

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета ФИТКБ



Грехихин А.В./

18.08.2025 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Языки программирования»**

**Специальность 10.05.02 Информационная безопасность телекоммуникационных систем**

**Специализация специализация № 9 "Управление безопасностью телекоммуникационных систем и сетей"**

**Квалификация выпускника специалист по защите информации**

**Нормативный период обучения 5 лет и 6 м.**

**Форма обучения очная**

**Год начала подготовки 2025**

Автор программы  
Заведующий кафедрой  
Систем информационной  
безопасности

А.С. Кольцов

А.Г. Остапенко

Руководитель ОПОП

С.С. Куликов

Воронеж 2025

## 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

**1.1. Цели дисциплины:** Овладение фундаментальными знаниями и практическими навыками по применению языков программирования для реализации алгоритмов и разработки программного обеспечения для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности.

### 1.2. Задачи освоения дисциплины

- Изучение стандартов языков программирования.
- Изучение концепций программирования.
- Изучение стандартных библиотек и платформ языков программирования.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Языки программирования» относится к дисциплинам обязательной части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Языки программирования» направлен на формирование следующих компетенций:

ОПК-7 - Способен создавать программы на языке высокого уровня, применять существующие реализации структур данных и алгоритмов;

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-7	Знать <ul style="list-style-type: none"><li>– представления данных в памяти компьютера;</li><li>– основные конструкции и библиотеки языка программирования;</li><li>– принципы построения программ в процедурно-ориентированной и объектно-ориентированной парадигмах.</li></ul>
	Уметь <ul style="list-style-type: none"><li>– реализовывать алгоритмы на языке программирования;</li><li>– работать с интегрированной средой разработки программного обеспечения;</li><li>– использовать сторонние программные библиотеки.</li></ul>
	Владеть <ul style="list-style-type: none"><li>– навыками разработки и тестирования программ по поставленной спецификации.</li></ul>

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Языки программирования» составляет 7 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры	
		2	3
<b>Аудиторные занятия (всего)</b>	126	54	72
В том числе:			
Лекции	54	18	36
Лабораторные работы (ЛР)	72	36	36
<b>Самостоятельная работа</b>	126	90	36
<b>Курсовая работа</b>	+		+
Виды промежуточной аттестации - зачет, зачет с оценкой	+	+	+
Общая трудоемкость: академические часы зач.ед.	252	144	108
	7	4	3

#### 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий  
**очная форма обучения**

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	Основы языка программирования C/ C++	Характеристика языка, стандарты, этапы создания исполняемого файла. Инструментальные средства разработки программного обеспечения на языке C/C++. Базовые конструкции языка C/C++. Алфавит, идентификаторы, служебные слова, константы. Операторы последовательного выполнения, операторы выбора, операторы цикла, операторы передачи управления/	14	16	30	60
2	Основы структурного программирования	Указатели и ссылки. Операции с указателями. Организация работы с динамической памятью. Массивы. Одномерные массивы.	14	18	32	64

		<p>Двумерные массивы.  Динамические массивы.  Строки. Структуры,  объединения, битовые поля.  Перечислимые типы.  Переопределение типов.  Форматируемый ввод-вывод  данных. Работа с файлами.  Функции. Определение,  описание и вызов функции.  Способы передачи параметров  в функцию. Рекурсивные  функции. Указатели на  функции. Встраиваемые  функции. Параметры по  умолчанию. Параметры  функции main. Функции с  переменным числом  параметров. Перегрузка  функций. Шаблоны функций.  Классы памяти. Область  видимости, действия, время  жизни переменной.  Пространства имен.  Пространство имен std.  Препроцессорные средства.  Директивы и стадии  препроцессорной обработки.  Условная компиляция.  Макроподстановки. Операции  с разрядами. Поразрядные  логические операции и  операции сдвига.</p>				
3	Объектно-ориентированное программирование на языке C++	<p>Понятия абстракции,  абстрактного типа данных.  Основные понятия  объектно-ориентированного  программирования (ООП).  Инкапсуляция, наследование и  полиморфизм. Классы.  Понятие класса. Доступ к  компонентам класса.  Конструкторы и деструкторы.  Поля и методы класса.  Указатель this. Статические  компоненты класса.  Дружественные функции и  классы. Перегрузка  операторов. Локальные  классы. Наследование.  Полиморфизм. Виртуальные  функции. Абстрактные классы.</p>	14	20	34	68

		Потоковые средства ввода/вывода данных. Иерархия классов потоков в языке C++. Форматированный ввод/вывод. Создание собственных манипуляторов. Файловый ввод/вывод. Исключительные ситуации. Основные понятия. Механизм обработки исключений. Классы исключений. Шаблоны классов. Определение шаблона классов, типы параметров шаблона. Инстанцирование шаблона. Специализация шаблона.				
4	Стандартная библиотека шаблонов в языке программирования C++	Статическое и динамическое связывание. Принципы работы, использование стандартных библиотек. Создание статических и динамических библиотек пользователя. Общие принципы функционирования STL. Функторы. Контейнеры. Итераторы. Алгоритмы. Адаптеры. Функциональные объекты.	12	18	30	60
<b>Итого</b>			<b>54</b>	<b>72</b>	<b>126</b>	<b>252</b>

## 5.2 Перечень лабораторных работ

1. Программирование алгоритмов линейной структуры. Арифметические функции
2. Программирование алгоритмов разветвленной структуры. Управляющие конструкции.
3. Программирование циклов с неизвестным заранее числом повторений.
4. Программирование циклов с параметром.
5. Стандартные алгоритмы работы с одномерными массивами
6. Обработка двумерных массивов
7. Динамические массивы
8. Строки
9. Работа с текстовыми файлами
10. Подпрограммы. Функции. Указатели.
11. Void-функции.
12. Рекурсивные функции.
13. Работа с текстовыми файлами
14. Классы и объекты
15. Конструкторы и деструкторы

16. Статические и константные члены класса
17. Создание классов с использованием наследования
18. Виртуальные методы и полиморфизм
19. Обработка исключений
20. Ассоциация и зависимость
21. Агрегация и композиция
22. Стандартные методы приведения типов
23. Оператор и конструктор преобразования
24. Шаблоны
25. Стандартная библиотека шаблонов STL
26. Проектирование иерархии классов, полиморфизм.
27. Работа с классами, реализующими формирование и проверку цифровой подписи.

## **6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ**

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсовой работы в 3 семестре для очной формы обучения.

Примерная тематика курсовой работы: ««Разработка программного обеспечения на базе принципов объектно-ориентированного программирования» (по вариантам)

Варианты заданий:

- 1) Разработка интерактивной программы записи чисел в различных системах счисления.
- 2) Разработка программы интерактивного календаря.
- 3) Разработка программы взлома шифра Цезаря.
- 4) Разработка программы шифрования/дешифрования коротких сообщений.
- 5) Разработка программы очистки кода от комментариев.
- 6) Разработка программы сжатия текста.
- 7) Разработка программы автоматизации работы пункта проката
- 8) Разработка программы «Контрольная по информатике. Выходной тестовый контроль
- 9) Разработка программы «Успеваемость студентов»
- 10) Разработка программы «Решение алгебраических уравнений n-й степени»
- 11) Разработка программы «Операции с алгебраическими матрицами. Сложение матриц»
- 12) Разработка программы «Операции с алгебраическими матрицами. Нахождение обратной матрицы
- 13) Разработка программы «Автоматизация рабочего места кассира по продаже билетов на поезда»
- 14) Разработка программы «Оболочка для анкетирования»
- 15) Разработка программы «Телефонный справочник»

- 16) Разработка программы «Расчет стоимости электроэнергии»
- 17) Разработка программы «Автоматическое рабочее место для работника склада»
- 18) Разработка программы «Вычисление интеграла методом Ньютона - Котеса»
- 19) Разработка программы «Вычисление затрат аварийно-диспетчерской службы предприятия»
- 20) Разработка программы «Паспортный стол»
- 21) Разработка программы «Морской бой»
- 22) Разработка программы «Крестики - нолики»
- 23) Разработка программы «Система классов, описывающих различные транспортные средства»
- 24) Разработка программы «Система классов для реализации матричной арифметики»
- 25) Разработка программы «Система классов, описывающих сотрудников предприятия/организации с их функциями (сотрудник, менеджер, ...)»

Задачи, решаемые при выполнении курсовой работы:

- Анализ объекта, определение статических и динамических характеристик.
- Разработка алгоритма, реализующего задачу.
- Реализация алгоритма, реализующего задачу на выбранном языке программирования.

Курсовая работа включает в себя графическую часть и расчетно-пояснительную записку.

## **7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ**

**7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

### **7.1.1 Этап текущего контроля**

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

- «аттестован»;
- «не аттестован».

<b>Компетенция</b>	<b>Результаты обучения, характеризующие сформированность компетенции</b>	<b>Критерии оценивания</b>	<b>Аттестован</b>	<b>Не аттестован</b>
ОПК-7	Знать – представления данных в памяти компьютера; – основные конструкции и библиотеки языка	Активное участие в устных опросах на занятиях, защита лабораторных	Выполнение работ в срок, предусмотренный в рабочих	Невыполнение работ в срок, предусмотренный в

	программирования; – принципы построения программ в процедурно-ориентированной и объектно-ориентированной парадигмах.	работ, ответы на теоретические и практические вопросы промежуточного контроля	программах	рабочих программах
	Уметь – реализовывать алгоритмы на языке программирования; – работать с интегрированной средой разработки программного обеспечения; – использовать сторонние программные библиотеки.	Выполнение лабораторных работ, выполнение этапов курсовой работы.	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть – навыками разработки и тестирования программ по поставленной спецификации.	Решение стандартных и конкретных прикладных задач на лабораторных работах, выполнение этапов курсовой работы	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 2, 3 семестре для очной формы обучения по двух/четырёхбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ОПК-7	Знать – представления данных в памяти компьютера; – основные конструкции и библиотеки языка программирования; – принципы построения программ в процедурно-ориентированной и объектно-ориентированной парадигмах.	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь – реализовывать алгоритмы на языке программирования; – работать с интегрированной средой разработки программного обеспечения; – использовать сторонние программные библиотеки.	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть	Решение прикладных задач в	Продемонстрирован	Задачи не решены

	– навыками разработки и тестирования программ по поставленной спецификации.	конкретной предметной области	верный ход решения в большинстве задач	
--	---	-------------------------------	--	--

или

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-7	Знать – представления данных в памяти компьютера; – основные конструкции и библиотеки языка программирования; – принципы построения программ в процедурно-ориентированной и объектно-ориентированной парадигмах.	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	Уметь – реализовывать алгоритмы на языке программирования; – работать с интегрированной средой разработки программного обеспечения; – использовать сторонние программные библиотеки.	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть – навыками разработки и тестирования программ по поставленной спецификации.	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

## 7.2 Примерный перечень оценочных средств (типичные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

### 7.2.1 Примерный перечень заданий для подготовки к тестированию

1. С помощью чего реализуется принцип полиморфизма в C ++?

а) наличия множественного наследования.

- б) наличия виртуальных методов.
- в) использование виртуального наследования.
- г) наличия абстрактных классов.

2. Какая из перечисленных функций не может быть конструктором?

- а) `void String ()`
- б) `String ()`;
- в) `String (String & s)`
- г) `String (const int a)`

3. Отметьте правильное утверждение для абстрактного класса для языка C++.

- а) Класс, у которого все методы чисто виртуальные, называется абстрактным.
- б) Абстрактный базовый класс навязывает определенный интерфейс всем производным из него классам.
- в) Невозможно создать объект абстрактного класса.
- г) В абстрактном классе не описываются методы вообще.

4. Если в программе на языке C++ в производном классе переопределена операция `new` то ...

- а) все объекты этого класса и все объекты классов, выведенных из него, будут использовать эту операцию независимо от зоне видимости, в которой она переопределена.
- б) производные от этого класса могут использовать глобальную операцию применив операцию `базовый_класс :: new`.
- в) операцию `new` нельзя переопределить.
- г) в любом случае эта операция будет доступна только в пределах класса потомка.

5. Какой из перечисленных методов может быть конструктором для класса

- а) `String` в языке C++?
- б) `String * String ()`;
- в) `void String ()`;
- г) `String (String & s)`;
- д) `const String (int a)`.

6. Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает `void`.

- а) `virtual void SomeFunction (int x)`;
- б) `void SomeFunction (int x) virtual`;
- в) `virtual SomeFunction (int x)`;
- г) `virtual void SomeFunction (int * x)`;

7. Укажите правильное использование оператора friend.

- а) `class A {int_friend CountPass (); private: short i;};`
- б) `class A {public: friend int H :: CountPass (); private: short i;}; +`
- в) `class A {public: int A1 :: CountPass (); friend: short i;};`
- г) `class A {public: friend int H :: q; short i;};`

8. Принцип объектно-ориентированного программирования, заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

- а) Наследование.
- б) Сочетание.
- в) Инициализация.
- г) Инкапсуляция.

9 В программе описано абстрактный класс A и производный от этого класса класс A1. Какой из записей заведомо неверный?

- а) `A * a = new A;`
- б) `A1 a1;`
- в) `A1 a1; A & a = a1;`
- г) `A1 a1; A1 a2;`

10. Выберите верное утверждение о деструкторе класса в C ++.

- а) Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить.
- б) Деструктор принимает в качестве параметра указатель this.
- в) Деструктор не содержит параметров. +
- г) Деструктор принимает в качестве параметра тот объект, который нужно уничтожить.

11 Какой из вариантов записи абстрактного класса в C ++ является правильным?

- а) `abstract class A {virtual int f () = 0;};`
- б) `class A {virtual int f () = 0;}; +`
- в) `class A {virtual int f () = 0;} abstract;`
- г) `class A {virtual int f ();};`

12 Что такое ООП:

- а) объектно-ориентированное программирование
- б) онтологические основы программирования
- в) организация обособленных программистов

13 Выберите правильное подключения заголовочного файла данных *cmath* к вашей программе:

- а) `#include cmath`
- б) `#include "cmath"`
- в) `include "cmath"`
- г) `include "cmath";`

- д) `include cmath;`
- е) `# cmath include`

14 В какой из перечисленных строк компилятор выдаст ошибку:

- а) `unsigned int count = -1;`
- б) `int x, y, z = 0x65;`
- в) `char s = 0`

15 Какое значение будет иметь переменная *result* после выполнения следующих строк кода:

```
double result = 0,5;  
result = (result + 0.5) * 4 + 0.5;
```

- а) 2.5
- б) 4.5
- в) компилятор выдаст ошибку

### 7.2.2 Примерный перечень заданий для решения стандартных задач

1. Свойство `Items` компонента `ComboBox` обозначает:

- а) количество элементов в списке;
- б) текст в поле редактирования;
- в) сами элементы списка;
- г) размер компонента.

2. Для отображения текстовой информации нельзя применить компонент:

- а) `richtextbox;`
- б) `checkbox;`
- в) `label;`
- г) `textbox.`

3. К вещественным типам в `C#` относятся:

- а) `float, single;`
- б) `single, double;`
- в) `real, double;`
- г) `float, real.`

4. Строка, полученная из строки `s1` путем замены всех символов `s2` на символы `s3`:

- а) `Replace(s1, s2, s3);`
- б) `s1=Replace(s2, s3);`
- в) `s1.Replace(s2,s3);`
- г) `s1= s2 Replace s3.`

5. Математические функции в языке `C#` принадлежат пространству имен:

- а) `Namespace;`
- б) `Algebra;`

- в) Mathematics;
- г) Math.

6. В языке C# поддерживаются массивы:

- а) только статические;
- б) только динамические;
- в) и статические, и динамические;
- г) не поддерживаются никакие массивы.

7. Строковый тип данных в C#:

- а) не существует, записывается как массив символов;
- б) существует, обозначается string;
- в) не объявляется, но можно преобразовать к строке ToString();
- г) содержится в специальном пространстве имен и подключается через библиотеку #include «string.h».

8. Процедура обработки кнопки в C# имеет вид:

- а) private void button1\_Click(object sender, EventArgs e);
- б) procedure TForm1.Button1Click(Sender: TObject);
- в) Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click;
- г) procedure Form1.button1\_Click(sender: Object; e: EventArgs).

9. Для вывода табличных данных можно использовать компонент:

- а) A[i]:=strtoint(datagridview1[0,i-1].Value.ToString);
- б) a(i) = Val(DataGridView1.Item(0, i).Value);
- в) a[i]:=strtoint(stringgrid1.cells[i,1]);
- г) a[i] = Convert.ToSingle(dataGridView1[0, i].Value).

10. Датчик случайных чисел описывается командами:

- а) srand(time(NULL)); a= rand() % 10;
- б) a= int(random()\*10);
- в) randomize; a:=random(10);
- г) Random rand = new Random(); a= rand.Next(0, 10).

### 7.2.3 Примерный перечень заданий для решения прикладных задач

Реализовать пользовательский класс в соответствии с вариантом задания. При реализации классов поля должны быть скрытыми. Определить метод установки свойств (при недопустимых аргументах функции возвращать «false» и выдавать текст ошибки на экран). Определить метод чтения свойств. Написать демонстрационную программу, в которой показать использование объектов созданного класса.

Вариант 1 Класс Треугольник Свойства: три стороны Операции:

- увеличение/уменьшение размера сторон в заданное количество раз;
- вычисление периметра;
- вычисление площади;
- определение значений углов.

Вариант 2.

Класс Треугольник Свойства: три стороны Операции:

- увеличение/уменьшение размера сторон на заданное количество процентов;
- вычисление средней линии для любой из сторон;
- определение вида треугольника по величине углов (Остроугольный, Тупоугольный, Прямоугольный);
- определение значений углов.

Вариант 3.

Класс Треугольник

Свойства: две стороны и угол между ними Операции:

- увеличение/уменьшение размера угла на заданное количество процентов;
- определение вида треугольника по числу равных сторон (Разносторонний, Равнобедренный, Равносторонний);
- определение расстояния между центрами вписанной и описанной окружностей.
- определение значений углов.

Вариант 4.

Класс Треугольник

Свойства: две стороны и угол между ними Операции:

- уменьшение/увеличение размера угла (из свойств) в заданное количество раз;
- вычисление длины биссектрисы, принадлежащей любому углу;
- вычисление длин отрезков, на которые биссектриса делит любую сторону;
- определение значений углов.

Вариант 5.

Класс Треугольник

Свойства: сторона и два прилежащих к ней угла Операции:

- уменьшение/увеличение размера стороны (из свойств) в заданное количество раз;
- вычисление длины медианы, принадлежащей любой стороне;
- определение подобен ли другой треугольник данному (указанному по индексу массива);
- определение значений сторон.

Вариант 6.

### Класс Треугольник

Свойства: сторона и два прилежащих к ней угла

- Операции:
- увеличение/уменьшение значения любого угла (из свойств) на заданное количество процентов;
  - вычисление длины высот, принадлежащей любой стороне;
  - определение значений сторон.

### Вариант 7.

Класс Прямоугольный треугольник

- Свойства: две стороны
- Операции:
- увеличение/уменьшение размера любой стороны (из свойств) на заданное количество процентов;
  - вычисление радиуса описанной окружности;
  - вычисление полупериметра;
  - определение значений углов.

### Вариант 8.

Класс Прямоугольный треугольник

- Свойства: сторона и угол
- Операции:
- уменьшение/увеличение размера любой стороны (из свойств) на заданный процент;
  - вычисление радиуса вписанной окружности;
  - определение расстояния между центрами вписанной и описанной окружностей;
  - вычисление квадратного корня из площади;
  - определение значений сторон.

### Вариант 9.

Класс Равнобедренный треугольник

- Свойства: основание и боковая сторона
- Операции:
- увеличение/уменьшение размера на определенный процент;
  - вычисление длины медианы, принадлежащей любой стороне;
  - вычисление периметра и площади;
  - определение значений углов.

### Вариант 10.

Класс Равнобедренный треугольник

Свойства: боковая сторона и угол при основании

Операции:

- увеличение/уменьшение размера в заданное количество раз;
- вычисление длины биссектрисы, принадлежащей любому углу;
- вычисление длины высот, принадлежащей любой стороне;
- определение значений сторон.

## 7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Этапы развития технологий программирования
2. Понятие класса и объекта класса. Создание объектов. Данные и методы класса

3. Передача параметров в методы класса
4. Конструкторы и деструкторы
5. Инкапсуляция
6. Индексаторы
7. Свойства индексаторов
8. Основное понятие наследования. Наследование методов и данных
9. Виртуальные методы и их переопределение
10. Применение абстрактных классов
11. Интерфейсы
12. Ссылочные типы и типы значения
13. Тип enum
14. Типы структур
15. Арифметические операторы
16. Операторы отношения и логические операторы
17. Управляющие операторы
18. Операции для работы с указателями
19. Объявление и инициализация одномерного массива.
20. Объявление и инициализация двумерного массива.
21. Доступ к элементам одномерного массива по индексу.
22. Доступ к элементам одномерного массива по адресу.
23. Доступ к элементам двумерного массива по индексу.
24. Доступ к элементам двумерного массива по адресу.
25. Строки. Постоянство строк

### **7.2.5 Примерный перечень заданий для решения прикладных задач**

1. История языков программирования. Эволюция архитектуры программного обеспечения.
2. Парадигмы программирования. Общие принципы построения и использования языков программирования; средства описания данных; средства описания действий.
3. Теоретические проблемы разработки языков программирования. Стандарты языков программирования.
4. Основные принципы объектно-ориентированного программирования.
5. Преимущества объектно-ориентированного программирования.
6. Понятие класса и объекта класса.
7. Передача параметров в методы класса
8. Конструкторы и деструкторы
9. Принципы объектно-ориентированного программирования. Инкапсуляция.
10. Принципы объектно-ориентированного программирования. Полиморфизм.
11. Принципы объектно-ориентированного программирования. Наследование.
12. Структуры. Объявление. Инициализация.

13. Прямой доступ к элементам структуры.
14. Косвенный доступ к элементам структуры.
15. Массивы структур. Объявление.
16. Массивы структур. Инициализация.
17. Классы. Объявление открытых переменных.
18. Классы. Инициализация открытых переменных.
19. Классы. Объявление закрытых переменных.
20. Классы. Инициализация закрытых переменных.
21. Спецификаторы доступа.
22. Индексаторы. Свойства индексаторов
23. Виртуальные методы и их переопределение
24. Применение абстрактных классов
25. Интерфейсы
26. Ссылочные типы и типы значения
27. Тип enum
28. Арифметические операторы
29. Операторы отношения и логические операторы
30. Управляющие операторы
31. Массивы в C++. Одномерные массивы. Многомерные массивы.

Ступенчатые массивы.

32. Строки. Постоянство строк
33. Основы работы платформы. NET. Понятие единой среды выполнения. Принципы организации выполнения процессов в .NET
34. Пространство имен
35. Сборки
36. Тип System.CG
37. Финализация объектов
38. Создание высвобождаемых объектов
39. Основы обработки исключительных ситуаций
40. Применение пары ключевых слов try и catch
41. Генерирование исключений вручную
42. Градиентные методы
43. Использование блока finally
44. Ключевые слова checked и unchecked
45. Реализация класса - обобщения
46. Интерфейсы необобщенных коллекций
47. Интерфейс ICollection
48. Интерфейс IList
49. Интерфейс IDictionary
50. Простой запрос
51. Общая форма запроса
52. Сортировка результатов запроса с помощью оператора orderby
53. Операторы group, into, let, join
54. Методы запроса

## 55. Формирование запроса с помощью методов запроса.

### 7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет и экзамен проводятся по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» («не зачтено») ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» («зачтено») ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» («зачтено») ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» («зачтено») ставится, если студент набрал от 16 до 20 баллов.)

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Основы языка программирования C/ C++	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовой работе
2	Основы структурного программирования	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовой работе
3	Объектно-ориентированное программирование на языке C++	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовой работе
4	Стандартная библиотека шаблонов в языке программирования C++	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовой работе
5	Основы языка программирования C#	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовой работе
6	Платформа .NET. основные	ОПК-7	Тест, защита

	понятия и принцип работы		лабораторных работ, защита реферата, требования к курсовой работе
--	--------------------------	--	---

### **7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

Основная литература:

1. Карпеев, Д.О. Объектно-ориентированное программирование на языке C++ [Электронный ресурс]: Учеб. пособие / Д. О. Карпеев, В. А. Транин. - Электрон. текстовые, граф. дан. (630 Кб). - Воронеж: ГОУВПО "Воронежский государственный технический университет", 2011. – 1 файл. – 30-00.

2. Молдованова, О.В. Языки программирования и методы трансляции [Электронный ресурс]: учебное пособие/ Молдованова О.В. — Электрон. текстовые данные. — Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2012. – 134 с. – Режим доступа: <http://www.iprbookshop.ru/54809.html>. — ЭБС «IPRbooks».

3. Стенли Липпман Язык программирования C++ [Электронный ресурс]: полное руководство/ Стенли Липпман, Жози Лажойе – Электрон. текстовые данные. - Саратов: Профобразование, 2014. - 1104 с. – Режим доступа: <http://www.iprbookshop.ru/63964.html>. — ЭБС «IPRbooks».

4. Павловская, Т. А. Программирование на языке высокого уровня C# : учебное пособие / Т. А. Павловская. – 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. – 245 с. – Режим доступа: <https://www.iprbookshop.ru/102051.html>. — ЭБС «IPRbooks».

Дополнительная литература:

1. Основы объектно-ориентированного программирования [Электронный ресурс]: учеб. пособие / А. Г. Остапенко [и др.]. - Воронеж: ВГТУ, 2001. - 110 с. - 20.00.

2. Меньшиков, Ф. Олимпиадные задачи по программированию / Ф. Меньшиков. - СПб.: Питер, 2006. - 315 с.: ил. - ISBN 5-469-00765-0: 175-00.

3. Методические указания и задания на курсовую работу по дисциплине Теория языков программирования [Электронный ресурс]/ – Электрон. текстовые данные. – М.: Московский технический университет связи и информатики, 2016. – 20 с. – Режим доступа: <http://www.iprbookshop.ru/61767.html>. — ЭБС «IPRbooks».

**8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

1. Среда разработки MS Visual Studio
2. Среда разработки IntelliJ IDEA
3. Система контроля версий Git

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

Дисплейный класс, оснащенный компьютерными программами для проведения лабораторного практикума.

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Языки программирования» читаются лекции, проводятся лабораторные работы, выполняется курсовая работа.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Методика выполнения курсовой работы изложена в

учебно-методическом пособии. Выполнять этапы курсовой работы должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсовой работы, защитой курсовой работы.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none"> <li>- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;</li> <li>- выполнение домашних заданий и расчетов;</li> <li>- работа над темами для самостоятельного изучения;</li> <li>- участие в работе студенческих научных конференций, олимпиад;</li> <li>- подготовка к промежуточной аттестации.</li> </ul>
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.