

ВЫСШИЙ КОЛЛЕДЖ «ПОЛИТЕХНИК»

УТВЕРЖДАЮ
Заместитель директора по УМР
Е.Ю. Кузнецов
«26» июня 2020г.

2020

РАССМОТРЕНА И ОДОБРЕНА

Предметно-цикловой комиссией

Протокол № 7

« 25 » июня 20 20 г.

Председатель ПЦК  /Л.И.Логинова/

Разработчик: Кречетов Александр Александрович, преподаватель с ученой степенью кандидата технических наук, доцент кафедры ИВС ФГБОУ ВО «Поволжский государственный технологический университет».

Методические рекомендации предназначены для обучающихся специальности 10.02.05 Обеспечение информационной безопасности автоматизированных систем и направлены на оказание практической помощи при выполнении практических работ по дисциплине ОП.03 Основы алгоритмизации и программирования.

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ
2. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
3. ТЕМАТИКА, СОДЕРЖАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ
ЛАБОРАТОРНЫХ РАБОТ
4. КОНТРОЛЬ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ И КРИТЕРИИ
ИХ ОЦЕНКИ
5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ
ОБЕСПЕЧЕНИЕ

1. ВВЕДЕНИЕ

Методические рекомендации предназначены в качестве методических материалов при проведении лабораторных работ по дисциплине Основы алгоритмизации и программирования для специальности 10.02.05 Обеспечение информационной безопасности автоматизированных систем, составлены в соответствии с требованиями Федерального государственного образовательного стандарта по специальности среднего профессионального образования.

Теоретический материал курса Основы алгоритмизации и программирования охватывает обширный круг актуальных вопросов по организации, ведению и управлению хозяйственной деятельности в организации. Методические указания позволят улучшить усвоение учебного материала, изученного на лекционных занятиях. Обучающиеся смогут овладеть и свободно оперировать техническими категориями по различным областям деятельности организации. Решение практических задач, сформированных в данных методических указаниях, позволит студентам укрепить знания теоретического материала по указанной дисциплине.

Лабораторные занятия проводятся после изучения соответствующих разделов и тем учебной дисциплины. Так как учебная дисциплина имеет прикладной характер, то выполнение обучающимися практических работ позволяет им понять, где и когда изучаемые теоретические положения, и практические умения могут быть использованы в будущей практической деятельности.

2.ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Методические рекомендации по выполнению лабораторных работ разработаны в соответствии с рабочей программой учебной дисциплины Основы алгоритмизации и программирования среднего профессионального образования 10.02.05 Обеспечение информационной безопасности автоматизированных систем.

Целью лабораторных работ является закрепление теоретических знаний и приобретение практических умений по определению организационно-правовых форм организаций, расчету по принятой методике основных технических показателей деятельности организации, организации контроля на предприятии и др.

В результате выполнения лабораторных работ по дисциплине ОП.03 Основы алгоритмизации и программирования обучающийся должен овладеть предусмотренными ФГОС умениями, знаниями, которые формируют общие и профессиональные компетенции.

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ПК 2.1. Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.

ПК 2.2. Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.

ПК 2.3. Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.

ПК 2.4. Осуществлять обработку, хранение и передачу информации ограниченного доступа.

П.К2.6. Осуществлять регистрацию основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак.

При выполнении лабораторной работы обучающийся должен уметь:

- работать в среде программирования;
- использовать языки программирования высокого уровня.

Знать:

- типы данных;
- базовые конструкции изучаемых языков программирования;
- интегрированные среды программирования на изучаемых языках.

3. ТЕМАТИКА, СОДЕРЖАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ

Наименование темы	лабораторная работа обучающихся	Количество часов
Раздел 1. Основные принципы алгоритмизации и программирования		
Тема 1.2 Принципы разработки алгоритмов	1.Разработка линейных алгоритмов и алгоритмов ветвления. 2.Разработка циклических алгоритмов. 3.Разработка алгоритмов шифрования.	6
Раздел 2. Язык программирования		
Тема 2.2 Элементы языка. Простые типы данных	Знакомство с инструментальной средой программирования.	6
Тема 2.3 Базовые конструкции структурного программирования	1.Разработка программ разветвляющейся структуры; 2.Разработка программ с использованием цикла с предусловием; 3.Разработка программ с использованием цикла с постусловием; 4.Разработка программ с использованием цикла с параметром.	20
Тема 2.4 Работа с массивами и указателями. Структурные типы данных	1.Разработка программ с использованием одномерных массивов и указателей; 2.Сортировка одномерных массивов; 3.Разработка программ с использованием двумерных массивов; 4.Разработка программ с использованием структур;	20

	5.Разработка программ с использованием строк.	
Тема 2.5 Процедуры и функции	1.Разработка программ с использованием функций. 2.Разработка программ с использованием рекурсивных функций.	6
Тема 2.6 Работа с файлами	1.Разработка программ работы со структурированными файлами. 2.Разработка программ работы с текстовыми файлами. 3.Разработка программ работы с неструктурированными файлами.	8
Раздел 3. Основы объектно-ориентированного программирования		
Тема 3.1 Класс - как механизм создания объектов	1.Организация классов и принцип инкапсуляции. 2.Разработка приложений с использованием классов.	6
Тема 3.2 Принципы наследования и полиморфизма	1.Программная реализация принципов наследования. 2.Программная реализация принципов полиморфизма.	6
Тема 3.3 Понятия деструктора и конструктора	Разработка конструкторов и деструкторов.	4
Раздел 4. Модульное программирование		
Тема 4.2 Разработка приложений	Разработка многомодульных приложений.	18
Итого		100

Лабораторная работа № 1

Тема:

- 1.Разработка линейных алгоритмов и алгоритмов ветвления;
- 2.Разработка циклических алгоритмов;
- 3.Разработка алгоритмов шифрования;

Цель: Построение алгоритмов линейной структуры и разветвляющейся структуры;

Построение алгоритмов циклической структуры;

Построение алгоритмов шифрования.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций, углубить свои знания по разработке линейных алгоритмов и алгоритмов

ветвления, циклических алгоритмов и алгоритмов шифрования.

Количество часов: 6

Порядок работы:

Задание 1: Составить алгоритм решения задачи. Заданы длины двух катетов в прямоугольном треугольнике. Найти длину гипотенузы, площадь треугольника и величину его углов. Составить схему алгоритма.

Задание 2: Составить алгоритм нахождения суммы положительных чисел из 100 возможных.

Задание 3: Составить алгоритм шифрования текста с помощью шифра Цезаря. Описание шифра: чтобы зашифровать текст, записанный с помощью русских букв и знаков препинания, его можно переписать, заменив каждую букву непосредственно следующей за ней буквой по алфавиту (буква Я заменяется на А). Обобщив этот способ шифровки, можно производить сдвиг не на одну букву, а на N букв (N – натуральное число);

Задание 4. Ответить на теоретические вопросы:

4.1 Понятие алгоритма. 4.2. Свойства алгоритма. 4.3. Способы представления алгоритмов. 4.4. Основные структуры алгоритмов. 4.5. Алгоритмы линейной структуры. 4.6. Алгоритмы разветвляющейся структуры; 4.7. Алгоритмы шифрования; 4.8. Алгоритмы циклической структуры. 4.9 Циклы с предусловием. 4.10. Циклы с постусловием. 4.11. Циклы с параметром. 4.12. Итерационные циклы. 4.13. Вложенные циклы.

Лабораторная работа № 2

Тема: Знакомство с инструментальной средой программирования

Цель: получение навыков работы со средой разработки Visual Studio, создание программ линейной структуры на языке C++ в Visual Studio.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций;

Количество часов: 6

Порядок работы:

Задание 1. Запустите Visual Studio из главного меню путем выполнения следующей последовательности команд: Пуск – Все Программы – Microsoft Visual Studio – Microsoft Visual Studio. Ознакомьтесь со средой разработки Visual Studio.

Ответить на теоретические вопросы:

- 1 Достоинства и недостатки объектно-ориентированного программирования.
2. Основные элементы окна среды Visual Studio.
3. Состав меню среды разработки Visual Studio.
4. Принципы организации справочной системы Visual Studio.
5. Этапы создания приложения.
6. Создание консольного приложения в Visual Studio на языке C++.
7. Понятие линейного алгоритма.
8. Лексемы языка C++.

9. Литералы языка C++.

10. Встроенные типы данных языка C++.

Лабораторная работа № 3

Тема: 1. Разработка программ разветвляющейся структуры;

2. Разработка программ с использованием цикла с предусловием;

3. Разработка программ с использованием цикла с постусловием;

4. Разработка программ с использованием цикла с параметром.

Цель: 1. Получение навыков работы со средой разработки Visual Studio, создание программ разветвляющейся структуры на языке C++ в Visual Studio.

2. Получение навыков работы со средой разработки Visual Studio, создание программ циклической структуры на языке C++ в Visual Studio.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать учебно-методическую литературу и материалы лекций.

Количество часов: 20

Порядок работы:

Задание 1. Напишите код программы, которая перераспределит заданные значения x , y так, что в x окажется большее значение, а в y меньшее:

//перестановка x , y ; большее – вперед #include <iostream> #include <math.h>
using namespace std; int main () { double x, y, z ;

15

cout<<"Введи значение x y "; cin>> x >> y ; if ($x < y$) { $z = x$; $x = y$; $y = z$;} cout<<"\n Итог
 $x =$ << x <<"\n Итог $y =$ << y <<"\n"; return 0;

} Сохраните проект. Откомпилируйте и запустите программу.

Ответить на теоретические вопросы: 1.1 Понятие разветвляющегося алгоритма. 1.2. Условный оператор языка C++. 1.3. Оператор выбора языка C++. 1.4. Тернарный оператор языка C++ 1.5. Конъюнкция, дизъюнкция и инверсия условий, использование в C++.

Задание 2. Среди K первых членов последовательности вида: $1, 1 + 1/2, 1 + 1/2 + 1/3, \dots$ найти первый, больший заданного числа A . Если же его нет (среди первых K членов), то напечатать об этом сообщение. Использовать цикл с предусловием.

Ответить на теоретические вопросы:

Понятие циклического алгоритма.

Оператор цикла с предусловием языка C++.

Задание 3. Рассмотреть указанные последовательности в цикле и выйти из цикла с постусловием, достигнув указанного условия с выдачей порядкового номера члена, при котором достигнуто условие. Если же за m оборотов цикла условие не достигнуто, напечатать об этом сообщение.

Задание 4. Используя цикл с постусловием, написать программу, которая определяет максимальное число из введенной последовательности

положительных чисел (длина последовательности не ограничена). Для завершения ввода нужно нажать 0.

Задание 5. Дано 10 вещественных чисел: a_1, a_2, \dots, a_{10} . Требуется определить, сколько из них принимает значение, большее заданного числа b ; Б) Написать программу, которая выводит таблицу квадратов первых десяти целых положительных чисел. Ответить на теоретические вопросы: Понятие циклического алгоритма.

Оператор цикла с параметром языка C++. Назначение команд команд break и continue.

Лабораторная работа № 4

Тема: 1. Разработка программ с использованием одномерных массивов и указателей;

2. Сортировка одномерных массивов;

3. Разработка программ с использованием двумерных массивов;

4. Разработка программ с использованием структур;

5. Разработка программ с использованием строк.

Цель: 1. Получение навыков работы со средой разработки Visual Studio, создание программ обработки одномерных массивов на языке C++ в Visual Studio, изучение основных принципов работы с указателями.

2. получение навыков работы со средой разработки Visual Studio, создание программ обработки одномерных массивов на языке C++ в Visual Studio, изучение основных методов сортировки массивов.

3. получение навыков работы со средой разработки Visual Studio, создание программ обработки двумерных массивов на языке C++ в Visual Studio.

4. получение навыков работы со средой разработки Visual Studio, создание программ обработки двумерных массивов на языке C++ в Visual Studio, изучение основных методов сортировки массивов.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать учебно-методическую литературу и материалы лекций;

Количество часов: 20

Порядок работы:

Задание 1: Дан массив из N элементов (вещественные числа). Вычислить:

1) сумму отрицательных элементов массива;

2) произведение элементов массива, расположенных между максимальным и минимальным элементами. Упорядочить элементы по возрастанию.

Ответить на теоретические вопросы: Операторы цикла.. Одномерные массивы. Описание массивов в C++. Основные принципы работы с указателями.

Задание 2: Напишите программу, которая сортирует одномерный массив «методом выбора». При сортировке этим методом при просмотре массива ищется наименьший элемент, сравнивая его с первым. Если такой элемент найден, но меняется местами с первым. Затем эти действия повторяются, но не

с первого элемента, а со второго. Так продолжается до тех пор, пока не будет отсортирован весь массив.

Ответить на теоретические вопросы: Одномерные массивы. Описание массивов в C++. Алгоритмы сортировки массивов;

Задание 3. Дана матрица 5×5 . Для данного натурального M найти сумму тех элементов матрицы, сумма индексов которых равна M ;

Ответить на теоретические вопросы Двумерные массивы. Описание массивов в C++.

Задание 4. Составить программу на языке C++, которая выполняет ввод строки с клавиатуры и удаляет пробелы в начале строки.

Задание 5. Составить программу на языке C++, которая выполняет ввод строки с клавиатуры и определяет количество символов * в строке.

Ответить на теоретические вопросы: Строки. Описание символьных переменных и массивов в C++. Стандартные процедуры и функции для работы со строками. Понятие структуры в языке C++. Формат описания структурного типа. Обращение к элементам структуры.

Лабораторная работа № 5

Тема: Разработка программ с использованием функций.

2. Разработка программ с использованием рекурсивных функций.

Цель: Получение навыков работы со средой разработки Visual Studio, создание программ работы с функциями на языке C++ в Visual Studio; создание программ работы с рекурсивными функциями на языке C++ в Visual Studio.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций.

Количество часов: 6

Порядок работы:

Задание 1 Составить программу на языке C++, которая с помощью функции определяет среди чисел $1 + 1/2$, $1 + 1/2 + 1/3$, $1 + 1/2 + 1/3 + 1/4$, ... первое большее a .

Задание 2. Даны числа a, b, c, d . Получить $x = \max(a, b)$, $y = \max(c, d)$, $z = \max(x, y)$. Вычисление $\max(k, m)$ (большого из двух чисел k, m) оформить функцией.

Ответить на теоретические вопросы: Использование функций в программах. Параметры функции. Функция типа `void` и глобальные переменные. Использование инструкции `return`.

Задание 3. Написать программу вычисления факториала числа с помощью рекурсивной функции.

Ответить на теоретические вопросы: Использование рекурсивных функций в программах. Структура рекурсивной функции.

Лабораторная работа № 6

Тема: Разработка программ работы со структурированными файлами.

2. Разработка программ работы с текстовыми файлами

3. Разработка программ работы с неструктурированными файлами

Цель: Получение навыков работы со средой разработки Visual Studio, создание программ для работы с файлами на языке C++ в Visual Studio;

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций,

Количество часов: 8

Порядок работы:

Задание 1. Дана структура с именем AEROFLOT, состоящая из полей:

название пункта назначения рейса;

номер рейса;

тип самолета.

Написать программу, которая выполняет следующие действия:

ввод с клавиатуры данных в массив, состоящий из 7 элементов типа AEROFLOT, и занесение их в файл данных;

чтение данных из файла и вывод их на экран;

вывод на экран пунктов назначения и номеров рейсов,

обслуживаемых самолетом, тип которого введен с клавиатуры (если таких нет – вывести об этом сообщение);

список должен быть упорядочен по алфавиту названий пунктов назначения.

Задание 2. Составить программу на языке C++, которая вычисляет среднее арифметическое чисел, находящихся в текстовом файле test.dat.

Задание 3. Составить программу на языке C++, которая выполняет частотный анализ текста, содержащего заглавные английские буквы, т.е. вычисляет, сколько раз в текстовом файле book.dat встречается заданная буква.

Задание 4 Создать двоичный файл и записать в него целое число n и n вещественных чисел. #include «stdafx.h» #include <iostream> using namespace std; int main() { int n, i; double a; FILE *f; //описываем файловую переменную //создаем двоичный файл в режиме записи f=fopen(«D:\\game\\noobs.dat», «wb»); //ввод числа n cout<<«n=»; cin>>n;

41

fwrite(&n, sizeof(int), 1, f); //цикл для ввода n вещественных чисел for (i=0; i<n; i++) { //ввод очередного вещественного числа cout<<«a=»; cin>>a; //запись вещественного числа в двоичный файл fwrite(&a, sizeof(double), 1, f); } //закрываем файл fclose(f); system(«pause»); return 0;

} Сохраните проект. Откомпилируйте и запустите программу.:

Ответить на теоретические вопросы: Файловый ввод/вывод в языке C. Стандартные процедуры и функции для работы с файлами;

Лабораторная работа № 7

Тема: Организация классов и принцип инкапсуляции.

2.Разработка приложений с использованием классов.

Цель: Получение навыков работы со средой разработки Visual Studio, создание программ с использованием классов на языке C++ в Visual Studio.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций.

Количество часов: 6

Порядок работы:

1. Ответить на теоретические вопросы:

1.1 Понятие класса в языке C++; 1.2. Основное отличие класса от структуры; 1.3. Синтаксис объявления класса; 1.4. Базовые понятия ООП: инкапсуляция, наследование и полиморфизм; 1.5. Открытые и закрытые члены класса; 1.6. Понятие конструктора и деструктора;

Задание 1. Создать класс EngMer для работы с английскими мерами длины: фунтами и дюймами, при этом учтем, что 1 фунт = 12 дюймов. Длина объекта будет задаваться парой чисел (фунты и дюймы), нужно реализовать сложение и вычитание длин, умножение и деление длин, сравнение длин.

Задание 2. Создать класс Date для работы с датами в формате «год. месяц. День». Дата представляется структурой с тремя полями типа unsigned int: для года, месяца и дня. Класс должен включать следующие операции: вычисление даты через заданное количество дней от указанной, вычитание заданного количества дней из даты, определение високосного года, присвоение и получение отдельных частей (год, месяц, день), сравнение дат (равно, до, после), вычисление количества дней между датами.

Задание 3 Создать класс Oct1, который будет содержать число в 8-ричной системе (в отдельном поле – целая часть, в другом поле – дробная часть). Разработайте методы для ввода 8-ричных чисел (с дробной частью), вывода 8-ричных чисел, методы для вычисления суммы и произведения 8-ричных чисел.

Ответить на теоретические вопросы:

Понятие класса в языке C++; Основное отличие класса от структуры; Синтаксис объявления класса; Базовые понятия ООП: инкапсуляция, наследование и полиморфизм; Открытые и закрытые члены класса; Понятие конструктора и деструктора;

Лабораторная работа № 8

Тема: Программная реализация принципов наследования.

2. Программная реализация принципов полиморфизма.

Цель: Получение навыков работы со средой разработки Visual Studio, создание программ с использованием классов на языке C++ в Visual Studio;

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций.

Количество часов: 6

Порядок работы:

Задание 1. Составить программу на языке C++, определяющую общий базовый класс fruit, описывающий некоторые характеристики фруктов. Этот класс наследуется двумя производными классами Apple и Orange. Эти классы содержат специальную информацию о конкретном фрукте (яблоке или апельсине).

Ответить на теоретические вопросы: Понятие класса в языке C++. Основное отличие класса от структуры. Синтаксис объявления класса. Базовые понятия ООП: инкапсуляция, наследование и полиморфизм. Открытые и закрытые члены класса. Понятие конструктора и деструктора

Задание 2. Написать набор классов, представляющий выражения. В этом наборе должен быть один абстрактный базовый тип, а также набор производных от него типов по видам выражений (константа, переменная, сумма, разность, произведение, частное, sin, cos, exp, ln). У каждого из классов должны быть следующие виртуальные функции: напечатать выражение (без параметров), вычислить выражение (параметр – значение переменной, результат – значение выражения), вернуть производную выражения (без параметров), создать копию выражения (тоже без параметров).

Ответить на теоретические вопросы: получение навыков работы со средой разработки Visual Studio; Создание программ с использованием классов на языке C++ в Visual Studio.

Лабораторная работа № 9

Тема: Разработка конструкторов и деструкторов.

Цель: Получение навыков работы со средой разработки Visual Studio, создание программ с использованием классов на языке C++ в Visual Studio

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций,

Количество часов: 4

Порядок работы:

Задание 1. Написать иерархию классов, описывающих имущество налогоплательщиков. Она должна состоять из абстрактного базового класса Property и производных от него классов Apartment, Car и CountryHouse. Базовый класс должен иметь поле worth (стоимость), конструктор с одним параметром, заполняющий это поле, и чисто виртуальный метод расчета налога, переопределенный в каждом из производных классов. Налог на квартиру вычисляется как 1/1000 ее стоимости, на машину – 1/200, на дачу – 1/500. Также каждый производный класс должен иметь конструктор с одним параметром, передающий свой параметр конструктору базового класса

Ответить на теоретические вопросы: Понятие класса в языке C++. Основное отличие класса от структуры. Синтаксис объявления класса. Базовые понятия ООП: инкапсуляция, наследование и полиморфизм. Открытые и закрытые члены класса. Понятие конструктора и деструктора.

Задание 2. Написать набор классов для настольной издательской системы. В этом наборе должен быть абстрактный базовый класс *Box*, представляющий собой прямоугольник с текстом на экране. В этом классе должны быть четыре поля типа *int*: размеры (ширина и высота в символах) и положение левого верхнего угла данного прямоугольника (две целые координаты относительно левого верхнего угла объемлющего прямоугольника). Также в этом классе должны быть следующие методы: *place* с двумя параметрами типа *int*, метод должен записывать их значения в поля, определяющие положение прямоугольника; *width* и *height*, возвращающие ширину и высоту прямоугольника, соответственно, и *layout* и *clone* без параметров, *print* с тремя параметрами (абстрактные методы). Также в этом классе должен быть виртуальный деструктор, переопределенный в каждом из производных классов.

Лабораторная работа № 10

Тема: Разработка многомодульных приложений.

Цель: Получение навыков работы со средой разработки Visual Studio, знакомство с основами модульного программирования на языке C++ в Visual Studio.

Форма самостоятельной деятельности: на этапе подготовки к лабораторной работе студенты должны использовать литературу и материалы лекций,

Количество часов: 18

Порядок работы:

Задание 1: Даны три тройки чисел: *a1, b1, c1; a2, b2, c2; a3, b3, c3*. Найти в каждой из них наибольшее и наименьшее, а затем наибольшее и наименьшее среди наибольших и отдельно – среди наименьших.

```
#include <iostream> #include <iomanip> using namespace std; void maxmin(double
x,double y,double z); double mx, mn;// глобальные переменные (из maxmin) int
main() { double max1,min1,max2,min2,max3,min3,mxmax,mnmax, mxmin,
mnmin, a1,b1,c1, a2,b2,c2, a3,b3,c3
    cout<<"\n введи a1,b1,c1 "; cin>>a1>>b1>>c1;
    cout<<"\n введи a2,b2,c2 "; cin>>a2>>b2>>c2;
    cout<<"\n введи a3,b3,c3 "; cin>>a3>>b3>>c3;
    maxmin(a1,b1,c1); max1=mx; min1=mn;
    cout<<"\n max1,min1"<<setw(5)<<max1<<setw(5) <<min1;
    maxmin(a2,b2,c2); max2=mx; min2=mn; cout<<"\n
    max2,min2"<<setw(5)<<max2<<setw(5) <<min2;
    maxmin(a3,b3,c3);max3=mx; min3=mn;
    cout<<"\n max3,min3"<<setw(5)<<max3<<setw(5) <<min3;
    maxmin(max1,max2,max3); mxmax=mx; mnmax=mn; cout<<"\n
    mxmax,minmax "<<setw(5)<<mxmax<<setw(5) <<mnmax;
    maxmin(min1,min2,min3); mxmin=mx; mnmin=mn;
    cout<<"\n maxmin,minmin "<<setw(5)<<mxmin<<setw(5) <<mnmin;
```

```

cout<<"\n";
return (0);
} void maxmin(double x,double y,double z) {
if ((x>y)&&(x>z)) mx=x;
if ((y>x)&&(y>z)) mx=y;
if ((z>x)&&(z>y)) mx=z;
if ((x<y)&&(x<z)) mn=x;
if ((y<x)&&(y<z)) mn=y;
if ((z<x)&&(z<y)) mn=z; }

```

Сохраните проект. Откомпилируйте и запустите программу.

Ответить на теоретические вопросы: Структура модуля. Раздельная компиляция. Сборка программы. Область действия. Область видимости.

4. КОНТРОЛЬ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ РАБОТ И КРИТЕРИИ ИХ ОЦЕНКИ

Критерии оценивания результатов выполнения практических работ, шкала оценивания

Критерии оценивания:

- умение самостоятельно выполнить работу (произвести расчеты, применить интеллектуальные и исследовательские приемы)
- качество выполнения работы и содержание информационного, расчётного, наглядного материала
- умение излагать программный материал (четкость, грамотность изложения материала, точность и полнота воспроизведения учебного материала).
- соответствие требованиям оформления письменной части

Шкала оценивания:

Результаты оцениваются по шкале «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется обучающемуся, если работа выполнена самостоятельно, произведена самооценка, продемонстрированы навыки самостоятельного использования оборудования, дидактического материала, ТСО; отличается новизной, нестандартным, творческим подходом к теме, решению задачи, оформлению; выполнена своевременно, отличается четким и грамотным выполнением в соответствии с рекомендациями преподавателя.

Оценка «хорошо» выставляется обучающемуся, если выполнение работы, самооценка, навыки самостоятельного использования оборудования, дидактического материала, ТСО происходят с посторонней помощью, исполнение работы частично соответствует рекомендациям преподавателя по оформлению, структуре, аккуратности исполнения, сдана в срок.

Оценка «удовлетворительно» выставляется обучающемуся, если в работе отсутствуют установленные рекомендациями порядок и структура работы, работа выполнена не самостоятельно, сдана с опозданием обозначенного

срока, объем информации незначительный, из ограниченного числа источников

Оценка «неудовлетворительно» выставляется обучающемуся, который не знает значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы.

5.УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

Голицына, О. Л. Основы алгоритмизации и программирования: учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва: ФОРУМ: ИНФРА-М, 2020. — 431 с.

Малашкевич, В.Б. Интернет-программирование: лабораторный практикум / В. Б. Малашкевич ; М-во образования и науки Рос. Федерации, ФГБОУ ВО "Поволж. гос. технол. ун-т". - Йошкар-Ола: ПГТУ, 2017. - 96 с.: ил.

Павловская, Т. А. Программирование на языке высокого уровня: для магистров и бакалавров [Текст]: [учебник : по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника"] / Т. А. Павловская. - Санкт-Петербург [и др.]: [б. и.], 2017. - 460 с.

Семакин И.Г.Основы алгоритмизации и программирования: учеб. для студ. учреждений сред. проф. образования / И. Г. Семакин, А. П. Шестаков. — М.: Издательский центр «Академия», 2018. — 304 с

Дополнительная литература

Городняя, Л.В. Парадигма программирования: учебное пособие / Л.В. Городняя. — Санкт-Петербург: Лань, 2019. — 232 с. — ISBN 978-5-8114-3565-4. — Текст: электронный // Электронно-библиотечная система «Лань»: [сайт]. — Режим доступа: <https://e.lanbook.com/reader/book/118647/#1>

Тюкачев, Н. А. С#. Программирование 2D и 3D векторной графики [Электронный ресурс]: учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. - 2-е изд., испр. и доп. - [Б. м.]: Лань, 2017. - 320 с. – Режим доступа: <https://e.lanbook.com/reader/book/94750/#1>

Маран, М. М. Программная инженерия [Электронный ресурс] : учебное пособие / М. М. Маран. - 1-е изд. - [Б. м.] : Лань, 2018. - 196 с. — Режим доступа: <https://e.lanbook.com/reader/book/106733/#1>