы с указателями.
3. Разобрать хранение двумерного массива с помощью указателей.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 7 – Проверка текстовой информации на соответствие определенным правилам (корректность адреса электронной почты, ФИО клиента, адреса проживания).

Цель занятия: изучение возможностей строк для хранения символьной информации. Применение строковых функций.

Задания для студентов:

1. Написать программу, которая проверяет корректность адреса электронной почты. Для примера использовать адрес от компании mail.ru.
2. Написать генератор русских фамилий, оканчивающихся на «ов», «ова», «ев», «ева».

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 8 – Создание и обработка простейшей таблицы с данными (сотрудники компании, товары на складе).

Цель занятия: использованием структур для описания сложных типов данных.

Задания для студентов:

1. Используя ранее разработанные генераторы фамилий составить случайную базу сотрудников, со случайными адресами проживания, электронной почты и телефонов.
2. Вывести данные по интересующим сотрудникам с помощью запроса у пользователя фамилии, номеров телефонов, электронной почты.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 9 – Файловые операции, ведение log файлов.

Цель занятия: научиться использовать файловый ввод/вывод при составлении программ.

Задания для студентов:

1. Написать программу, которая читает исходный код программы на C++ и заменяет все открывающиеся скобки на закрывающиеся.
2. Произвести ведение лог-файла по изменению файлов на C++.

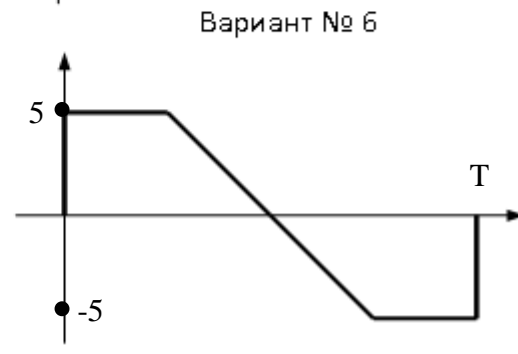
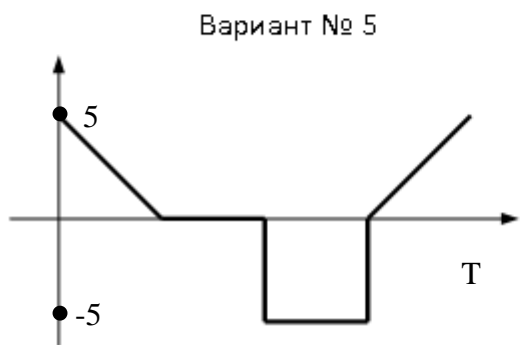
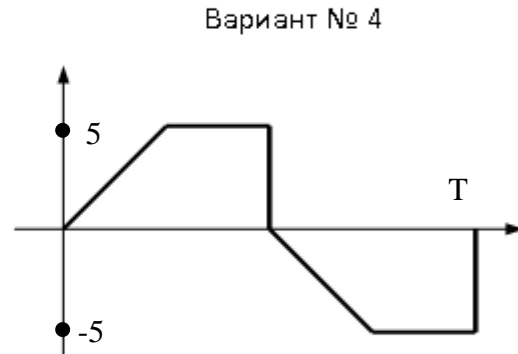
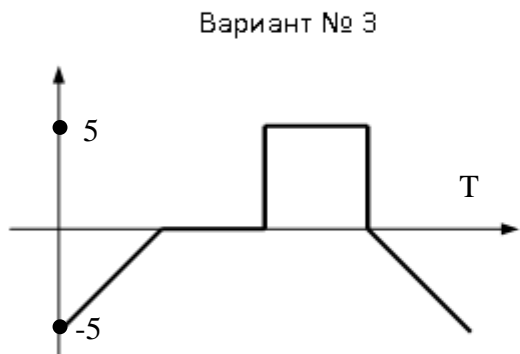
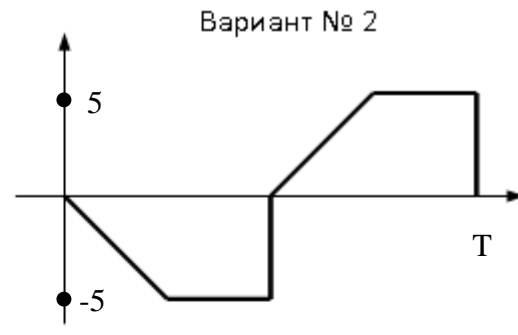
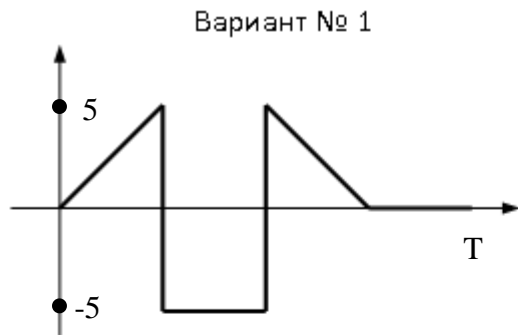
Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 10 – Программирование математических функций (ряды Фурье, Тейлора)..

Цель занятия: научиться использовать функции на C++.

Задания для студентов:

Задать на интервале $[0; T]$ следующие функции, заданные в виде графиков. Допустимые значения функций от -5 до 5. Вывести таблицу значений функции в 100 точках в текстовый файл и построить график в одном из редакторов (табличный редактор, математический пакет, онлайн-сервис).



Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 11 – Рекурсивный алгоритм вычисления определителя матрицы.

Цель занятия: программирование рекурсивных функций на C++.

Задания для студентов:

1. Написать рекурсивную функцию вычисления определителя.
2. Определить численным путем максимальный порядок определителя, который возможно вычислить в течение одного дня.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 12 – Написание функций для решения квадратного уравнения и линейного уравнений заданного строкой.

Цель занятия: использование строк для парсинга выражений.

Задания для студентов:

1. Написать функцию решения квадратного уравнения заданного строкой. Вызов должен происходить следующим образом `SolveQuadEq(expression, x1, x2)`;
2. Написать функцию решения линейного уравнения заданного строкой. Вызов должен происходить следующим образом `SolveLinearEq(expression, x)`;

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 13 – Создание динамического массива фигур.

Цель занятия: создание и работа с динамическими массивами.

Задания для студентов:

Создать динамический массив со следующей структурой. Организовать ввод данных для масштабирования массива и вывод полученных данных.

Вариант 1

1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	2	3												
1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	2	3												

Вариант 2

1														
1	2	3												
1	2	3	1	2	3									
1	2	3	1	2	3	1	2	3						
1	2	3	1	2	3	1	2	3	1	2	3			

Вариант 3

1														
1	2													
1	2	3												
1	2	3	3											

Вариант 4

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5										
1	2	3	4	5										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 14 – Использование стека для проверки скобочного выражения. Алгоритм Дейкстры.

Цель занятия: изучение стека, применение стека для прикладных задач.

Задания для студентов:

1. Дано скобочное выражение, состоящее из круглых, квадратных и прямоугольных скобок. Написать программу, которая проверяет правильность вложения скобок друг в друга по правилам математики. Например, выражение $((() () \{ \} [] [])$ - верное, $\{ () \}$ – ошибочное. Для решения задачи использовать стек по правилу, если текущая скобка будет закрывать скобку на вершине стека, то вытолкнуть из стека скобку, в противном случае поместить текущую скобку в стек. По окончании перебора скобок сделать проверку стека, если стек пуст то выражение – верное.

2. Написать программу, которая будет вычислить арифметическое выражение по двухстековому алгоритму Дейкстры. Записать исходное выражение в виде полного скобочного выражения, т.е. оформить каждую операцию круглыми скобками. Создать два стека: стек чисел и стек операций и разобрать выражение по правилу:

- 1) извлекаем из строки очередной символ;
- 2) числа помещаем в стек чисел;
- 3) операции помещаем в стек операций;
- 4) открывающаяся скобка игнорируется;
- 5) если встречается закрывающаяся скобка, то вытолкнуть из стека операций операцию, вытолкнуть из стека чисел два числа;
- 6) выполнить над числами операцию и поместить результат обратно в стек чисел.
- 7) взять следующий символ выражения.

Результат вычисления выражения будет находиться в стеке чисел.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 15 – Табулирование произвольной функции двух переменных.

Цель занятия: изучение указателя на функцию.

Задания для студентов:

Объявить пару функций двух переменных. Написать функцию по табулированию функции двух переменных на отрезках от x_0 до x_n и от y_0 до y_n , с количеством точек N_x и N_y . Вызов функции должен быть таким: `GetTable(FXY,x0,xn,y0,yn,Nx,Ny);`

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 16 – Работа с датой и временем.

Цель занятия: изучение библиотеки стандартных функций.

Задания для студентов:

Написать функцию, которая выдает текстовую строку о текущем времени.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 17 – Программирование операций матричной алгебры..

Цель занятия: изучение численных алгоритмов.

Задания для студентов:

Написать программу по вычислению обратной матрицы методом Гаусса-Жордана.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Тема занятия 18 – Алгоритм Ньютона для решений нелинейных уравнений.

Цель занятия: изучение численных алгоритмов.

Задания для студентов:

Написать программу по нахождению корня нелинейного уравнения методом Ньютона.

Задание для самостоятельной работы: Провести доработку создаваемой программы, снабдить программу необходимыми комментариями, оформить необходимый интерфейс для пользователя, выполнить тестирование написанной программы, проверить работу программы на разных входных данных.

Прием результатов выполнения лабораторных работ

Результаты выполнения лабораторных работ демонстрируются преподавателю. Во время приема выполненной работы преподаватель вправе:

- требовать у студента демонстрации выполненного задания в виде файлов, текстов, таблиц, мнемосхем, рисунков, в том числе, по возможности и необходимости, в бумажном письменном или распечатанном виде, либо в электронном виде;
- требовать у студента пояснений, относящихся к способам реализации задания.

Задание считается выполненным и принимается преподавателем только в том случае, если получены все результаты, предусмотренные заданием. Если какие-то результаты, предусмотренные заданием, не получены или неверны, то задание подлежит доработке.

До начала экзаменационной сессии студент должен сдать результаты выполнения всех лабораторных работ, предусмотренных настоящими указаниями. В противном случае студенты к сдаче зачета не допускаются.

Заключение

Выполнение методических указаний к лабораторным работам по дисциплине «Программирование» способствует успешному ее освоению и развитию у обучающихся навыков написания программ с использованием языка C++.

Список используемых источников

1. Кривцов, А. Н. Технологии программирования. Технология программирования на C/C++ : учебное пособие / А. Н. Кривцов. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2021. — 274 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/279680> (дата обращения: 12.12.2022).
2. Мариус, Б. Решение задач на современном C++ / Б. Мариус ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2019. — 302 с. — ISBN 978-5-97060-666-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/123704> (дата обращения: 12.12.2022).