

**федеральное государственное бюджетное образовательное учреждение
высшего образования «Мордовский государственный педагогический
университет имени М.Е. Евсевьева»**

Физико-математический факультет

Кафедра информатики и вычислительной техники

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Объектно-ориентированное программирование**

Направление подготовки: 44.03.05 Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Физика.Информатика

Форма обучения: Очная

Разработчики:

Черемухина Е. В., старший преподаватель кафедры информатики и вычислительной техники

Программа рассмотрена и утверждена на заседании кафедры, протокол № 9 от 19.03.2020 года

Зав. кафедрой _____  _____ Зубрилин А. А.

Программа с обновлениями рассмотрена и утверждена на заседании кафедры, протокол № 1 от 31.08.2020 года

Зав. кафедрой _____  _____ Зубрилин А. А.

1. Цель и задачи изучения дисциплины

Цель изучения дисциплины - формирование научного и практического представления о современных технологиях программирования, о концепциях объектно-ориентированного программирования, овладение навыками решения компетентностно-ориентированных заданий по объектно-ориентированному программированию в инструментальной среде программирования Microsoft Visual Studio на языке программирования C++ для постановки и решения исследовательских задач в предметной области.

Задачи дисциплины:

- сформировать знания о современных технологиях программирования;
- выработать навыки проектирования и решения исследовательских задач в предметной области программирования;
- сформировать знания об объектно-ориентированном программировании, об основных принципах объектно-ориентированного программирования, о концепциях объектно-ориентированного программирования;
- сформировать знания об истории развития языка C++, о возможностях, назначении и особенностях языка программирования C++;
- выработать навыки анализа условия задач по программированию, поиска вариантов решения компетентностно-ориентированных заданий, навыки решения задач по объектно-ориентированному программированию, навыки разработки алгоритмов и программирования различных алгоритмов на языке программирования C++, навыки проведения обоснованного тестирования разработанных приложений на языке C++;
- сформировать представление об особенностях работы в среде программирования Visu Studio с использованием языка программирования C++ для решения задач различного уровня сложности и сформировать точку зрения обучающегося на современную теорию и практику объектно-ориентированного программирования в C++;
- выработать навыки проектирования основных и дополнительных образовательных программ в области объектно-ориентированного программирования в C++.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина К.М.06.25 «Объектно-ориентированное программирование» изучается на 3 курсе, в 5, 6 семестрах.

Для изучения дисциплины требуется: знания, умения, навыки, виды деятельности, полученные и сформированные в рамках школьного курса информатики, а также формируемые при изучении учебных дисциплин «Программирование» и «Теоретические основы информатики».

Изучению дисциплины «Объектно-ориентированное программирование» предшествует освоение дисциплин (практик):

Программирование;

Теоретические основы информатики;

История и методология информатики и вычислительной техники.

Освоение дисциплины «Объектно-ориентированное программирование» является необходимой основой для последующего изучения дисциплин (практик):

Производственная (педагогическая) практика;

Учебная (получение первичных навыков научно-исследовательской работы) практика;

Технология организации внеурочной деятельности по информатике;

Методы решения задач по информатике;

Визуализация решений математических задач;

Разработка дополнительных образовательных программ по информатике;

Технология подготовки учащихся к олимпиадам по информатике.

Область профессиональной деятельности, на которую ориентирует дисциплина «Объектно-ориентированное программирование», включает: 01 Образование и наука (в сфере дошкольного, начального общего, основного общего, среднего общего образования,

профессионального обучения, профессионального образования, дополнительного образования).

Типы задач и задачи профессиональной деятельности, к которым готовится обучающийся, определены учебным планом.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

Компетенция в соответствии ФГОС ВО	
Индикаторы достижения компетенций	Образовательные результаты
<p>ПК-11. Способен использовать теоретические и практические знания для постановки и решения исследовательских задач в предметной области (в соответствии с профилем и уровнем обучения) и в области образования.</p> <p>педагогическая деятельность</p>	
<p>ПК-11.1 Использует теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.</p>	<p>знать:</p> <ul style="list-style-type: none"> - этапы постановки и решения исследовательских задач в области современных технологий программирования; - концепции объектно-ориентированного программирования и ошибки в реализации алгоритма функции анализа возможностей и инструментальной среды программирования Microsoft Visual Studio; - особенности и возможности языка при составлении решения задачи и при его реализации в программной среде; - программирование объектно-ориентированных задач и решения объектно-ориентированному программированию в C++; владеть: <p>навыками решения компетентностно-ориентированных заданий по объектно-ориентированному программированию в C++.</p> <ul style="list-style-type: none"> - навыками анализа исходных данных задачи и оценки получаемого результата, навыками определения типичных ошибок при тестировании листинга программного кода в C++; - навыками учета и оценки возможных типичных ошибок при составлении алгоритма решения задачи и при его реализации в программной среде.

ПК-6. Способен проектировать содержание образовательных программ и их элементов.

проектная деятельность

ПК-6.1 Участвует в проектировании основных и дополнительных образовательных программ.	знать: - основы проектирования основных и дополнительных образовательных программ; - технологии объектно-ориентированного программирования; - возможности языка программирования С++ для реализации алгоритмов решения задач; уметь: - проектировать основные и дополнительные образовательные программы с использованием технологий объектно-ориентированного программирования, возможностей языка программирования С++ для решения компетентностно-ориентированных заданий; владеть: - навыками проектирования основных и дополнительных образовательных программ с использованием технологии объектно-ориентированного программирования, возможностей языка программирования С++ для выполнения компетентностно-ориентированных заданий.
---------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

4. Объем дисциплины и виды учебной работы

Вид учебной работы	Все го часов	Пяты й семес тр	Шест ой семес тр
Контактная работа (всего)	64	30	34
Лекции	14	14	
Лабораторные	50	16	34
Самостоятельная работа (всего)	24	6	18
Виды промежуточной аттестации	20		20
Экзамен	20		20
Общая трудоемкость часы	108	36	72
Общая трудоемкость зачетные единицы	3	1	2

5. Содержание дисциплины

5.1. Содержание разделов дисциплины

Раздел 1. Основы работы в интегрированной среде разработки программ Microsoft Visual Studio на языке программирования С++:

Особенности программной среды Visual Studio. Программирование алгоритмов линейно структуры в С++. Операторы в С++. Операторы цикла в С++. Создание консольных приложений с использованием операторов цикла в С++. Решение задачи табулирования функции на определенном отрезке. Рекуррентные соотношения в С++. Программирование алгоритмов вычисления конечных и бесконечных сумм и произведений в С++. Использование циклов для решения задач численными методами.

Раздел 2. Обработка структур данных в С++:

Одномерные массивы в С++. Составление программ в С++ с использованием массивов. Составление программ в С++ с использованием двумерных массивов. Двумерные массивы в С++. Многомерные массивы в С++. Обработка строк в С++. Обработка строк с использованием функций в языке С++.

Раздел 3. Концептуальные основы и семантика объектно-ориентированного подхода: Типы данных в С++, создаваемые пользователем. Обработки структур данных в С++.

Организация файлового ввода/вывода в C++. Программирование в C++ с использованием функций. Программирование задач с использованием динамических структур данных в C++. Программирование на языке C++ с использованием классов. Форматируемый ввод/вывод. Манипуляторы в C++. Создание пользовательских классов в C++. Перегрузка операторов в C++.

Раздел 4. Объектно-ориентированное программирование на языке C++ :
Программирование с использованием наследования классов в C++. Работа с файловыми потоками в языке C++. Текстовые файлы. Работа с файловыми потоками в языке C++. Двоичные файлы. Шаблоны (параметризованные типы) в C++. Исключения в языке C++. Стандартная библиотека шаблонов в C++. Строковый класс в C++. Разработка приложений Windows Forms с использованием элементов управления.

52. Содержание дисциплины: Лекции (14 ч.)

Раздел 1. Основы работы в интегрированной среде разработки программ Microsoft Visual Studio на языке программирования C++ (8 ч.)

Тема 1. Особенности программной среды Visual Studio (2 ч.)

- 1.1 Работа в интегрированной среде разработки программ MS Visual Studio. Интерфейс особенности работы в среде MS Visual Studio. Основные пункты меню и их назначение.
- 1.2 История развития языков программирования. История развития языка C++, назначение и особенности.
- 1.3 Этапы создания проекта. Запуск программы. Создание консольных приложений на C++. Директивы. Заголовочные файлы.
- 1.4 Описание разделов программы. Синтаксис языка C++. Структура программы на C++.
- 1.5 Тестирование программного кода, анализ ошибок. Устранение ошибок в программе. Основные команды среды. Использование встроенного отладчика среды.
- 1.6 Основные команды среды. Использование встроенного отладчика среды.

Тема 2. Программирование алгоритмов линейной структуры в C++ (2 ч.)

- 2.1 Базовые элементы языка C++. Состав языка. Структура программы.
- 2.2 Идентификаторы языка C++
- 2.3 Стандартные типы данных C++.
- 2.4 Константы.
- 2.5 Переменные.
- 2.6 Организация консольного ввода/вывода данных.
- 2.7 Операции.
- 2.8 Выражения и преобразование типов.
- 2.9 Линейные программы на C++. Программирование алгоритмов линейной структуры в C++.
- 2.10 Работа с типами данных в C++. Ввод и вывод данных в C++.
- 2.11 Вычисление значения выражений с использованием математических функций в C++

Тема 3. Операторы в C++ (2 ч.)

- 3.1 Операторы следования в C++.
- 3.2 Операторы ветвления в C++.
- 3.3 Условный оператор if, составной и вложенный оператор, оператор множественного выбора switch.
- 3.4 Примеры использования операторов ветвления при решении задач в C++.
- 3.5 Программирование алгоритмов с использованием основных алгоритмических структур(следование и ветвление) и их реализация при решении задач в C++

Тема 4. Операторы цикла в C++ (2 ч.)

- 4.1 Операторы цикла:
 - 4.1.1 Оператор цикла for
 - 4.1.2 Оператор цикла do-while
 - 4.1.3 Оператор цикла while
- 4.2 Примеры использования операторов цикла при решении задач
- 4.3 Изменение естественного порядка выполнения программы. Оператор безусловного перехода, оператор break
- 4.4 Решение задач с использованием условного оператора, операторов цикла и операторов управления в C++

4.5 Создание консольных приложений с использованием операторов цикла, операторов передачи управления языка C++

Раздел 2. Обработка структур данных в C++ (6 ч.)

Тема 5. Одномерные массивы в C++ (2 ч.)

5.1 Указатели и ссылки

5.2 Массивы в C++: понятие, объявление и инициализация. Ввод одномерных статических массивов в C++

5.3 Примеры использования одномерных массивов. Составление программ с использованием одномерных массивов

5.4 Использование одномерных массивов при решении задач в C++

Тема 6. Составление программ в C++ с использованием массивов (2 ч.)

6.1 Решение задач с использованием ввода элементов одномерного массива с клавиатуры. Инициализация динамических одномерных массивов в C++

6.2 Ввод элементов статических и динамических массивов в C++ с помощью генератора случайных чисел

6.3 Примеры использования алгоритмов обработки массивов данных при решении задач в C++. Решение задач на обработку одномерных массивов в C++: поиск элемента, сортировка одномерных массивов в C++.

Тема 7. Составление программ в C++ с использованием двумерных массивов (2 ч.)

7.1 Двумерные массивы: объявление в C++. Инициализация матрицы в C++

7.2 Примеры использования двумерных массивов

7.3 Ввод двумерных массивов с клавиатуры

7.4 Ввод двумерных массивов с помощью генератора случайных чисел

7.5 Программирование алгоритмов обработки матриц в C++: работа с матрицей в целом, работа со строкой (столбцом) матрицы, работа с диагональными элементами матрицы в C++

53. Лабораторные (50 ч.)

Раздел 1. Основы работы в интегрированной среде разработки программ Microsoft Visual Studio на языке программирования C++ (8 ч.)

Тема 1. Создание консольных приложений с использованием операторов цикла в C++. Решение задачи табулирования функции на определенном отрезке (2 ч.)

1.1 Вложенные циклы

1.2 Использование организации вложенных циклов при решении задач в C++

1.3 Решение задачи табулирования функции на определенном отрезке.

1.4 программирование алгоритмов с использованием организации вложенных циклов в C++

Тема 2. Рекуррентные соотношения в C++ (2 ч.)

2.1 Вычисление членов рекуррентной последовательности

2.2 Программирование алгоритмов вычисления членов рекуррентной последовательности в C++

Тема 3. Программирование алгоритмов вычисления конечных и бесконечных сумм и произведений в C++ (2 ч.)

3.1 Организация повторений (циклов) в C++

3.2 Вычисление конечных сумм и произведений в C++

3.3 Вычисление бесконечных сумм в C++

3.4 Разработка консольных приложений в C++ с использованием операторов циклов для нахождения конечных и бесконечных сумм, произведений

Тема 4. Использование циклов для решения задач численными методами (2 ч.)

4.1 Программирование алгоритма вычисления интеграла методом прямоугольников в C++

4.2 Программирование алгоритма вычисления интеграла по формуле Симпсона в C++

4.3 Программирование алгоритма вычисления с заданной точностью ϵ корня уравнения $f(x)$ методом простых итераций

4.4 Программирование алгоритма вычисления с заданной точностью ϵ корня уравнения $f(x)$ методом деления отрезка пополам

Раздел 2. Обработка структур данных в C++ (8 ч.)

Тема 5. Двумерные массивы в C++ (2 ч.)

- 5.1 Программирование алгоритмов обработки статических одномерных и двумерных массивов с использованием генерации случайных чисел
- 5.2 Использование генерации псевдослучайных чисел при инициализации одномерных и двумерных статических и динамических массивов
- 5.3 Программирование алгоритмов изменения массивов в C++: вставка и удаление элементов в массивах
- 5.4 Программирование алгоритмов сортировки массивов в C++ с использованием различных методов

Тема 6. Многомерные массивы в C++ (2 ч.)

- 6.1 Инициализация многомерного массива
- 6.2 Программирование алгоритмов обработки многомерных массивов в C++
- 6.3 Контрольная работа по теме "Программирование алгоритмов обработки массивов в C++"

Тема 7. Обработка строк в C++ (2 ч.)

- 7.1 Символьный тип данных. Строка как массив символов
- 7.2 Стандартный класс string языка C++. Строки в C++
- 7.3 Описание и инициализация переменных строкового типа
- 7.4 Инициализация массива строк
- 7.5 Ввод и вывод строки. Доступ к компонентам строки
- 7.6 Программирование алгоритмов обработки строк в C++
- 7.7 Работа с символьными массивами на языке C++ в среде Microsoft Visual Studio

Тема 8. Обработка строк с использованием функций в языке C++ (2 ч.)

- 8.1 Функции для работы со строками в C++. Функции проверки символов в C++
- 8.2 Выделение слов из строки. Формирование слов из строки с использованием функций в C++
- 8.3 Программирование алгоритмов обработки строк в C++
- 8.4 Решение задач на обработку символьных данных с использованием функций, содержащихся в <string.h>

Раздел 3. Концептуальные основы и семантика объектно-ориентированного подхода (18 ч.)

Тема 9. Типы данных в C++, создаваемые пользователем (2 ч.)

- 1.1 Определение структурного типа. Структурный тип данных на языке C++
- 1.2 Инициализация переменной структурного типа
- 1.3 Доступ к значениям полей структурного типа
- 1.4 Вложенные структурные типы
- 1.5 Массивы элементов структурного типа

Тема 10. Обработки структур данных в C++ (2 ч.)

- 2.1 Типы данных "структуры" в C++
- 2.2 Тип данных "объединение" в C++
- 2.3 Тип данных "перечисления" в C++
- 2.4 Программирование алгоритмов обработки структур данных в C++

Тема 11. Организация файлового ввода/вывода в C++ (2 ч.)

- 3.1 Файловый ввод/вывод
- 3.2 Текстовые файлы. Файловая система
- 3.3 Основные методы обработки текстовых файлов в C++. Построчное чтение информации из входного файла и вывод ее на экран
- 3.4 Двоичные файлы. Последовательный доступ к элементам двоичных файлов
- 3.5 Составление в C++ программы записи заданного количества целых, случайно выбранных чисел, в двоичный файл
- 3.6 Организация произвольного доступа к элементам двоичных файлов
- 3.7 Программирование алгоритмов с организацией файлового ввода/вывода в C++

Тема 12. Программирование в C++ с использованием функций (2 ч.)

- 4.1 Понятие функции. Стандартные функции. Программируемые пользователем функции в C++

4.2 Прототип - описание функции в C++. Особенности синтаксиса для прототипа функций с параметрами

4.3 Параметры функции. Примеры использования параметров функции в C++

4.4 Возвращение значений функции с помощью оператора `return`

4.5 Передача массивов в качестве аргументов функции

4.6 Понятие об указателях

4.7 Понятие ссылки

4.8 Решение задач по теме "Программирование в C++ с использованием функций"

Тема 13. Программирование задач с использованием динамических структур данных в C++ (2 ч.)

5.1 Понятие о самоссылочных структурах

5.2 Формирование очереди в C++. Составление программы формирования очереди из 10 элементов в C++ и вывода ее на экран

5.3 Формирование стека. Составление программы формирования стека из 10 элементов в C++ и вывода его на экран

5.4 Добавление и удаление элементов в односвязных списках

5.5 Программирование алгоритмов формирования в C++ динамического списка (стек или очередь) с учетом заданной длины списка (определенного количества элементов)

5.6 Решение задач с использованием динамических структур данных в C++

Тема 14. Программирование на языке C++ с использованием классов (2 ч.)

6.1 Определение класса

6.2 Управление доступом: метка-спецификатор `public` (открытый), метка-спецификатор `protected` (защищенный), метка-спецификатор `private` (закрытый)

6.3 Элементы класса. Функция-элемент класса. Доступ к данным-элементам класса

6.4 Вызов функций-элементов класса

6.5 Указатель `this`

6.6 Конструктор. Деструктор

6.7 Программирование алгоритмов с использованием классов в C++

Тема 15. Форматируемый ввод/вывод. Манипуляторы в C++ (2 ч.)

7.1 Специальные функции форматирования - манипуляторы ввода/вывода в C++. Заголовочный файл `<iomanip.h>`

7.2 Использование специальных функций форматирования при организации ввода/вывода в C++

7.3 Программирование алгоритмов решения задач с организацией вывода данных с использованием манипуляторов в C++

Тема 16. Создание пользовательских классов в C++ (2 ч.)

8.1 Обзор стилей программирования. Процедурное, структурное, функциональное, логическое и объектно-ориентированное программирование

8.2 Основные принципы объектно-ориентированного программирования

8.3 Элементы объектно-ориентированного программирования на языке C++

8.4 Создание пользовательских классов в C++. Вызов функций-членов классов из основного приложения

8.5 Объявление классов и объектов. Создание экземпляров класса в C++. Конструкторы и деструкторы

8.6 Область видимости компонент класса. Определение компонентных функций класса. Статические компоненты классов

8.7 Свойства класса: проверка области допустимых значений, вычисляемое свойство

8.8 Программирование классами в C++

Тема 17. Перегрузка операторов в C++ (2 ч.)

9.1 Дружественные классы

9.2 Дружественные функции

9.3 Перегрузка операций в C++

9.4 Правила, действующие на функции-операции и на перегрузку операций

9.5 Программирование алгоритмов решения задач с использованием перегрузки операторов в C++

Раздел 4. Объектно-ориентированное программирование на языке C++ (16 ч.)

Тема 18. Программирование с использованием наследования классов в C++ (2 ч.)

- 1.1 Доступ к элементам класса
- 1.2 Спецификаторы доступа к элементам класса при наследовании в C++
- 1.3 Конструкторы и деструкторы при наследовании
- 1.4 Виртуальные функции
- 1.5 Программирование алгоритма решения задач с наследованием класса стек от класса массив в C++

Тема 19. Работа с файловыми потоками в языке C++. Текстовые файлы (2 ч.)

- 2.1 Поточковый ввод/вывод дисковых файлов
- 2.2 Режимы доступа к элементам файлов
- 2.3 Текстовые файлы: создание и запись
- 2.4 Посимвольное чтение текста
- 2.5 Посимвольная запись текста
- 2.6 Построчное чтение файла
- 2.7 Построчная запись текста
- 2.8 Признак конца файла в файлах, открытых для чтения
- 2.9 Программирование алгоритмов работы с файловыми потоками в языке C++

Тема 20. Работа с файловыми потоками в языке C++. Двоичные файлы (2 ч.)

- 3.1 Сохранение данных в двоичных файлах. Сохранение в двоичных файлах данных стандартных типов
- 3.2 Алгоритм записи значения типа double в двоичный файл. Чтение значений типа double двоичного файла
- 3.3 Сохранение в двоичных файлах данных, имеющих тип, создаваемый пользователем в C++
- 3.4 Произвольный доступ к элементам файлов. Файловый указатель. Организация доступа к элементам двоичных файлов
- 3.5 Алгоритм замены в двоичном файле, содержащем целые числа, максимального значения файла суммой его четных элементов
- 3.6 Решение задач по теме "Работа с файловыми потоками в языке C++. Двоичные файлы"

Тема 21. Шаблоны (параметризованные типы) в C++ (2 ч.)

- 4.1 Шаблоны функций
- 4.2 Шаблоны классов. Синтаксис шаблона класса
- 4.3 Алгоритм создания шаблона класса, реализующего односвязный список в C++
- 4.4 Программирование алгоритмов решения задач с использованием шаблонов функций и шаблонов классов в C++

Тема 22. Исключения в языке C++ (2 ч.)

- 5.1 Механизм обработки исключительных ситуаций в языке C++
- 5.2 Синтаксис блока обработки исключений с использованием ключевых слов try, catch throw
- 5.3 Создание стека-массива, использующего исключения для отслеживания его верхней и нижней границы
- 5.4 Разработка приложений с использованием исключений в C++

Тема 23. Стандартная библиотека шаблонов в C++ (2 ч.)

- 6.1 Библиотека стандартных шаблонов (Standard Template Library, STL) как инструмент языка программирования C++. Основные определения STL
- 6.2 Основные элементы ядра библиотеки стандартных шаблонов: контейнеры, итераторы, алгоритмы, функциональные объекты
- 6.3 Контейнеры, определенные в STL. Методы для всех контейнеров
- 6.4 Алгоритмы предназначены для обработки контейнеров. Основные алгоритмы STL
- 6.5 Векторы. Класс vector. Функции-элементы класса vector. Основные операции производимые над векторами в C++. Пример использования векторов в C++ для хранения объектов
- 6.6 Списки. Класс list. Функции-элементы класса list. Пример сортировки списка в C++
- 6.7 Программирование алгоритмов решения задач с использованием стандартной библиотеки шаблонов в C++

Тема 24. Строковый класс в C++ (2 ч.)

- 7.1 Заголовочный файл < string>. Работа со строковым классом в C++
- 7.2 Поиск в строке. Реализация в C++ методов поиска в строке
- Модификация строки. Реализация в C++ методов модификации строк
- 7.3 Сравнение строк. Сравнение строк с помощью перегруженных операций
- 7.4 Программирование в C++ алгоритмов с использованием STL

Тема 25. Разработка приложений Windows Forms с использованием элементов управления (2 ч.)

- 8.1 Создание приложений с помощью Windows Forms Application. Элементы управления Windows Forms
- 8.2 Создание элементов управления с помощью визуальных средств. Элемент управления Label
- 8.3 Программное создание элементов управления Button, Panel, TextBox
- 8.4 Программное создание элементов управления MaskedTextBox, CheckBox, NumericUpDown, LinkLabel
- 8.5 Программное создание элементов управления: коллекции ComboBox, ListBox, CheckedListBox, ImageList
- 8.6 Программное создание элементов управления: коллекции TreeView, ListView, DataGridView
- 8.7 Разработка приложений с использованием элементов управления на форме. Обработчик щелчка левой кнопкой мыши на форме, пример обработки событий OnMouseDown
- 8.8 Графические возможности среды Visual Studio. Рисование по пикселям. Пример рисования функции $y = \cos(x)$ на отрезке $x \in [0, 2\pi]$. Работа с графическими примитивами, выбор цвета контура и цвета заливки
- 8.9 Рисование пером. Пример выполнения рисунка с использованием пера
- 8.10 Программирование графики в Visual Studio

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (разделу)

6.1 Вопросы и задания для самостоятельной работы

Пятый семестр (6 ч.)

Раздел 1. Основы работы в интегрированной среде разработки программ Microsoft Visual Studio на языке программирования C++ (3 ч.)

Вид СРС: *Подготовка к практическим / лабораторным занятиям

Перечень контрольных вопросов по модулю 1 "Основы работы в интегрированной среде разработки программ Microsoft Visual Studio на языке программирования C++".

1. Перечислите этапы подготовки к выполнению программы на языке C++?
2. Перечислите основные пункты меню Microsoft Visual Studio и их назначение?
3. Какова структура программы на языке C++?
4. Для чего необходима директива препроцессору # include?
5. Каким образом можно осуществить вывод информации на экран?
6. Как можно организовать ввод данных с клавиатуры?
7. Что называют идентификатором?
8. Какие операторы позволяют организовать выбор между несколькими вариантами?
9. Что понимают под логическим выражением?
10. Чем отличается логическое И от логического ИЛИ?
11. Какие операции отношения вы знаете?
12. В чем особенность использования операторов if и switch?
13. Каким образом цикл while может имитировать цикл for?
14. Каким образом цикл while может имитировать цикл do-while?
15. В каких случаях используются операторы break, continue, exit?
16. Почему в языке C++ нет необходимости использовать оператор goto?
17. Каким образом цикл while может имитировать цикл for?

18. Каким образом цикл while может имитировать цикл do-while?
19. В каких случаях используются операторы break, continue, exit?
20. Почему в языке C++ нет необходимости использовать оператор goto?

Задания для самостоятельного выполнения.

Задание 1. Выполните анализ предложенного программного кода, определите допущенные ошибки, поясните их и исправьте найденные ошибки.

Вариант 1

Исправить ошибки:

```
includ <iostream.h>
main {}
( cout << "Problems, problems \n";
cout << Problems all day long! \n;
cout << "The Everly Brothers \n")
```

Вариант 2

```
include <io stream.h>
main {} // эта программа печатает число недель в году
( int s
S := 56;
cin << В году S недель;
```

Вариант 3

```
# include <iostream.h>
main()
( cout (" Здравствуйте \n ");
cout ( ' Я – программа на C' \n );
cout (" которая причинит кучу неприятностей \n ?")
)
```

Вариант 4

Исправить ошибки:

```
# include <iostream.c>
main()
Begin
int S;
S :=56;
print ( ' В году' , S, 'неделя' ) End
```

Вариант 5

Исправить ошибки:

```
# include <ioctrim.c>;
main()
{ int include, j, k
SUM = include + j;
cout ('С умма будет равна % d, sum!);
cout ( ' Ну что? Опять ошибки?')}
```

Вариант 6

Исправить ошибки:

```
include <iostream.h>
main()
( cout (« Программирование – это сказка: / n»);
cout (« чем дальше, тем страшнее! \n);
cout (« Станем сказочными героями ! \n»)
)
```

Вариант 7

Исправить ошибки:

```
include <include.h>
main(); // эта программа печатает число месяцев
// в году
( int m;
m := 12;
cout << « В году m месяцев» )
```

Вариант 8

Исправить ошибки:

```
# include <iostream.h>
main()
( cout << ' я изучаю, \ у ';
cout << ' язык программирования
// C++.'\7;
cout << ' Объектно-ориентированное программирование\ n'
)
```

Вариант 9

Исправить ошибки:

```
# include <stream.c>
main();
Begin
int d;
D :=29;
cin >> « В январе»' , d, «недель»
End
```

Вариант 10

Исправить ошибки:

```
# include <iostream.c>;
main()
{ int I, j, k
SUM = I + j;
cout ('С умма будет равна % d, sum!);
cout (' Ну что? Опять ошибки?') }
```

Вариант 11

Исправить ошибки:

```
#incluye <stream.h>
main;
{cin ("Visual Studio.\N");
cin ("Язык программирования C++" \ N);
cin (" Объектно-ориентированное программирование\ N") }
```

Вариант 12

Исправить ошибки:

```
$includ <iostream.h>
main()
int min
min= 60;
cout (В часу &min минут);
```

Вариант 13

Исправить ошибки:

```
# include <iostream.cpp>
MAIN();
Begin
cout << (" Язык программирования C+
+, \ff ");
cout << ( " Среда программирования
Visual Studio" \l); End
```

Вариант 14

Исправить ошибки:

```
# includ <iostream.c>

main()
Begin int S;
scanf (S);
print ( ' В году' , % S, 'недель')
End
```

Вариант 15

Исправить ошибки:

```
# include <studio.c>;
main();
{ int i, j, MUL
MUL = i * j;
printf ( ' Произведение равно &MUL')
printf ( ' Оно действительно такое?') }
```

Вид СРС: *Работа с электронными ресурсами и информационными системами

Пройти дистанционное обучение по указанному курсу.

Курс «Алгоритмизация. Введение в язык программирования C++».

Режим доступа:

<https://www.intuit.ru/studies/courses/16740/1301/info>

Информация о курсе «Алгоритмизация. Введение в язык программирования C++». Материал, представленный в данном курсе «Алгоритмизация. Введение в язык программирования C++», позволит обучающимся рассмотреть понятие алгоритма, его виды. В курсе изложены принципы использования и реализации разных видов алгоритмов графически и программно на языке программирования C++. Данный курс является необходимой базой для дальнейшего изучения языка C++. В курсе рассматриваются особенности программы Borland C++ Microsoft Visual Studio, описаны принципы работы с этими программами, каждая задача рассмотрена в обеих средах, даются подробные комментарии и результаты выполнения программ. После каждой лекции представлены задания для самостоятельного выполнения.

Вид СРС: *Подготовка к контрольной работе

Разработайте программу на языке программирования C++.

1. По номеру y ($y > 0$) некоторого года определить s – номер его столетия. Учесть, например, что началом XXI столетия был 2001, а не 2000 год. Задачу решить двумя способами: с использованием операторов принятия решения и без использования.
2. Даны произвольные числа a , b , c . Если нельзя построить треугольник с такими длинами сторон, то выдать соответствующее сообщение; если можно, то напечатать какой он: равносторонний, равнобедренный, разносторонний и вычислить его площадь по формуле Герона.
3. Составить программу, которая по введенному году и номеру месяца определяет число дней в этом месяце.
4. По введенному времени и известному расписанию занятий вывести сообщение о том, что это: пара (с указанием ее номера) или перемена.

5. Элементы окружности пронумерованы таким образом: 1 – радиус (R), 2 – диаметр (D), 3 – длина (L), 4 – площадь круга (S). По номеру элемента и его значению вычислить значения остальных элементов.

Раздел 2. Обработка структур данных в C++ (3 ч.)

Вид СРС: *Подготовка к практическим / лабораторным занятиям

Вид СРС: *Работа с электронными ресурсами и информационными системами

Пройти дистанционное обучение по указанному курсу.

Курс «Язык программирования C++».

Режим доступа:

<https://www.intuit.ru/studies/courses/17/17/info>

Информация о курсе «Язык программирования C++».

В систематизированном виде излагаются основные понятия и описываются возможности языка C++, при этом основное внимание уделяется объяснению того, как теми или иными возможностями пользоваться. Язык программирования C++ – это универсальный язык программирования, который позволяет разрабатывать программы в соответствии с разными парадигмами: процедурным программированием, объектно-ориентированным, параметрическим. В данном курсе рассматриваются все основные возможности языка C++ и их применение при разработке объектно-ориентированных программ. Дается краткое описание библиотек языка C++, необходимых для создания типичных программ.

Вид СРС: *Подготовка к контрольной работе

1. Дана целочисленная матрица 7×7 . Найти сумму положительных элементов, стоящих над главной диагональю и количество нечетных элементов, стоящих под побочной диагональю. Заменить полученными значениями соответственно элементы четных и нечетных столбцов
2. Дана действительная матрица 6×8 . Найти сумму элементов каждой строки верхней половины матрицы и произведение элементов каждой строки ее нижней половины. Определить значение и местоположение максимального элемента верхней половины матрицы и минимального элемента ее нижней половины
3. Дана вещественная квадратная матрица 5×5 . Получить новую матрицу путем прибавления к элементам каждой строки матрицы наименьшего значения элементов этой строки. Сформировать одномерный массив, содержащий количество четных элементов в четных столбцах полученной матрицы и одномерный массив, содержащий количество нечетных элементов в нечетных столбцах.

Шестой семестр (18 ч.)

Раздел 3. Концептуальные основы и семантика объектно-ориентированного подхода (9 ч.)

Вид СРС: *Подготовка к практическим / лабораторным занятиям

Перечень контрольных вопросов по модулю 3 "Концептуальные основы и семантика объектно-ориентированного подхода".

1. Каков допустимый уровень вложенности функции в языке C++?
2. Как определить функцию? Что такое прототип функции? Всегда ли обязательно объявление прототипов?
3. Как передать информацию функции?
4. В чем разница между формальными и фактическими аргументами? Где описываются аргументы?
5. Где описываются локальные переменные функции?
6. Для чего служит оператор return? Обязательно ли его использование?
7. Проверьте, все ли правильно в следующем определении функции:

```
hallo (num)
```

```
{ int num, count;
```

```
for (count =1; count <= num; num++)
```

```
cout<<"Hello, my friend!"<<endl;
```

}

8. Какие спецификации доступа используются при описании класса?
9. Что является элементами класса?
10. Как осуществляется доступ к элементам класса?
11. Для чего используется указатель `this`?
12. Что такое конструктор?
13. Что такое деструктор?
14. Какие классы и функции называются дружественными?
15. Как осуществляется перегрузка операций?
16. Сколько аргументов требуется для определения перегруженной унарной (бинарной) операции?
17. Чем отличается действие перегруженной операции «`++`» при ее использовании в префиксной форме от использовании в постфиксной форме?
18. Какой класс называется базовым?
19. Какой класс называется производным?
20. Какие ключи доступа используются при наследовании?
21. Наследуются ли конструкторы?
22. Наследуются ли деструкторы?
23. Что собой представляет виртуальная функция?
24. Можно ли виртуальную функцию объявить как `static`?

Вид СРС: *Работа с электронными ресурсами и информационными системами
Пройти дистанционное обучение по одному из указанных в перечне курсов.

1. Курс «Язык программирования C++ для профессионалов». Режим доступа:

<https://www.intuit.ru/studies/courses/98/98/info>

Информация о курсе «Язык программирования C++ для профессионалов».

В курсе представлено описание языка C++, его ключевых понятий и основных приемов программирования на нем, содержится описание всех средств C++, в том числе управление исключительными ситуациями, шаблоны типа (параметризованные типы данных) и множественное наследование. C++ является языком программирования общего назначения. Естественная для него область применения – системное программирование, понимаемое в широком смысле этого слова. Кроме того, C++ успешно используется во многих областях приложения, далеко выходящих за указанные рамки. Реализации C++ теперь есть на всех машинах, начиная с самых скромных микрокомпьютеров до самых больших супер-ЭВМ, и практически для всех операционных систем. Поэтому представлен материал, связанный с описанием собственно языка без объяснения особенности конкретных реализаций, среды программирования или библиотек.

2. Курс «Основы объектно-ориентированного программирования».

Режим доступа:

<https://www.intuit.ru/studies/courses/71/71/info>

Информация о курсе «Основы объектно-ориентированного программирования».

В курсе представлен материал по основам объектно-ориентированного программирования и инженерии программ, подробно излагаются основные понятия объектной технологии – классы, объекты, управление памятью, типизация, наследование, универсализация. Большое внимание уделяется проектированию по контракту и обработке исключений, как механизмам, обеспечивающим корректность и устойчивость программных систем.

Изложение материала по основам объектно-ориентированного программирования начинается с рассмотрения критериев качества программных систем и обоснования того, как объектная технология разработки может обеспечить требуемое качество. Основные понятия объектной технологии и соответствующая нотация появляются как результат тщательного анализа и обсуждений. Подробно рассматривается понятие класса – центральное понятие объектной технологии. Рассматривается абстрактный тип данных, лежащий в основе класса, совмещение классом роли типа данных и модуля и другие аспекты построения класса. Подробно рассматриваются объекты и проблемы управления памятью. В курсе рассматриваются

отношения между классами – наследование, универсализация и их роли в построении программных систем, представлено описание технологии проектирования по контракту, как механизма, обеспечивающего корректность создаваемых программ. Рассматриваются такие важные темы объектного программирования как скрытие информации, статическая типизация, динамическое связывание и обработка исключений.

Вид СРС: *Подготовка к контрольной работе

1. Создать класс «Мобильный телефон», включающий данные-элементы: номер, имя владельца, количество денег на счете.

Функции-элементы: создание и инициализация (конструктор), пополнение счета (сумма – в аргументе), оплата разговоров (тариф и время – в аргументе), выдача сообщения об остатке средств на счете, деструктор.

Раздел 4. Объектно-ориентированное программирование на языке C++ (9 ч.)

Вид СРС: *Подготовка к практическим / лабораторным занятиям

Перечень контрольных вопросов по модулю 4 "Объектно-ориентированное программирование на языке C++".

1. Какую цель преследует использование в программе обработки исключений?
2. Как оформляется блок обработки исключений?
3. Что такое обработчики исключений?
4. Продемонстрируйте разработанное приложение с использованием обработки исключений в C++.
5. Расскажите о реализации графических возможностей в Visual Studio.
6. Поясните основные свойства и назначение управляющего элемента ComboBox, ListBox, Timer, MenuStrip.
7. Опишите назначение и основные свойства элементов управления в Visual Studio.
8. Что такое визуальные и не визуальные компоненты формы?
9. Каким образом можно задать свойства компонента?
10. Каким образом элемент формы настроить на выполнение определенного действия?
11. Каким образом можно отформатировать текст компонента Label?
12. Какие существуют способы рисования графиков? Поясните их особенность.

Вид СРС: *Работа с электронными ресурсами и информационными системами

Пройти дистанционное обучение по указанному курсу.

Курс «Основы объектно-ориентированного проектирования».

Режим доступа:

<https://www.intuit.ru/studies/courses/72/72/info>

Информация о курсе «Основы объектно-ориентированного проектирования».

В курсе подробно рассматривается объектная технология бесшовной разработки программных систем, включающая этапы анализа, проектирования, разработки и сопровождения. Рассматриваются перечень таких тем, как способы описания классов, правильное использование наследования, таксономия наследования, объектно-ориентированный анализ и др. Курс посвящен рассмотрению процесса разработки программных систем, когда объектная технология применяется на самых ранних этапах разработки – анализа и проектирования. Рассмотрение начинается с двух важных образцов проектирования. На этих примерах демонстрируются преимущества объектной технологии. Далее идет систематическое изложение основ объектного анализа и проектирования. Подробно обсуждаются вопросы поиска нужных абстракций данных, правильное применение наследования, как важнейшего механизма проектирования систем, роль абстрактных классов. Представлен материал, посвященный принципам проектирования классов и объектно-ориентированному анализу, подробно обсуждаются и более сложные механизмы – параллельности и распределенных вычислений. Эти темы начинают играть все более важную роль в современных разработках.

Специальный интерес могут представлять темы, посвященные проблемам обучения и сравнительному анализу языков программирования.

Вид СРС: *Подготовка к контрольной работе

1. Опишите функцию вычисления тангенса, обработайте ошибку вычисления арктангенса при аргументе, косинус которого равен 0.
2. Опишите функцию вычисления арктангенса, обработайте ошибку вычисления арктангенса при аргументе, равном 1 и -1 .

7. Тематика курсовых работ(проектов)

Курсовые работы (проекты) по дисциплине не предусмотрены.

8. Оценочные средства

8.1. Компетенции и этапы формирования

№ п/п	Оценочные средства	Компетенции, этапы их формирования
1	Предметно-методический модуль	ПК-6, ПК-11.
2	Предметно-технологический модуль	ПК-6.
3	Учебно-исследовательский модуль	ПК-11.

8.2 Показатели и критерии оценивания компетенций, шкалы оценивания

Шкала, критерии оценивания и уровень сформированности компетенции			
2 (не зачтено) ниже порогового	3 (зачтено) пороговый	4 (зачтено) базовый	5 (зачтено) повышенный
ПК-11 Способен использовать теоретические и практические знания для постановки и решения исследовательских задач в предметной области (в соответствии с профилем и уровнем обучения) и в области образования			
ПК-11.1 Использует теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.			
Не способен использовать теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.	В целом успешно, но бессистемно Использует теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.	В целом успешно, но с отдельными недочетами Использует теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.	Способен в полном объеме использовать теоретические и практические знания для постановки и решения исследовательских задач в предметной области в соответствии с профилем и уровнем обучения и в области образования.
ПК-11.2 Проектирует и решает исследовательские задачи в предметной области в соответствии с профилем и уровнем обучения и в области образования.			
Не способен проектировать и решать	В целом успешно, но бессистемно проектирует и	В целом успешно, но с отдельными недочетами	Способен в полном объеме проектировать и решать

исследовательские задачи в предметной области в соответствии с профилем и уровнем обучения и в области образования.	решает исследовательские задачи в предметной области в соответствии с профилем и уровнем обучения и в области образования.	проектирует и решает исследовательские задачи в предметной области в соответствии с профилем и уровнем обучения и в области образования.	исследовательские задачи в предметной области в соответствии с профилем и уровнем обучения и в области образования.
ПК-6 Способен проектировать содержание образовательных программ и их элементов			
ПК-6.1 Участвует в проектировании основных и дополнительных образовательных программ.			
Не способен участвовать в проектировании основных и дополнительных образовательных программ.	В целом успешно, но бессистемно участвует в проектировании основных и дополнительных образовательных программ.	В целом успешно, но с отдельными недочетами участвует в проектировании основных и дополнительных образовательных программ.	Способен в полном объеме участвовать в проектировании основных и дополнительных образовательных программ.

Уровень сформированности компетенции	Шкала оценивания для промежуточной аттестации	Шкала оценивания по БРС
	Экзамен (дифференцированный зачет)	
Повышенный	5 (отлично)	90 – 100%
Базовый	4 (хорошо)	76 – 89%
Пороговый	3 (удовлетворительно)	60 – 75%
Ниже порогового	2 (неудовлетворительно)	Ниже 60%

8.3. Вопросы промежуточной аттестации

Шестой семестр (Экзамен, ПК-11.1, ПК-11.2, ПК-6.1)

- Выделите и раскройте принципы объектно-ориентированного программирования.
- Раскройте особенности объектно-ориентированного программирования.
- Расскажите об истории программирования и поколениях языков программирования, об истории стилей программирования и этапах развития технологии программирования, об объектно-ориентированной парадигме программирования. Опишите сущность объектно-ориентированного подхода к программированию. Расскажите об основных понятиях объектно-ориентированного программирования. Опишите идею и цель объектно-ориентированного программирования.
- Расскажите о понятии инструментального программного обеспечения, языка и системы программирования. Укажите, что является составными частями языка программирования? Перечислите компоненты системы программирования. Расскажите об истории создания и развития языков C/C++, о современном состоянии и сферах применения языка C/C++.
- Расскажите о структуре программы на языке C++. Приведите пример простейшей программы на языке C++. Опишите назначение элементов программы. Какие разделы являются обязательными? Объясните назначение функции `main()`. Расскажите о директива препроцессора. Поясните, какая команда компилятора C++ требуется для подключения заголовочных файлов? Расскажите об использовании библиотеки математических функций. Приведите примеры.

6. Расскажите о базовых элементах языка C++. Поясните состав языка, расскажите об алфавите языка, о лексемах языка (идентификаторы, ключевые (зарезервированные) слова, знаки операций, константы, разделители).
7. Расскажите о стандартных типах данных C++. Расскажите об особенностях представления данных. Опишите простые и составные типы данных. Опишите простые типы данных и приведите примеры операций над числовыми данными. Перечислите простейшие типы данных, которые определяет язык C++, и приведите типичные диапазоны их значений.
8. Представьте определение переменных. Расскажите о правилах именования переменных и функций языка C++, правилах записи констант. Расскажите о понятии ключевого или зарезервированного слова, представьте список ключевых слов языка C++. Расскажите о глобальных переменных, об области видимости переменных в программах на языке C++, приведите примеры.
9. Расскажите об использовании комментариев в языке C++ и escape-последовательностей. Поясните синтаксис комментариев. Расскажите об операторах управления в языке C++ приведите примеры их использования.
10. Расскажите об организации консольного ввода/вывода в языке C++. Расскажите о потоках и об операциях << и >> для потоков. Приведите примеры использования манипуляторов и расскажите о возможности форматирования ввода/вывода. Продемонстрируйте на примере использование манипулятора setw. Расскажите о заголовочном файле iomanip.
11. Расскажите о правилах формирования и вычисления выражений в языке C++. Расскажите об операциях языка C++ (арифметические, логические, условные операции, операции сравнения, операции присваивания) и о приоритетах операций в C++.
12. Расскажите о псевдослучайных числах и генерации псевдослучайных чисел на языке C++. Приведите примеры. Представьте листинг программного кода на языке C++ для генерации случайных чисел в диапазоне от [a,b] и вывода этой последовательности из m чисел.
13. Раскройте понятия объявления и инициализации переменной. Расскажите о целочисленных, строковых, символьных константах и константных переменных. Приведите примеры использования констант в языке C++. Расскажите о директиве # define и префикс const. Поясните существующие отличия в объявлении констант.
14. Расскажите о понятии оператора, об операторах следования (оператор-выражение и составной оператор).
15. Расскажите о понятии оператора, об операторах ветвления в языке C++. Приведите примеры использования условного оператора if и оператора выбора switch в C++.
16. Расскажите об использовании оператора множественного выбора switch в языке C++. Приведите примеры. Представьте сравнительную характеристику использования условного оператора if...else и оператора множественного выбора switch. Поясните назначение оператора break и назначение секции default.
17. Расскажите об организации цикла с параметром, об организации цикла с предусловием и об организации цикла с постусловием в языке C++. Представьте фрагменты программного кода и поясните синтаксис при использовании операторов для организации циклов в языке C++. Приведите примеры.
18. Расскажите об организации цикла с параметром в языке C++. Поясните структуру цикла с параметром и его синтаксис. Расскажите об организации вложенных циклов. Представьте фрагмент программного кода и поясните структуру вложенных циклов на примере цикла for. Приведите примеры.
19. Расскажите о понятии оператора, об операторах безусловного перехода в языке C++. Приведите примеры использования операторов безусловного перехода и представьте фрагменты программного кода.
20. Расскажите о вычислении конечных и бесконечных сумм и произведений в языке C++. Приведите примеры. Представьте фрагменты программного кода вычисления конечных и бесконечных сумм и произведений на языке C++.
21. Расскажите о понятии массива, об индексации и размерности массива. Опишите способы объявления и инициализации массивов языке C++. Приведите примеры.

22. Охарактеризуйте тип данных «массив»: особенности объявления, множества значений, допустимые операции, способ доступа к элементам. Приведите примеры объявления одномерного массива для различных типов значений элементов. Расскажите об одномерных массивах в языке C++. Представьте алгоритм поиска наибольшего и наименьшего значений элементов массива. Поясните отличия динамических одномерных массивов от статических. Приведите примеры использования одномерных массивов.
23. Охарактеризуйте тип данных «массив»: особенности объявления, множества значений, допустимые операции, способ доступа к элементам. Приведите пример объявления двумерного массива для различных типов значений элементов. Расскажите об двумерных массивах в языке C++. Представьте алгоритм поиска наибольшего и наименьшего значений элементов массива. Поясните отличия динамических двумерных массивов от статических. Поясните формат объявления динамических двумерных массивов. Приведите примеры использования двумерных массивов.
24. Расскажите об инициализации двумерных массивов в языке C++. Поясните, как осуществляется доступ к элементам массива, инициализация массива. Приведите примеры и фрагменты программного кода для заполнения матрицы $m \times n$ случайными числами от -15 до 25, от 0 до 20, от a до b. Представьте соответствующие фрагменты программного кода.
25. Расскажите о понятии массива, об индексации элементов массива и размерности массива. Опишите способы объявления и инициализации массивов языке C++. Приведите примеры. Поясните, как осуществляется объявление массива фиксированного размера и объявление динамического массива в среде программирования Microsoft Visual Studio. Представьте программную реализацию генерации псевдослучайных чисел при инициализации одномерных и двумерных статических и динамических массивов в среде Microsoft Visual Studio на языке C++.
26. Поясните алгоритм вставки и алгоритм удаления элементов в массиве на языке C++. Представьте соответствующие фрагменты программного кода.
27. Расскажите о концепции объектно-ориентированного программирования. Опишите идею и цель объектно-ориентированного программирования. Расскажите об основных понятиях объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм.
28. Расскажите о способах описания классов в языке C++, о понятии класса, об определении методов класса, о создании объектов и обращении к атрибутам и методам объектов. Приведите примеры и представьте соответствующие фрагменты программного кода. Представьте программную реализацию решения задачи создания пользовательских классов в C++. Поясните, каким образом осуществляется вызов функций-членов классов из основного приложения.
29. Расскажите об использовании функций в языке C++: прототип, описание функции. Перечислите формальные и фактические параметры функции. Расскажите о локальных и глобальных переменных. Приведите примеры. Расскажите о перегрузке функции.
30. Расскажите об основных единицах построения программ при процедурном программировании на языке C++ – о функциях. Представьте определение функции, правила записи, вызова и передачи параметров. Расскажите о понятии подпрограммы и модульного программирования, о понятии функции, прототипе функции, локальных и глобальных переменных.
31. Опишите, как реализуется решение задачи табулирования функции на определенном отрезке. Приведите соответствующие примеры, листинги программного кода и их тестирование в среде Microsoft Visual Studio.
32. Продемонстрируйте на конкретном примере использование организации вложенных циклов при решении задач в среде программирования Microsoft Visual Studio на языке C++.
33. Представьте программную реализацию разработки консольного приложения с использованием операторов циклов для нахождения суммы и произведения членов числовой последовательности.
34. Расскажите о понятии массива, об индексации элементов массива и размерности массива. Опишите способы объявления и инициализации массивов языке C++. Приведите примеры. Поясните, как осуществляется объявление массива фиксированного размера и объявление динамического массива в среде программирования Microsoft Visual Studio. Представьте

программную реализацию генерации псевдослучайных чисел при инициализации одномерных и двумерных статических и динамических массивов в среде Microsoft Visual Studio на языке C++.

35. Представьте программную реализацию решения задачи создания пользовательских классов в C++. Поясните, каким образом осуществляется вызов функций-членов классов из основного приложения.

36. Расскажите о понятии интерфейса приложения и опишите интерфейсные объекты среды программирования Microsoft Visual Studio. Расскажите об основных управляющих элементах среды программирования Microsoft Visual Studio. Охарактеризуйте функционал и основные свойства элементов управления.

37. Расскажите об объектах визуального выбора в интегрированной среде разработки языка C++: флажок, переключатель. Охарактеризуйте функционал и основные свойства флажка и переключателя. Расскажите об объектах визуального выбора в интегрированной среде разработки языка C++: списки и объединение элементов формы. Охарактеризуйте основные свойства списка и объединения элементов формы.

38. Расскажите об инициализации графики в среде программирования Microsoft Visual Studio и о построении графических примитивов в среде программирования Microsoft Visual Studio.

39. Расскажите об использовании однородной заливки при работе с графическими примитивами в среде программирования Microsoft Visual Studio. Представьте фрагмент программного кода для построения закрашенных объектов с использованием однородной и градиентной заливки.

40. Расскажите о возможности использования различных типов заливки при работе с графическими примитивами в среде программирования Microsoft Visual Studio. Представьте фрагмент программного кода для построения закрашенных объектов с использованием штриховой заливки.

41. Расскажите о возможности использования различных типов линий в среде программирования Microsoft Visual Studio. Представьте фрагмент программного кода для построения штриховых линий различного типа и цвета.

42. Расскажите о возможности использования различных типов заливки при работе с графическими примитивами в среде программирования Microsoft Visual Studio. Представьте фрагмент программного кода для построения закрашенных объектов с использованием градиентной заливки. Поясните, что необходимо выполнить для заливки градиентом в среде программирования Microsoft Visual Studio.

43. Расскажите об инициализации графики в среде программирования Microsoft Visual Studio. Представьте фрагмент программного кода для построения пятиугольника, шестиугольника и семиугольника, отличающихся по цвету границы с использованием различных типов заливки.

44. Продемонстрируйте разработку приложения Windows Forms с использованием элементов управления Label, Button, RadioButton, TextBox. Охарактеризуйте функционал и основные свойства указанных компонентов.

45. Продемонстрируйте разработку приложения Windows Forms с использованием элементов управления CheckBox, NumericUpDown, коллекции ComboBox, ListBox, коллекции CheckedListBox. Охарактеризуйте функционал и основные свойства указанных компонентов.

46. Расскажите о типах данных, создаваемых пользователем: структуры, объединения, перечисления. Поясните синтаксис и продемонстрируйте использование пользовательского типа на конкретном примере в C++.

47. Расскажите о перегрузке операторов в C++.

48. Напишите и протестируйте программный код для решения практического задания в среде программирования Microsoft Visual Studio.

49. Составьте алгоритм использования теоретических и практических знаний для постановки и решения исследовательских задач в области программирования Microsoft Visual Studio.

50. Опишите этапы проектирования содержания образовательных программ в области программирования Microsoft Visual Studio.

8.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Промежуточная аттестация проводится в форме экзамена.

Экзамен по дисциплине или ее части имеет цель оценить сформированность компетенций, теоретическую и практическую подготовку студента, его способность к творческому мышлению, приобретенные им навыки самостоятельной работы, умение синтезировать полученные знания и применять их при решении практических задач.

Устный ответ на экзамене

При определении уровня достижений студентов на экзамене необходимо обращать особое внимание на следующее:

- дан полный, развернутый ответ на поставленный вопрос;
- показана совокупность осознанных знаний об объекте, проявляющаяся в свободном оперировании понятиями, умении выделить существенные и несущественные его признаки, причинно-следственные связи;
- знание об объекте демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей;
- ответ формулируется в терминах науки, изложен грамотным литературным языком, логичен, доказателен, демонстрирует авторскую позицию студента;
- теоретические постулаты подтверждаются примерами из практики.

Вопросы и задания для устного опроса

При определении уровня достижений студентов при устном ответе необходимо обращать особое внимание на следующее:

- дан полный, развернутый ответ на поставленный вопрос;
- показана совокупность осознанных знаний об объекте, проявляющаяся в свободном оперировании понятиями, умении выделить существенные и несущественные его признаки, причинно-следственные связи;
- знание об объекте демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей;
- ответ формулируется в терминах науки, изложен литературным языком, логичен, доказателен, демонстрирует авторскую позицию студента;
- теоретические постулаты подтверждаются примерами из практики.

Оценка за опрос определяется простым суммированием баллов:

Критерии оценки ответа

Правильность ответа – 1 балл.

Всесторонность и глубина (полнота) ответа – 1 балл.

Наличие выводов – 1 балл.

Соблюдение норм литературной речи – 1 балл.

Владение профессиональной лексикой – 1 балл.

Итого: 5 баллов.

Практические задания

При определении уровня достижений студентов при выполнении практического задания необходимо обращать особое внимание на следующее:

- задание выполнено правильно;
- показана совокупность осознанных знаний об объекте, проявляющаяся в свободном оперировании понятиями, умении выделить существенные и несущественные его признаки, причинно-следственные связи;
- умение работать с объектом задания демонстрируется на фоне понимания его в системе данной науки и междисциплинарных связей;
- ответ формулируется в терминах науки, изложен литературным языком, логичен, доказателен, демонстрирует авторскую позицию студента;
- выполнение задания теоретически обосновано.

Оценка за опрос определяется простым суммированием баллов:

Критерии оценки ответа

Правильность выполнения задания – 1 балл.

Всесторонность и глубина (полнота) выполнения – 1 балл.

Наличие выводов – 1 балл.

Соблюдение норм литературной речи – 1 балл.

Владение профессиональной лексикой – 1 балл.

Итого: 5 баллов.

Контрольная работа

Виды контрольных работ: аудиторные, домашние, текущие, экзаменационные, письменные, графические, практические, фронтальные, индивидуальные. Система заданий письменных контрольных работ должна:

- выявлять знания студентов по определенной дисциплине (разделу дисциплины);
- выявлять понимание сущности изучаемых предметов и явлений, их закономерностей;
- выявлять умение самостоятельно делать выводы и обобщения;
- творчески использовать знания и навыки.

Требования к контрольной работе по тематическому содержанию соответствуют устному ответу.

Также контрольные работы могут включать перечень практических заданий.

Критерии оценки ответа

Правильность ответа – 1 балл.

Всесторонность и глубина (полнота) ответа – 1 балл.

Наличие выводов – 1 балл.

Соблюдение норм литературной письменной речи – 1 балл.

Владение профессиональной лексикой – 1 балл.

Итого: 5 баллов.

9. Перечень основной и дополнительной учебной литературы

Основная литература

1. Корчуганова, М. Р. Объектно-ориентированное программирование на C++ : электронное учебное пособие / М. Р. Корчуганова, К. С. Иванов, Л. В. Бондарева ; Министерство образования и науки РФ, Кемеровский государственный университет, кафедра вычислительной математики. – Кемерово : Кемеровский государственный университет, 2015.

– 196 с. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=481559>. – Библиогр. в кн. – ISBN 978-5-8353-1832-2. – Текст : электронный.

2. Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. – Москва : Издательство Юрайт, 2019. – 335 с. – (Бакалавр и специалист). – Режим доступа: <http://biblio-online.ru/bcode/438987>. – ISBN 978-5-534-05123-0. – Текст : электронный.

Дополнительная литература

1. Слабнов, В. Д. Программирование на C++ : лекции / В. Д. Слабнов ; Институт экономики, управления и права (г. Казань). – Казань : Познание, 2012. – 136 с. : табл., схем. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=364222>. – Библиогр. в кн. – ISBN 978-5-8399-0386-9. – Текст : электронный.

2. Березин, Б. И. Начальный курс C и C++ : учебное пособие / Б. И. Березин, С. Б. Березин. – Москва : Диалог-МИФИ, 2012. – 280 с. : ил., табл. – Режим доступа: <http://biblioclub.ru/index.php?page=book&id=448000>. – ISBN 5-86404-075-4. – Текст : электронный.

10. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. <https://www.visualstudio.com/ru> - Microsoft Visual Studio [Электронный ресурс] / Режим доступа: <https://www.visualstudio.com/ru>

2. <http://www.intuit.ru> - Интернет-Университет Информационных Технологий [Электронный ресурс] / Бесплатные учебные курсы по информационным технологиям. – М. : НОУ «ИНТУИТ». - URL: <http://www.intuit.ru>

11. Методические указания обучающимся по освоению дисциплины (модуля)

При освоении материала дисциплины необходимо:

- спланировать и распределить время, необходимое для изучения дисциплины;
- конкретизировать для себя план изучения материала;
- ознакомиться с объемом и характером внеаудиторной самостоятельной работы для полноценного освоения каждой из тем дисциплины.

Сценарий изучения курса:

- проработайте каждую тему по предлагаемому ниже алгоритму действий;
- регулярно выполняйте задания для самостоятельной работы, своевременно отчитывайтесь преподавателю об их выполнении;
- изучив весь материал, проверьте свой уровень усвоения содержания дисциплины и готовность к сдаче зачета/экзамена, выполнив задания и ответив самостоятельно на примерные вопросы для промежуточной аттестации.

Алгоритм работы над каждой темой:

- изучите содержание темы вначале по лекционному материалу, а затем по другим источникам;
- прочитайте дополнительную литературу из списка, предложенного преподавателем;
- выпишите в тетрадь основные понятия и категории по теме, используя лекционный материал или словари, что поможет быстро повторить материал при подготовке к промежуточной аттестации;
- составьте краткий план ответа по каждому вопросу, выносимому на обсуждение на аудиторном занятии;
- повторите определения терминов, относящихся к теме;
- продумайте примеры и иллюстрации к обсуждению вопросов по изучаемой теме;
- подберите цитаты ученых, общественных деятелей, публицистов, уместные с точки зрения обсуждаемой проблемы;
- продумывайте высказывания по темам, предложенным к аудиторным занятиям.

Рекомендации по работе с литературой:

- ознакомьтесь с аннотациями к рекомендованной литературе и определите основной метод изложения материала того или иного источника;
- составьте собственные аннотации к другим источникам, что поможет при подготовке рефератов, текстов речей, при подготовке к промежуточной аттестации;
- выберите те источники, которые наиболее подходят для изучения конкретной темы;
- проработайте содержание источника, сформулируйте собственную точку зрения на проблему с опорой на полученную информацию.

12. Перечень информационных технологий

Реализация учебной программы обеспечивается доступом каждого студента к информационным ресурсам – электронной библиотеке и сетевым ресурсам Интернет. Для использования ИКТ в учебном процессе используется программное обеспечение, позволяющее осуществлять поиск, хранение, систематизацию, анализ и презентацию информации, экспорт информации на цифровые носители, организацию взаимодействия в реальной и виртуальной образовательной среде.

Индивидуальные результаты освоения дисциплины студентами фиксируются в электронной информационно-образовательной среде университета.

12.1 Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

1. 1С: Университет ПРОФ
2. Microsoft Windows 7 Pro
3. Microsoft Office Professional Plus 2010

12.2 Перечень информационных справочных систем (обновление выполняется еженедельно)

1. Справочная правовая система «КонсультантПлюс» (<http://www.consultant.ru>)
2. Информационно-правовая система "ГАРАНТ" (<http://www.garant.ru>)

12.3 Перечень современных профессиональных баз данных

1. Электронная библиотечная система Znanium. [com \(<http://znanium.com/>\)](http://znanium.com/)
2. Единое окно доступа к образовательным ресурсам (<http://window.edu.ru>)
3. Профессиональная база данных «Открытые данные Министерства образования и науки РФ» (<http://xn----8sblcdzzacvuc0jbg.xn--80abucjiibhv9a.xn--plai/ope>)

13. Материально-техническое обеспечение дисциплины(модуля)

Для проведения аудиторных занятий необходим стандартный набор специализированной учебной мебели и учебного оборудования, а также мультимедийное оборудование для демонстрации презентаций на лекциях. Для проведения практических занятий, а также организации самостоятельной работы студентов необходим компьютерный класс с рабочими местами, обеспечивающими выход в Интернет.

Индивидуальные результаты освоения дисциплины фиксируются в электронной информационно-образовательной среде университета.

Реализация учебной программы обеспечивается доступом каждого студента к информационным ресурсам – электронной библиотеке и сетевым ресурсам Интернет.

Для использования ИКТ в учебном процессе необходимо наличие программного обеспечения, позволяющего осуществлять поиск информации в сети Интернет, систематизацию, анализ и презентацию информации, экспорт информации на цифровые носители.

Учебная аудитория для проведения учебных занятий. №206

Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, курсового проектирования (выполнения курсовых работ).

Лаборатория вычислительной техники.

Помещение оснащено оборудованием и техническими средствами обучения.

Основное оборудование:

Автоматизированное рабочее место в составе (системный блок, монитор, клавиатура, мышь), интерактивный дисплей.

Автоматизированное рабочее место (компьютеры – 13 шт.).

Учебно-наглядные пособия:

Презентации.

Помещение для самостоятельной работы, №225.

Помещение оснащено оборудованием и техническими средствами обучения.

Основное оборудование:

Компьютерная техника с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета (персональный компьютер 10 шт.).

Учебно-наглядные пособия:

Презентации.

Помещение для самостоятельной работы.

Читальный зал, №101.

Основное оборудование:

Компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета (компьютер 10 шт., проектор с экраном 1 шт., многофункциональное устройство 1 шт., принтер 1 шт.)

Учебно-наглядные пособия:

Учебники и учебно-методические пособия, периодические издания, справочная литература.

Стенды с тематическими выставками.