ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сведения о сертификате ЭП
Сертификат: 1ea208e57bd5c15d3d0fdd330db1ae60
Владелец: Соколов Максим Александрович
Действителен: с 06.08.2026 по 30.10.2027

**УТВЕРЖДЕНО:**
**Ученым советом Института Сервисных технологий**
**Протокол № 24 от «16» января 2025 г.**

**РАБОЧАЯ ПРОГРАММА**

**ДИСЦИПЛИНЫ (СПО)**

# *ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ*

**основной профессиональной образовательной программы среднего профессионального образования – программы подготовки специалистов среднего звена**

**по специальности:** ***09.02.07 Информационные системы и программирование***

**Квалификация:** ***Специалист по информационным системам***

***год начала подготовки:2025***

**Разработчики:**

| должность | ученая степень и звание, ФИО |
|---|---|
| ***преподаватель*** | ***Коваленко М.В.*** |

**Рабочая программа согласована и одобрена руководителем ППССЗ:**

| должность | ученая степень и звание, ФИО |
|---|---|
| ***преподаватель*** | ***Границына М.С*** |

## *СОДЕРЖАНИЕ*

| | | |
|---|---|---|
| **1** | **Общая характеристика рабочей программы дисциплины** | **3** |
| **2** | **Структура и содержание учебной дисциплины** | **6** |
| **3** | **Методические указания по проведению практических занятий/лабораторных работ/семинаров, занятий в форме практической подготовки (при наличии), и самостоятельной работе** | **14** |
| **4** | **Фонд оценочных средств дисциплины** | **19** |
| **5** | **Условия реализации программы дисциплины** | **36** |
| **6** | **Информационное обеспечение реализации программы** | **36** |

**1.Общая характеристика рабочей программы дисциплины**
**«ОП.04 Основы алгоритмизации и программирования»**
(наименование дисциплины)

**1.1 Место дисциплины в структуре основной образовательной программы:**

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования»» является обязательной частью цикла общепрофессиональных дисциплин примерной основной образовательной программы в соответствии с ФГОС по специальности 09.02.07 Информационные системы и программирование.

Особое значение дисциплина имеет при формировании и развитии ОК: ОК 01, ОК 02, ОК 04, ОК 05, ОК 09, ПК2.4, ПК2.5.

Программа учебной дисциплины может быть использована в дополнительном профессиональном образовании в рамках реализации программ подготовки и переподготовки кадров в учреждениях СПО.

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования» относится к циклу общепрофессиональных дисциплин основной профессиональной образовательной программы.

**1.2 Цель и планируемые результаты освоения дисциплины:**

**В результате освоения дисциплины обучающийся должен уметь:**

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

**В результате освоения дисциплины обучающийся должен знать:**

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- Эволюцию языков программирования, их классификацию, понятие системы программирования.
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- Подпрограммы, составление библиотек подпрограмм
- Объектно-ориентированную модель программирования, основные принципы объектно--ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

В рамках программы учебной дисциплины обучающимися осваиваются умения и знания:

| Код ПК, ОК | Умения | Знания |
|---|---|---|

| | | |
|---|---|---|
| ОК 01<br>ОК 02<br>ОК 04<br>ОК 05<br>ОК 09<br>ПК2.4<br>ПК2.5 | – Разрабатывать алгоритмы для конкретных задач.<br>– Использовать программы для графического отображения алгоритмов.<br>– Определять сложность работы алгоритмов.<br>– Работать в среде программирования.<br>– Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.<br>– Оформлять код программы в соответствии со стандартом кодирования.<br>– Выполнять проверку, отладку кода программы. | – Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.<br>– Эволюцию языков программирования, их классификацию, понятие системы программирования.<br>– Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.<br>– Подпрограммы, составление библиотек подпрограмм<br>– Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения |

## 2.Структура и содержание учебной дисциплины

### 2.1. Объем учебной дисциплины и виды учебной работы

| **Вид учебной работы** | **Объем в часах** |
|---|---|
| **Объем образовательной программы учебной дисциплины** | 162 |
| ***в т.ч. в форме практической подготовки (если предусмотрено)*** | |
| в т. ч.: | |
| теоретическое обучение | 70 |
| лабораторные работы *(если предусмотрено)* | |
| практические занятия *(если предусмотрено)* | 70 |
| курсовая работа (проект) *(если предусмотрено для специальностей*) | |
| контрольная работа *(если предусмотрено)* | |
| консультации | 2 |
| *Самостоятельная работа* [1] | 8 |
| **Промежуточная аттестация** | 12 |

[1] Самостоятельная работа в рамках образовательной программы планируется образовательной организацией в соответствии с требованиями ФГОС СПО в пределах объема учебной дисциплины в количестве часов, необходимом для выполнения заданий самостоятельной работы обучающихся, предусмотренных тематическим планом и содержанием учебной дисциплины.

**2.2. Тематический план и содержание учебной дисциплины Основы алгоритмизации и программирования**

*наименование*

| Наименование разделов и тем | Содержание учебного материала и формы организации деятельности обучающихся | Объем в часах | Коды компетенций, формированию которых способствует элемент программы |
|---|---|---|---|
| **1** | ***2*** | ***3*** | ***4*** |
| **Раздел 1.** | **Введение в программирование** | | |
| Тема 1.1. Алгоритмы и языки программирования | **Содержание учебного материала** | **6** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 1. Основные понятия алгоритмизации. Понятие алгоритма. Свойства алгоритмов. Общие принципы построения алгоритмов. Формы записи алгоритмов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 2. Данные. Структуры данных. Классификация структур данных. Языки программирования. Понятие системы программирования.<br>Развитие языков программирования. Обзор языков программирования. Области применения языков программирования.<br>Стандарты языков программирования. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 3. Среда проектирования. Компиляторы и интерпретаторы. Жизненный цикл программы. Программа. Программный продукт и его характеристики. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **2** | |
| | Практическое занятие № 1.Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Способы отображения алгоритмов. Блок-схемы алгоритмов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| **РАЗДЕЛ 2.** | **Лексические основы языка программирования СИ++. Типы данных. Операторы.** | | |

| | | | |
|---|---|---|---|
| Тема 2.1. Алфавит, лексические элементы языка СИ++ | **Содержание учебного материала** | **6** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 4. Назначение СИ++. Пример программы на СИ++. Схема подготовки исполнения программы. Алфавит СИ++. Лексические элементы языка СИ++ (идентификаторы, ключевые слова) | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 5. Константы. Знаки операций в СИ++. Унарные и бинарные операции. Поразрядные операции. Операции отношения. Логические операции. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 6. Операции присваивания. Операции динамического распределения памяти – new, delete. Условная операция. Операции с компонентами классов и структур. Приоритет операций. Разделители. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **8** | |
| | 2. Практическое занятие № 2. Работа с данными различных типов в СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 3. Практическое занятие № 3. Поразрядные операции, операции сдвигов в языке СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 4. Практическое занятие № 4. Операции отношения. Логические операции в языке СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 5. Практическое занятие № 5. Операции присваивания в языке СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| Тема 2.2. Объекты и их атрибуты | **Содержание учебного материала** | **6** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 7. Объекты и их атрибуты. Скалярные типы и выражения. Определение типов данных. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 8. Объекты и их атрибуты (класс памяти, область действия, видимость, продолжительность существования, тип компоновки). | 2 | ОК01,К02,ОК04, ОК05,ОК09, |

| | | | |
|---|---|---|---|
| | | | ПК2.4, ПК2.5 |
| | 9. Определение и описание объектов. Преобразование типов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **6** | |
| | 6.Практическое занятие № 6. Программирование алгоритмов линейной структуры. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 7.Практическое занятие № 7. Программирование алгоритмов линейной структуры. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 8.Практическое занятие № 8. Программирование алгоритмов линейной структуры. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| Тема 2.3. Операторы языка программирования | **Содержание учебного материала** | **4** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 10. Операторы языка СИ++. Операторы выбора. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 11. Операторы цикла. Операторы передачи управления. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **6** | |
| | 9. Практическое занятие № 9. Программирование алгоритмов разветвляющейся структуры | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 10. Практическое занятие № 10. Программирование алгоритмов циклической структуры | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 11. Практическое занятие № 11. Программирование алгоритмов сложной циклической структуры (вложенные циклы). | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |

| | | | |
|---|---|---|---|
| **Раздел 3.** | **Адреса, указатели, массивы, структуры и объединения в СИ++** | | |
| Тема 3.1. Указатели и адреса объектов, массивы, структуры, объединения | **Содержание учебного материала** | **10** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 12. Указатели и адреса объектов. Адресная арифметика. Типы указателей и операции над ними. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 13. Одномерные числовые массивы. Двумерные числовые массивы. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 14. Массивы и указатели. Символьные массивы. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 15. Структуры и объединения. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 16.Размещение элементов структур и объединений в памяти. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **10** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 12. Практическое занятие № 12.Разработка программ с использованием указателей. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 13. Практическое занятие № 13.Разработка программ с использованием массивов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 14. Практическое занятие № 14.Разработка программ с использованием массивов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 15.Практическое занятие № 15.Разработка программ с использованием структур и объединений. | 2 | ОК01,К02,ОК04, ОК05,ОК09, |

| | | | |
|---|---|---|---|
| | | | ПК2.4, ПК2.5 |
| | 16.Практическое занятие № 16.Контрольная практическая работа. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| **Раздел 4.** | **Функции, указатели, ссылки** | | |
| Тема 4.1. Определение, описание и вызов функций. Ссылки и указатели на функцию. | **Содержание учебного материала** | **10** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 17.Функции. Определение, описание и вызов. Формальные и фактические параметры. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 18. Программирование пользовательских функций. Перегрузка функций. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 19. Ссылки. Использование ссылок в качестве параметров функции, ссылки на функцию, ссылки на возвращаемый функцией результат. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 20. Указатели и адреса объектов. Указатель на функцию. Указатель на возвращаемый функцией результат. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 21. Функции с переменным числом параметров. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **10** | |
| | 17. Практическое занятие № 17. Программирование функций | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 18. Практическое занятие № 18. Программирование функций | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 19. Практическое занятие № 19. Программирование функций со ссылками в качестве формальных параметров. | 2 | ОК01,К02,ОК04, ОК05,ОК09, |

| | | | |
|---|---|---|---|
| | | | ПК2.4, ПК2.5 |
| | 20. Практическое занятие № 20.<br>Программирование функций с переменным числом параметров. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 21. Практическое занятие № 21.<br>Разработка функций. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| **Раздел 5** | **Класс как абстрактный тип** | | |
| Тема 5.1 Класс как абстрактный тип в языке СИ++ | **Содержание учебного материала** | **10** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 22. Классы и объекты классов. Определение данных и методов класса в языке СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 23. Конструкторы и деструкторы класса в СИ++. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 24. Доступность компонентов класса. Инкапсуляция данных. Указатель this. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 25. Примеры программ с классами в СИ++. | 4 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **8** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 22. Практическое занятие № 22. Разработка программ с классом «Комплексное число». | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 23. Практическое занятие № 23. Разработка классов с конструктором класса. Деструктор класса. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 24. Практическое занятие № 24. Разработка программ с изменением доступа к компонентам | 2 | ОК01,К02,ОК04, |

| | | | |
|---|---|---|---|
| | класса. Инкапсуляция данных. | | ОК05,ОК09, ПК2.4, ПК2.5 |
| | 25. Практическое занятие № 25. Разработка программ с классами. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| **Раздел 6** | **Объектно-ориентированное программирование** | | |
| Тема 6.1 Основные принципы объектно-ориентированного программирования (ООП) | **Содержание учебного материала** | **6** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 26. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 27. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 28. Событийно-управляемая модель программирования. Компонентно-ориентированный подход. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **6** | |
| | 26. Практическое занятие № 26. Использование указателей для организации связанных списков. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 27. Практическое занятие № 27. Изучение интегрированной среды разработчика. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 28. Практическое занятие № 28. Создание проекта с использованием компонентов для работы с текстом. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| Тема 6.2 Интегрированная среда разработчика. | **Содержание учебного материала** | **10** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 29. Требования к аппаратным и программным средствам интегрированной среды разработчика. | 2 | ОК01,К02,ОК04, ОК05,ОК09, |

| | | | |
|---|---|---|---|
| | | | ПК2.4, ПК2.5 |
| | 30. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 31. Панель компонентов и их свойства. Окно кода проекта. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 32. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 33. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **10** | |
| | 29. Практическое занятие № 29. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 30. Практическое занятие № 30. События компонентов (элементов управления), их сущность и назначение. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 31. Практическое занятие № 31. Создание процедур на основе событий. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 32. Практическое занятие № 32. Создание проекта с использованием кнопочных компонентов. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 33. Практическое занятие № 33. Создание проекта с использованием компонентов стандартных диалогов и системы меню. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| Тема 6.3. Этапы разработки приложений | **Содержание учебного материала** | **2** | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 34. Проектирование объектно-ориентированного приложения. Создание интерфейса | | |

| | | | |
|---|---|---|---|
| | пользователя. Тестирование, отладка приложения | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **В том числе практических занятий** | **4** | |
| | 34. Практическое занятие № 34. Разработка приложения. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | 35. Практическое занятие № 35. Тестирование, отладка приложения. | 2 | ОК01,К02,ОК04, ОК05,ОК09, ПК2.4, ПК2.5 |
| | **Самостоятельная работа** | | |
| | Проработка конспекта лекций по теме, изучение дополнительной литературы и источников сети интернет. | **8** | |
| **Консультации** | | **2** | |
| **Промежуточная аттестация** | | ***12*** | |
| **Всего:** | | ***162*** | |

**3. Методические указания по проведению практических занятий/лабораторных работ/семинаров, занятий в форме практической подготовки (при наличии) и самостоятельной работе**

Практические занятия проводятся в компьютерном классе в формах:

1) индивидуальная самостоятельная работа по заданию;
2) выполнение проверочных контрольных работ;
3) обучающий тренинг.

**3.1. Тематика и содержание практических занятий**

Тематика практических занятий соответствует рабочей программе дисциплины.

**Практическое занятие № 1.**

Тема: Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические. Способы отображения алгоритмов. Блок-схемы алгоритмов.

– Цель: Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. Использовать программы для графического отображения алгоритмов.

Результаты обучения (умения):

– Разрабатывать алгоритмы для конкретных задач.

**Практическое занятие № 2.**

Тема: Работа с данными различных типов в СИ++.

Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие № 3.**

Тема: Поразрядные операции, операции сдвигов в языке СИ++.

Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие № 4.**

Тема: Операции отношения. Логические операции в языке СИ++.

Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №5.**

Тема: Операции отношения. Логические операции в языке СИ++.

Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №6.**

Тема: Программирование алгоритмов линейной структуры.

Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №7.**
Тема: Программирование алгоритмов линейной структуры.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие № 8.**
Тема: Программирование алгоритмов линейной структуры.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №9.**
Тема: Программирование алгоритмов разветвляющейся структуры
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №10.**
Тема: Программирование алгоритмов циклической структуры
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №11.**
Тема: Программирование алгоритмов сложной циклической структуры (вложенные циклы).
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №12.**
Тема: Разработка программ с использованием указателей.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №13.**
Тема: Практическое занятие № 13.Разработка программ с использованием массивов.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №14**.
Тема: Разработка программ с использованием массивов.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
- Работать в среде программирования.

**Практическое занятие №15**.
Тема: Разработка программ с использованием структур и объединений.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №16.**
Тема: Контрольная практическая работа
Цель: Рубежный контроль.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №17.**
Тема: Программирование функций
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Знать подпрограммы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №18.**
Тема: Программирование функций.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Знать подпрограммы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №19.**
Тема: Программирование функций со ссылками в качестве формальных параметров.
Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Знать подпрограммы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №20.**
Тема: Программирование функций с переменным числом параметров.
– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Знать подпрограммы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №21.**
Тема: Программирование функций.
– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Знать подпрограммы.
Результаты обучения (умения):
– Работать в среде программирования.

**Практическое занятие №22.**
Тема: Разработка программ с классом «Комплексное число».

– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №23.**

Тема: Разработка классов с конструктором класса. Деструктор класса. Разработка программ с классами.

– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №24.**

Тема: Разработка программ с изменением доступа к компонентам класса. Инкапсуляция данных.

– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения)

– Работать в среде программирования.

**Практическое занятие №25.**

Тема: Разработка программ с классами.

– Цель: Знать основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №26.**

Тема: Использование указателей для организации связанных списков.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №27.**

Тема: Изучение интегрированной среды разработчика.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №28.**

Тема: Создание проекта с использованием компонентов для работы с текстом.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №29.**

Тема: Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №30.**

Тема: События компонентов (элементов управления), их сущность и назначение.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №31.**

Тема: События компонентов (элементов управления), их сущность и назначение.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №32.**

Тема: События компонентов (элементов управления), их сущность и назначение. . Создание проекта с использованием кнопочных компонентов.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №33.**

Тема: Создание проекта с использованием компонентов стандартных диалогов и системы меню.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Работать в среде программирования.

**Практическое занятие №34.**

Тема: Разработка приложения.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Оформлять код программы в соответствии со стандартом кодирования.

**Практическое занятие №35.**

Тема: Тестирование, отладка приложения.

– Цель: Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

Результаты обучения (умения):

– Оформлять код программы в соответствии со стандартом кодирования.
– Выполнять проверку, отладку кода программы.

### 3.2. Тематика и содержание самостоятельной работы

На самостоятельную работу студентов выделяется 8 академических часов в рамках часов на дисциплину Основы алгоритмизации и программирования.

Тема: Проработка конспекта лекций по теме, изучение дополнительной литературы и источников сети интернет.
Содержание: Ознакомление с рекомендованными источниками по дисциплине.

## 4. Фонд оценочных средств дисциплины

### 4.1. Результаты освоения учебной дисциплины, подлежащие проверке

Формы промежуточной аттестации по семестрам:

| № семестра | Форма контроля |
|---|---|
| 4 | экзамен |

В результате промежуточной аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также формирования компетенций:

| **Результаты обучения: умения, знания** | **Показатели оценки результата** | **Форма контроля и оценивания** |
|---|---|---|
| **Уметь:** | | |
| Разрабатывать алгоритмы для конкретных задач. | Умеет разрабатывать алгоритмы для конкретных задач. | *Для текущего контроля:* оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование. *Для промежуточной аттестации:* экзамен |
| Использовать программы для графического отображения алгоритмов. | Умеет использовать программы для графического отображения алгоритмов. | *Для текущего контроля:* оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование. *Для промежуточной аттестации:* экзамен |
| Определять сложность работы алгоритмов. | Умеет определять сложность работы алгоритмов. | *Для текущего контроля:* оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование. *Для промежуточной аттестации:* экзамен |
| Работать в среде программирования | Умеет работать в среде программирования | *Для текущего контроля:* оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование. *Для промежуточной аттестации:* экзамен |
| Реализовывать построенные | Умеет реализовывать | *Для текущего контроля:* |

| | | |
|---|---|---|
| алгоритмы в виде программ на конкретном языке программирования. | построенные алгоритмы в виде программ на конкретном языке программирования. | оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование.<br>*Для промежуточной аттестации:*<br>экзамен |
| Оформлять код программы в соответствии со стандартом кодирования. | Умеет оформлять код программы в соответствии со стандартом кодирования. | *Для текущего контроля:*<br>оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование.<br>*Для промежуточной аттестации:*<br>экзамен |
| Выполнять проверку, отладку кода программы. | Умеет выполнять проверку кода программы | *Для текущего контроля:*<br>оценка работы на практических занятиях, оценка выполнения контрольных работ, устный опрос, тестирование.<br>*Для промежуточной аттестации:*<br>экзамен |
| **Знать:** | | |
| Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. | Знает понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции. | *Для текущего контроля:*<br>фронтальный опрос<br>*Для промежуточной аттестации:*<br>Контрольная работа, экзамен |
| Эволюцию языков программирования, их классификацию, понятие системы программирования | Знает эволюцию языков программирования, их классификацию, понятие системы программирования | *Для текущего контроля:*<br>фронтальный опрос<br>*Для промежуточной аттестации:*<br>Контрольная работа, экзамен |
| Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. | Знает основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти. | *Для текущего контроля:*<br>фронтальный опрос<br>*Для промежуточной аттестации:*<br>Контрольная работа, экзамен |
| Подпрограммы, составление библиотек подпрограмм | Знает подпрограммы, составление библиотек подпрограмм | *Для текущего контроля:*<br>фронтальный опрос<br>*Для промежуточной аттестации:*<br>Контрольная работа, экзамен |
| Объектно-ориентированную модель программирования, основные принципы объектно-- | Знает объектно-ориентированную модель программирования, | *Для текущего контроля:*<br>фронтальный опрос<br>*Для промежуточной* |

| | | |
|---|---|---|
| ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуля-ция и полиморфизма, наследования и переоп-ределения | основные принципы объектно-- ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переоп-ределения | *аттестации:* Контрольная работа, экзамен |

Формируемые компетенции:

| **Код формируемой компетенции** | **Наименование компетенции** | **Формы и методы контроля и оценки результатов обучения** |
|---|---|---|
| ОК 1. | Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |
| ОК 2. | . Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |
| ОК 4. | Эффективно взаимодействовать и работать в коллективе и команде | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |
| ОК 5. | Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |
| ОК 09. | Пользоваться профессиональной документацией на государственном и иностранном языке. | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |
| ПК 2.4. | Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения. | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* |

| | | |
|---|---|---|
| | | экзамен |
| ПК 2.5. | Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования. | *Для текущего контроля:* оценка работы на практических занятиях, устный опрос. *Для промежуточной аттестации:* экзамен |

**4.2. Методика применения контрольно-измерительных материалов**
**Формы контроля**
**Текущий контроль** осуществляется в течение семестра в виде оценки результатов практических занятий, оценки выполнения контрольных работ, тестов, оценки устных опросов.
**Промежуточная аттестация** осуществляется в виде экзамена в 4 семестре

**4.3. Контрольно-измерительные материалы включают:**
**4.3.1. Типовые задания для оценки знаний и умений текущего контроля**
Контроль и оценка результатов освоения темы осуществляется преподавателем в процессе выполнения обучающимися индивидуальных заданий в виде внешнего контроля. Формы контроля: проверка практического задания, контрольная работа, тестирование, устный опрос.
**Тестовое задание (по разделам 2-5) :**

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| 1 | Выберите неправильные утверждения: К лексическим элементам языка СИ++ относятся: | а) идентификаторы | б) комментарии | в) ключевые слова | г) знаки операций | б |
| 2 | Идентификатор - это: | а) последовательность из латинских букв, цифр и пробела | б) последовательность из латинских букв, цифр, символа подчёркивания | в) любая последовательность символов | г) последовательность из латинских и русских букв | б |
| 3 | Укажите неправильный идентификатор: | а) ba_d | б) 1fda | в) zabc | г) _AH | б |
| 4 | Укажите восьмеричную константу: | а) 016 | б) ox16 | в) 125 | г) 65. | а |
| 5 | Выберите правильное объявление константы: | а) const int ac=1,502; | б) const int bit word=32; | в) float pi(const)=3.14; | г) float pi const=3.14; | г |
| 6 | Выберите неправильное объявление перечислимых констант: | а) enum {a=1, b=2, c=3}; | б) enum {a,b,c}; | в) enum {a=b*2; 3; d=a+2}; | г) enum {a=2, b=a*2}; | в |
| 7 | Выберите правильную запись символьной | а) ' \n ' | б) " \n ” | в) " \f " | г)" \v " | а |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | константы: | | | | | |
| 8 | Сколько констант в следующем фрагменте кода:int k=3;const int m=5; int m1;m1=k+m; | а) 2 | б) 3 | в) 4 | г) 5 | а |
| 9 | Выберите правильный ответ.Знак операции может: | а) Употребляться в различных выражениях и всегда одинаково интерпретироваться | б) Употребляться в различных выражениях и по- разному интерпретироваться | в) Обозначать только одну операцию | г) Записываться только специальными символами | б |
| 10 | Найдите неверный знак операции СИ++ : | а) ++ | б) && | в) %= | г) =>> | г |
| 11 | Операция получения адреса операнда: | а) * | б) ~ | в) & | г) ! | в |
| 12 | Операция обращения по адресу: | а) new | б) <<= | в)&= | г) * | г |
| 13 | Выберите правильную запись: | а) ++i | б) ++5 | в) 84++ | г) ++(j+k) | а |
| 14 | Выберите правильную запись операции sizeof : | а) (sizeof) int; | б) sizeof (double); | в) sizeof int | г) sizeof float | б |
| 15 | Укажите среди перечисленных битовую операцию: | а) * | б) << | в) + | г) = | б |
| 16 | Укажите среди перечисленных логическую операцию: е) \| \| | а) & | б) ! = | в) <= | г) && | г |
| 26 | Выберите неправильное преобразование типа: | а) long(1) | б) float (1) | в) double (2) | г) unsigned long (2) | г |
| 27 | В приведенном примере найдите имя указателя: int * h; h = new int (10); | а) new | б) int | в) new int | г) h | г |
| 28 | Фрагмент программы на языке СИ++ int*ptr; ptr = new int ( 25); позволяет: | а) присвоить указателю ptr возвращаемое операцией new значение адреса | б) обеспечивает доступ к выделенному участку памяти | в) объявить указатель | г) определить ссылку | а |
| 29 | delete h: | а) выделяет участок памяти | б) освобождает выделенный участок памяти | в) уничтожает переменную h | г) не имеет смысла | б |
| 30 | Укажите наиболее приоритетную операцию из приведенных: | а) && | б) % | в) ^ | г) new | г |
| 31 | Как обозначается условная операция: | а) % = | б) <= | в) >= | г) ? : | г |
| 17 | Укажите операцию | а) = | б) *= | в) = = | г) / = | в |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | отношения: | | | | | |
| 18 | Укажите среди перечисленных операцию отношения: | а) && | б) ^ | в) :: | г) > = | г |
| 19 | Укажите среди перечисленных унарную операцию: | а) ++ | б) >> | в) << | г) < = | а |
| 20 | Укажите среди перечисленных бинарную операцию: | а) ++ | б) ~ | в) sizeof | г) / | г |
| 21 | Выберите операцию, результат выполнения которой равен 7: | а) 6 & 5 | б) 6 ^ 5 | в) 6 \| 5 | г) 6*5 | в |
| 22 | Выберите операцию, результат которой "истина": | а) 25 = = 20 | б) 25 ! = 20 | в) 25 ! =20 && 25 = = 20 | г) 25<20 | б |
| 23 | Выберите операцию указания области видимости переменной: | а) .* | б) - > | в) : : | г) / | в |
| 24 | Операция ( ) используется для: | а) индексирования элементов массива | б) при вызове функции для указания параметров | в) обозначения начало и конца блока | г) для указания размерности массива | б |
| 25 | Выберите правильное объявление: | а) char x [ ] = ‘DIXI’; | б) char y [ ] = ”DIXI”; | в) int a [ ] = “DIXI”; | г) float d=“DIXI”; | б |
| 32 | Укажите смысл разделителя { }: | а) ограничивает индексы элементов массива | б) входит в описание функции | в) обозначает начало и конец блока | | в |
| 33 | Какое ключевое слово не используются в качестве имени скалярных типов данных в языке СИ++ : | а) long | б) char | в) main | г) double | в |
| 34 | Выберите неправильное описание переменной или функции: | а) long belbedin =10.5; | б) int a=10; | в) unsigned int i, j, k; | г) int f1 (void); | а |
| 35 | Объект – это: | а) знак операции | б) программа | в) некоторая именованная область памяти | г) блок | в |
| 36 | Тип данных определяет: | а) требуемое количество памяти, правила интерпретации двоичных кодов, набор допускаемых операций | б) размещение объекта в памяти и продолжительность его существования | в) какие операции допустимы для объекта | г) продолжительность существования объекта в программе | а |
| 37 | Класс памяти определяет: | а) правила интерпретации | б) размер памяти, | в) размещение объекта в | в) какие операции | в |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | | двоичных кодов | выделяемой объекту | памяти и продолжительность его существования | допустимы для объекта | |
| 38 | Класс auto -это: | а) автоматически выделяемая локальная память | б) автоматически выделяемая регистровая память | в) класс памяти, который приписывается глобальным объектам | г) класс памяти, который приписывается статическим объектам | а |
| 39 | Объект, описанный со специдентификатором static, будет : | а) существовать в пределах файла с исходным текстом программы, где он определен | б) существовать в пределах блока, в котором он описан | в) доступен во всех модулях программы | | а |
| 40 | Область действия идентификатора – это: | а) область, где идентификатор определен | б) часть программы , где идентификатор может быть использован для доступа к объекту | в) вся программа | г) блок main | б |
| 41 | Выберите среди описаний прототип функции: | а) typedes unsigned char Simbol; | б) extern const float pi; | в) float fn (int a, double b); | в)int sum; | в |
| 42 | Какой может быть продолжительность существования объекта: | а) статической, динамической, локальной | б) внешней, автоматической, глобальной | в) любой | г) автоматической | а |
| 43 | Укажите допустимые преобразования типов, гарантирующие сохранение точности: | а) float -> double | б) long double -> double | в) unsigned long -> unsigned int | г) int -> short | а |
| 44 | Укажите неверную запись: | а) if (x>0) y=-x; else y=x; | б) if (j>0) {int i; i=2*j; } else i= -j; | в) if(x | г) if (x==1) cout <<"x=1"; | б |
| 45 | Оператор switch применяется для: | а) организации цикла | б) организации ветвления | в) выхода из переключателя | г) это оператор, завершающий выполнение функции | б |
| 46 | Оператор break служит для: | а) выхода из цикла или переключателя | б) выхода из переключателя | в) организации цикла | г) это оператор, завершающий выполнение функции | б |
| 47 | Оператор goto – это: | а) оператор перехода | б) оператор возврата из функции | в) оператор организации цикла | г) это оператор, завершающий выполнение функции | а |
| 48 | Оператор return – это: | а) оператор перехода | б) оператор, завершающий выполнение функции и возвращающий | в) это оператор, завершающий выполнение функции | г) оператор организации цикла | б |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | | | управления в ту точку, откуда функция была вызвана | | | |
| 49 | Функция main в языке СИ++ обеспечивает: | а) вычисления по определенному алгоритму | б) создание точки входа в откомпилированную программу | | | б |
| 50 | В определении функции указывается: | а) имя функции, тип функции, совокупность формальных параметров | б) имя функции и тип функции | в) класс памяти | г) имя функции | а |
| 51 | Тело функции – это: | а) один оператор | б) блок или составной оператор | в) модуль | г) именованная область памяти | б |
| 52 | Укажите имя функции: void write (void) {cout << “\ n Hello!”;} | а) void | б) write | в) cout | г) Hello | б |
| 53 | Укажите тип возвращаемого результата: double exp (double x, unsigned int l); | а) exp | б) unsigned int | в) double | г) x | в |
| 54 | Укажите формальные параметры функции в следующем фрагменте:int sum (int a, int b) {int r; r= a + b; return r; }...int m =2, n =3;int y = sum ( m, n); | а) sum, a, b | б) a, b, r | в) m, n | г) a,b | г |
| 55 | Рекурсия – это: | а) одна функция вызывается из другой | б) функция вызывает саму себя | | | б |
| 56 | Указатель – это: | а) специальный объект в СИ++, значением которого служит адрес участка памяти | б) имя переменной | в) имя константы | г) ссылка | а |
| 57 | Выберите определение указателя: | а) int *i1p; | б) int i2p; | в) int I; | г) float ptr; | а |
| 58 | Укажите неправильное утверждение: | а) в качестве указателя можно использовать явно заданный участок памяти | б) указатель, уже имеющий значение | в) выражение, позволяющее получить адрес объекта с помощью операции & | г) выражение, позволяющее определить содержимое адреса с помощью операции * | г |
| 59 | Укажите правильное определение указателя | а) char a =’*’; | б) char *pc = &a; | в) char *p = 10; | г) unsigned int a; | б |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | на объект типа char: | | | | | |
| 60 | Определить результат выполнения следующего фрагмента программы: char cc1 = ‘a’;char cc = ‘d’;char *pc = &cc;cout << ‘\n’<< *pc; | а) cc | б) pc | в) d | г) a | в |
| 61 | Определите результат выполнения следующего фрагмента программы: char cc =’*’;char *pc =&cc;char *ptr (null);ptr = pc;cout <<< * ptr; | а) CC | б) NULL | в) * | г) pc | в |
| 62 | Операция & применима: | а) для определения адресов неименованных констант | б) для определения адресов объектов, имеющих имя и размещенных в памяти | | | б |
| 63 | Указатель long * ldp использует: | а) 4 байта памяти | б) 10 байтов памяти | в) 8 байтов памяти | в) 12 байтов памяти | а |
| 64 | Выберите из перечисленных операции над указателями: | а) * (операция доступа по адресу) | б) & (операция получения адреса) | в) деление | г) умножение | а |
| 65 | Выберите правильное определение массива целых чисел: | а) float d[ ]; | б) double r[]; | в)int d[5]; | г)char f[]; | в |
| 66 | Номер начального элемента в массиве: | а) 1 | б) 2 | в) 0 | г) -1 | в |
| 67 | В следующем фрагменте программы long ar[]={100,200,300,400}; long *arlo=ar; выполнено: | а)определен массив | б)определен массив, заданы его элементы | в)определен массив, заданы его элементы, определен указатель | г)определен массив, заданы его элементы, определен указатель и связан с массивом | г |
| 68 | После выполнения фрагмента программы float matrix [] [5]={(1), (2), (3)} начальные значения получают элементы: | а)matrix [0] [0]=3 matrix[1][0]=2 matrix [2] [0]=1 | б)matrix [0] [0]=1 | в) matrix [0] [0]=1 matrix [1] [0]=2 matrix [2] [0]=3 | г)matrix [1] [0]=2 matrix [2] [0]=3 | в |
| 69 | Ссылка в языке СИ++- это: | а) другое имя уже существующего объекта | б) содержимое участка памяти | в) адрес участка памяти | г) имя функции | а |
| 70 | Включение ссылок в язык СИ++ необходимо | а) использования | б) повышения эффективности | в) для эффективного | | в |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | для (укажите, что неверно): | имени функции в качестве леводопустимого выражения | обмена с функциями через аппарат формальных параметров | использования памяти | | |
| 71 | Структура - это: | а) совокупность элементов одного типа | б) совокупность элементов разного типа | в) совокупность числовых элементов | г) совокупность тектовых элементов | б |
| 72 | Укажите среди перечисленных директиву препроцессора: | а) #include | б) else | в)if | г) begin | а |
| 73 | Замены в тексте - это директива: | а) #include | б) #if | в) #else | г) #define | г |
| 74 | Компонентная функция класса void define (double re = 0.0, double im = 0.0){real = re;imag = im;} | а) задает значение компонентов real и imag | б) выводит на экран значения real и imag | в) обнуляет значение компонентов real и imag | | а |
| 75 | Класс - это: | а) полный аналог структуры | б) производный структурированный тип. | | | б |
| 76 | Фрагмент программы Complex1 x1, x2, D; | а) объявляет объекты класса Complex1 | б) объявляет указатель на объекты класса Complex1 | в) описывает данные класса | | а |
| 77 | Фрагмент программы Complex1 *point = &D; | а) объявляет объект (комплексное число) класса Complex1 | б) объявляет указатель на объект класса Complex1 | | | б |
| 78 | Конструктор класса нужен: | а) для автоматической инициализации объектов класса | б) для вывода данных на экран | в) для обработки данных | | а |
| 79 | Формат определения конструктора: | а)имя_класса (список_фактических_параметров) {операторы_тела_конструктора}; | б) имя_класса (список_формальных_параметров); | в)имя_класса (список_формальных_параметров) {операторы_тела_конструктора}; | | в |
| 80 | Основной принцип абстракции данных | а) использование конструктора класса | б) доступность компонентных данных | в) использование деструктора класса | г) инкапсуляция данных класса, т.е. сокрытие данных внутри объекта | г |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| 81 | Сигнатуру функции определяют: | а) имя функции | б) имя функции и совокупность формальных параметров | в) совокупность формальных параметров | | б |
| 82 | Определение функции имеет следующий формат: | а) тип_функции имя_функции; | б) тип_функции имя_функции (спецификация_ формальных_параметров) тело_функции; | в) тип_функции имя_функции (спецификация_ формальных_параметров); | | б |
| 83 | Имена функций как имена внешние (тип extern) должны быть : | а) уникальными среди других имен из модулей, в которых используется функция. | б) могут быть любыми | в) уникальными среди других имен данного модуля | | а |
| 84 | Функция float min(float a, float b) {if (a | а) сравнивает два числа | б) возвращает меньший из параметров | в) выводит на экран меньший из параметров | б) возвращает больший из параметров | б |
| 85 | Функция int sum(int a, int b, int c) {int result;result=a+b+c;retur n result;} | а) сравнивает три числа | б) выводит на экран меньший из параметров | в) возвращает сумму параметров | г) возвращает произведение параметров | в |
| 86 | Признаком указателя при лексическом разборе определения является символ | а) ? | б) * | в) & | г) % | б |
| 87 | Работая с указателями, постоянно используют операцию & - получение адреса объекта. Для нее существуют ограничения. Что из перечисленного неверно? | а) нельзя определить адрес неименованной константы, т.е. &3.14 или &’?’. | б) нельзя определить адрес переменной, относящейся к классу памяти register | в) нельзя определять адрес значения, получаемого при вычислении скалярных выражений:& (44*x-z) | г) операция & неприменима к объектам, имеющим имя и размещенным в памяти | г |
| 88 | Компонентная функция класса: void display ( ){cout <<”\n”<<name;cout<< “, розничная цена: “;cout << long (price * (1.0 + goods:: percent*0.01));} | а) выводит данные на экран | б) выводит данные на экран и проводит обработку данных | в) проводит обработку данных | | б |
| 89 | Доступ к статистическому компоненту класса может быть осуществлен тремя способами. Укажите | а) имя класса:: имя_компонент а | б) имя_объекта. имя_класса:: имя_компонента | в) имя_объекта. имя_компонента | г) имя_компонента | г |

| Номер вопроса | Вопрос | Ответ 1 | Ответ 2 | Ответ 3 | Ответ 4 | Правильный ответ(ы) |
|---|---|---|---|---|---|---|
| | неверный. | | | | | |
| 90 | Инициализация статического компонента класса int goods::percent=12; | а) должна быть размещена в глобальной области после определения класса | б) может размещаться в любом месте программы | | | а |
| 91 | В приведенном фрагменте укажите имя класса: void main (void) {goods wares [5]={{“Мужской костюм”, 19000}{“Косметический набор”, 27600}, {“Калькулятор”, 11000}}; | а) wares | б) goods | в) void | г) main | б |
| 92 | Укажите служебное слово, не являющееся спецификатором доступа к компонентам класса : | а) public: | б) private: | в) protected: | г) union | г |
| 93 | В приведенном фрагменте unsigned short int first = 60000; signed short int second = 60000; переменная second равна | а) 60000 | б) -5536 | в)-60000 | г) 5536 | б |
| 94 | Результат выполнения библиотечной функции rand() | а) последовательность случайных чисел в диапазоне от 0 до 32767 | б) последовательность случайных чисел в диане от 0 до RAND_MAX | в) последовательность случайных чисел в диане от 0 до 10 | | б |

**Критерии оценки теста**

Результаты оцениваются следующим образом: «неудовлетворительно» <50 % правильных ответов; «удовлетворительно» 50-69 % правильных ответов; «хорошо» 69-85 % правильных ответов; «отлично»> 85 % правильных ответов.

**Типовые задания для текущего контроля на практическом занятии**

**Задание 1.Каким будет результат выполнения программного кода:**

```
#include <iostream>
using namespace std;
int main()
{int k=35;
cout<<”\n k =k/4 равняется”<< ( k=k/4);
```

```
cout<<"\n k /=1+1+2 равняется"<< ( k/=1+1+2);
cout<<"\n k *=5-2 равняется"<< ( k*=5-2);
cout<<"\n k %=3+2 равняется"<< ( k%=3+2);
cout<<"\n k +=21/3 равняется"<< ( k+=21/3);
cout<<"\n k -=6-6/2 равняется"<< ( k-=6-6/2);
cout<<"\n k <<=2 равняется"<< ( k<<=2);
cout<<"\n k >>=6-5 равняется"<< ( k>>=6-5);
cout<<"\n k &=9+4 равняется"<< ( k&=9+4);
cout<<"\n k |=8-2 равняется"<< ( k|=8-2);
cout<<"\n k ^=10 равняется"<< ( k^=10);
}
```

**Задание 2. Набрать и отладить программу. Каким будет результат выполнения программного кода:**

```
// Функция суммирует //значения своих параметров типа int
#include < iostream >
using namespace std;
long summa (int k, ...) // k-число суммируемых параметров
{int *pik = &k;
long total =0;
for(; k; k --) total += *(++pik);
return total;}
void main()
{cout << "\n summa(2, 6, 4) = " << summa(2,6,4);
cout << "\n summa(6, 1, 2, 3, 4, 5, 6) = " <<
summa(6,l,2,3,4,5,6);
}
```

**Результат выполнения программы:**

```
summa(2, 6, 4) = 10
summa(6, 1, 2, 3, 4, 5, 6) = 21
```

**Задание 3. Набрать и отладить программу. Каким будет результат выполнения программного кода:**

**// определения и вызовы функций**

```
#include<iostream>
using namespace std;
int max(int n, int m) // Определение до вызова функции
{ return n < m? m: n;) // Точка с запятой не нужна
void print (char * name, int value) // Определение функции
{cout << "\n" << name << value; }
float cube(float x) // Определение функции
{ return x * x * x; }
using namespace std;
// main- точка входа в программу
int main ()
{setlocate(LC_ALL, "Russian");
int sum =5, k = 2;
// Вложенные вызовы функций:
sum = max((int)cube(float(k)), sum);
print("\nsum = ",sum);
}
```

**Результат выполнения программы:**
Длина строки sum=8

**Задание 4. Набрать и отладить программу. Каким будет результат выполнения программного кода?**

```
//В следующей программе класс point3 определяет точку в трехмерном пространстве и одновременно содержит статический счетчик N таких точек.
Обращение к счетчику обеспечивает статическая компонентная функция count ().
#include < iostream >
using namespace std;
class point3 // Точка в трехмерном пространстве
{ double x, y, z; // Координаты точки
static int N; // Количество точек (счетчик)
public:
// Конструктор инициализирует значения координат:
point3(double xn = 0.0, double yn = 0.0,
double zn = 0.0)
{ N++; x = xn; y = yn; z = xn; }
// Обращение к счетчику:
static int& count() { return N; }
};
// Внешнее описание и инициализация статического элемента:
int point3::N = 0;
int main ()
{ cout <<"\nsizeof(point3) = " << sizeof(point3);
point3 A(0.0,1.0,2.0);
cout <<"\n sizeof(A) = " <<sizeof(A);
point3 B(3.0,4.0,5.0);
cout <<"\n 0пределены" << point3::count() <<" точки.";
point3 C(6.0,7.0,8.0);
cout <<"\n 0пределены " << B.count() <<" точки.";
}
```

**Результат выполнения программы:**
sizeof(point3) = 24
sizeof(A) = 24
Определены 2 точки.
Определены 3 точки.

### 4.3.2. Типовые задания для оценки знаний и умений промежуточной аттестации

**Перечень теоретических вопросов для проведения экзамена (по разделам 1-6):**

1. Понятие алгоритма и свойства алгоритмов.
2. Основные базовые управляющие структуры.
3. Способы записи алгоритмов.
4. Классификация и примеры языков программирования.
5. Системы программирования.
6. Концепции программирования.
7. Структуры данных.
8. Назначение языка СИ++. Пример программы на СИ++.

9. Алфавит и лексические элементы языка СИ++. Идентификаторы и служебные слова.
10. Константы в языке СИ++. Целые, вещественные и перечислимые константы.
11. Символьные и строковые константы в языке СИ++.
12. Знаки операций в языке СИ++. Приоритет операций. Разделители.
13. Унарные операции в языке СИ++. Пример использования операции sizeof.
14. Аддитивные и мультипликативные операции в языке СИ++. Операции сдвига и поразрядные операции.
15. Операции отношения и логические операции в языке СИ++.
16. Операции присваивания в языке СИ++.
17. Операции выбора компонентов структурированного объекта, операции с компонентами класса, операция «,».
18. Условная операция.
19. Операции new и delete динамического распределения памяти.
20. Скалярные типы данных в языке СИ++. Примеры определения и описания переменных различных типов.
21. Что такое тип данных в языке СИ++? Преобразование типов в языке СИ++. Допустимые преобразования типов. Спецификатор typedef.
22. Определение объекта. Объекты и их атрибуты. Указание типа данных для объекта в программе на СИ++. Что определяет указанный для объекта тип данных?
23. Класс памяти, область действия, видимость объекта, продолжительность существования и тип компоновки.
24. Область действия идентификатора и пространство имен. Привести пример.
25. Операторы передачи управления. Привести примеры.
26. Операторы выбора. Привести примеры.
27. Операторы цикла. Привести примеры.
28. Определение указателя на объект и функцию. Примеры. Формы записи инициализирующего выражения и требования к нему.
29. Определение ссылки. Привести примеры. Формы записи инициализирующего выражения и требования к нему.
30. Ссылка на функцию, на возвращаемый функцией результат, использование ссылок в качестве параметров функции.
31. Определение, описание и вызов функции. Заголовок и тело функции. Сигнатура функции. Примеры определения функций с различными сигнатурами. Функция main.
32. Формальные и фактические параметры. Имена функций. Начальные (умалчиваемые) значения параметров функций. Способы передачи параметров функции и возврата данных.
33. Функция с переменным числом параметров. Привести пример.
34. Перегрузка функций. Привести пример.
35. Дать определение массива, привести способы инициализации и вывода на экран.
36. Массивы в качестве параметров функции. Привести пример программы с массивом в качестве формального параметра функции.
37. Символьные массивы. Способы инициализации.
38. Структуры и объединения, общее и различия. Пример описания структуры. Задание элементов структуры.
39. Определение класса. Компонентные данные и компонентные функции. Пример класса «Комплексное число».
40. Конструктор класса. Привести пример класса «Вектор» с конструктором класса.
41. Привести пример класса «Товары на складе» с конструктором класса.
42. Доступность компонентов класса. Инкапсуляция данных.
43. Привести пример класса «Комплексное число» с инкапсуляцией данных.
44. История развития ООП.
45. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
46. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.

47. Классы объектов. Компоненты и их свойства.
48. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.
49. Требования к аппаратным и программным средствам интегрированной среды разработчика.
50. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.
51. Панель компонентов и их свойства. Окно кода проекта.
52. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
53. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.
54. Разработка приложения. Проектирование объектно-ориентированного приложения. Создание интерфейса пользователя.
55. Тестирование, отладка приложения.

**Типовые практические задания для оценки знаний и умений промежуточной аттестации**

1. Составить программу линейной структуры.
2. Составить программу разветвляющейся структуры.
3. Составить программу циклической структуры
4. Составить программу поиска элемента массива.
5. Составить программу сортировки массива.
6. Разработать функцию.
7. Разработать класс.
8. Составить программу с использованием указателей или ссылок.
9. Составить программу, используя для форматирования вывода компонентные функции класса ios.

На экзамене студент отвечает на теоретический вопрос и выполняет практическое задание по изученному материалу.

Студент создает компьютерные файлы, сохраняя их на диске под своей фамилией. Выполненное задание оценивается по пятибальной системе.

**ПК 2.4 Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.**

Задание закрытого типа на установление последовательности

**Задание:** Установите последовательность шагов для разработки тестового сценария для программного обеспечения:

1. Определение требований к тестированию
2. Разработка тестовых случаев
3. Выполнение тестов
4. Анализ результатов тестирования

Задание закрытого типа на установление соответствия

**Задание:** Соответствие между терминами и их определениями:

1. Тестовый случай
2. Тестирование
3. Отчет о тестировании

a. Процесс проверки работоспособности программного обеспечения
b. Документ, описывающий, как будет проводиться тестирование
c. Описание конкретного сценария для проверки функциональности

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

**Задание:** Какой из следующих элементов является основным для создания тестового сценария?
a) Сложные алгоритмы
b) Четкое описание входных данных и ожидаемого результата
c) Использование только автоматизированных тестов
d) Игнорирование требований к программному обеспечению

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

**Задание:** Выберите действия, которые способствуют разработке качественных тестовых наборов (выберите несколько вариантов):
a) Определение четких критериев успешности
b) Игнорирование документации по требованиям
c) Проведение ревизии тестовых случаев
d) Использование разнообразных методов тестирования

Задание открытого типа с развернутым ответом

**Задание:** Опишите, как вы разрабатываете тестовые наборы и тестовые сценарии для программного обеспечения. Приведите примеры из своего опыта.

**ПК 2.5 Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.**

Задание закрытого типа на установление последовательности

**Задание:** Установите последовательность шагов для инспектирования кода на соответствие стандартам кодирования:

1. Ознакомление с требованиями стандартов кодирования
2. Проведение инспекции кода

3. Документирование найденных проблем
4. Предоставление рекомендаций по исправлению

Задание закрытого типа на установление последовательности

**Задание:** Соответствие между терминами и их определениями:

1. Стандарты кодирования
2. Инспекция кода
3. Отчет по инспекции

a. Набор правил и рекомендаций для написания кода
b. Процесс проверки кода на соответствие стандартам
c. Документ, содержащий результаты инспекции и рекомендации

Задание комбинированного типа с выбором одного верного ответа из четырех предложенных и обоснованием выбора

**Задание:** Какой из следующих аспектов является наиболее важным при инспекции кода?
a) Сложность алгоритмов
b) Соответствие стандартам кодирования
c) Использование последних технологий
d) Количество комментариев в коде

Задание комбинированного типа с выбором нескольких вариантов ответа из предложенных и развернутым обоснованием выбора

**Задание:** Выберите действия, которые способствуют эффективной инспекции кода (выберите несколько вариантов):
a) Использование автоматизированных инструментов анализа кода
b) Игнорирование документации по стандартам
c) Проведение совместных инспекций с командой
d) Оценка кода только по количеству строк

Задание открытого типа с развернутым ответом

**Задание:** Опишите, как вы производите инспекцию компонентов программного обеспечения на предмет соответствия стандартам кодирования. Приведите примеры из своего опыта.

### 4.4. Критерии и показатели оценивания

**Для текущего контроля**

| Оценка | Форма контроля | Критерии оценивания | Показатели оценивания |
|---|---|---|---|

| | | | |
|---|---|---|---|
| **«5»** | устный ответ | полнота и правильность ответа, степень осознанности, понимания изученного материала, четкость и грамотность речи. | ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, литературным языком: ответ самостоятельный. |
| **«4»** | устный ответ | полнота и правильность ответа, степень осознанности, понимания изученного материала, четкость и грамотность речи. | ответ полный и правильный на основании изученных теорий; материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию учителя. |
| **«3»** | устный ответ | полнота и правильность ответа, степень осознанности, понимания изученного материала, четкость и грамотность речи. | ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный. |
| **«2»** | устный ответ | полнота и правильность ответа, степень осознанности, понимания изученного материала, четкость и грамотность речи. | при ответе обнаружено непонимание учащимся основного содержания учебного материала или допущены существенные ошибки, которые учащийся не смог исправить при наводящих вопросах учителя. |

**Для промежуточной аттестации**

| **Оценка** | **Форма контроля** | **Критерии оценивания** | **Показатели оценивания** |
|---|---|---|---|
| **«5»** | Выполнение индивидуального задания | полнота и правильность выполнения задания | Задание выполнено правильно в полном объеме в соответствии с выданным заданием |
| **«4»** | Выполнение индивидуального задания | полнота и правильность выполнения задания | Задание выполнено правильно в полном объеме в соответствии с выданным заданием, имеются 1-2 ошибки исправленные самостоятельно по просьбе преподавателя |
| **«3»** | Выполнение индивидуального задания | полнота и правильность выполнения задания | Задание выполнено более чем на 50% |
| **«2»** | Выполнение индивидуального | полнота и правильность выполнения задания | Задание выполнено менее чем на 50% или не выполнено |

| | | | |
|---|---|---|---|
| | задания | | вовсе. |

**5.Условия реализации программы дисциплины**

**5.1.** Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

Лаборатория «Программирования и баз данных», оснащенный оборудованием: компьютерные столы, 11 персональных компьютеров, сетевое оборудование.

Технические средства обучения: проекционный экран 1, проектор 1, плакаты, стенды, доска.

Программное обеспечение: Windows 10, Microsoft Office 2016, Visual Studio 2019, СПС Консультант Плюс, Учебная версия программы 1С: Предприятие 8.

**6.Информационное обеспечение реализации программы**

Для реализации программы библиотечный фонд образовательной организации, преподаватели, студенты имеют доступ к электронной библиотечной системе Znanium.com.

**6.1. Основная литература**

1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ИНФРА-М, 2026. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-16-021353-8. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2222607

2. Гуриков, С.Р.Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154.
Режим доступа: https://znanium.com/read?id=397332

3. Объектно-ориентированное программирование на C++ : учебник / И. В. Баранова, С. Н. Баранов, И. В. Баженова [и др.]. – Красноярск : Сиб. федер. ун-т, 2019. – 288 с.ISBN 978-5-7638-4034-6
Режим доступа: https://znanium.com/read?id=380554

**6.3. Дополнительная литература**

1.Игнашева, Е.П. Системы счисления, алгоритмизация и программирование : учебное пособие / Е.П. Игнашева. — Москва : ИНФРА-М, 2023. — 224 с. — (Военное образование).
Режим доступа: https://znanium.com/read?id=426189

2. Гуриков, С.Р.Основы алгоритмизации и программирования на языке Microsoft Visual Basic :учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М,
2022. — 594 с. — (Среднее профессиональное образование).
Режим доступа: https://znanium.com/read?id=397645

3. Научно-технический и научно-производственный журнал «Информационные технологии» http://novtex.ru/IT/index.htm

4. Журнал «Бизнес-информатика» https://bijournal.hse.ru/

5. Журнал «Информационные системы и технологии» http://oreluniver.ru/science/journal/isit

6. Журнал «Электронные информационные системы»

https://elins.ru/eis-magazine/