


**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета  Гусев П.Ю.

«31» августа 2021 г.

**РАБОЧАЯ ПРОГРАММА**

дисциплины

«Языки программирования»

**Специальность** 10.05.01 Компьютерная безопасность

**Специализация** специализация № 4 "Безопасность компьютерных систем и сетей (связь, информационные и коммуникационные технологии)"

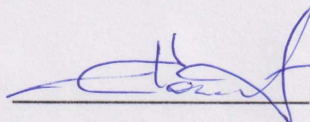
**Квалификация выпускника** специалист по защите информации

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


**Форма обучения** очная

**Год начала подготовки** 2021

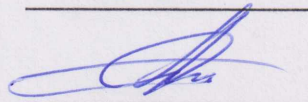
**Автор программы**

 / А.С. Кольцов /

**Заведующий кафедрой  
Систем информационной  
безопасности**

 / А.Г. Остапенко /

**Руководитель ОПОП**

 / А.Г. Остапенко /

Воронеж 2021

## 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 основных принципов программирования на языках высокого уровня;
- изучение базовых конструкций структурного программирования;
- изучение объектно-ориентированного программирования;
- изучение основные алгоритмов обработки массивов, ссылочных структур;
- изучение современных технологий 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 языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ;

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-7	<p>Знать</p> <ul style="list-style-type: none"><li>– общие принципы построения, области и особенности применения языков программирования высокого уровня;</li><li>– язык программирования высокого уровня (объектно-ориентированное программирование).</li></ul> <p>Уметь</p> <ul style="list-style-type: none"><li>– работать с интегрированной средой разработки программного обеспечения;</li><li>– разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач.</li></ul>

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Языки программирования» составляет 15 з.е.

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

Виды учебной работы	Всего часов	Семестры		
		2	3	4
<b>Аудиторные занятия (всего)</b>	216	72	72	72
В том числе:				
Лекции	108	36	36	36
Лабораторные работы (ЛР)	108	36	36	36
<b>Самостоятельная работа</b>	288	144	18	126
<b>Курсовой проект</b>	+		+	
Часы на контроль	36	-	-	36
Виды промежуточной аттестации - экзамен, зачет с оценкой	+	+	+	+
Общая трудоемкость: академические часы	540	216	90	234
зач.ед.	15	6	2.5	6.5

#### 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/	18	18	48	84
2	Основы структурного программирования	Указатели и ссылки. Операции с указателями. Организация работы с динамической памятью. Массивы. Одномерные массивы. 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.	18	18	48	84

		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.				
3	Объектно-ориентированное программирование на языке C++	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. Поточные 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.	18	18	48	84
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.	18	18	48	84
5	Основы языка программирования C#	Синтаксис языка C#. Система типов данных в языке C#. Простые и составные операторы языка C#. Инкапсуляция, наследований полиморфизм в языке C#	18	18	48	84
6	Платформа .NET. основные понятия и принцип работы	Основы работы платформы. NET. Понятие единой среды выполнения. Принципы организации выполнения процессов в .NET. Пространство имен. Сборки. Видимости переменных. Оператор new и сборка мусора. Исключения. генерация и обработка. Основные классы коллекций объектов. Работа с криптографическими классами.	18	18	48	84
<b>Итого</b>			<b>108</b>	<b>108</b>	<b>288</b>	<b>504</b>

## 5.2 Перечень лабораторных работ

1. Вычисление математических выражений.
2. Программирование алгоритмов с разветвляющейся структурой.
3. Циклы.
4. Обработка одномерных массивов.

5. Обработка двумерных массивов.
  6. Работа с динамическими массивами.
  7. Работа с текстовыми файлами.
  8. Создание структурных типов данных.
  9. Функции. Рекурсивные функции.
  10. Классы и объекты
  11. Конструкторы и деструкторы
  12. Статические и константные члены класса
  13. Создание классов с использованием наследования
  14. Виртуальные методы и полиморфизм
  15. Обработка исключений
  16. Ассоциация и зависимость
  17. Стандартные методы приведения типов
  18. Шаблоны. Стандартная библиотека шаблонов STL
  19. Основные конструкции языка C#. Система типов языка C#.
- Выражения и операторы. Массивы.
20. Основные принципы и этапы ООП языка C#. Классы и объекты в C#.
- Элементы класса. Поля и методы. Свойства объектов.
21. Наследование в C#.
  22. Виртуальные и динамические методы. Полиморфизм в C#
  23. Введение в разработку Windows-форм в C#
  24. Работа с элементами управления форм Windows в C#
  25. Использование окон диалога в формах Windows
  26. Основы создания графических приложений в системе Windows с использованием технологий Windows Forms
  27. Подключение к базе данных в C#.

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

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

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

Варианты заданий:

- 1) Разработка программы учета больных вирусом COVID-19
- 2) Разработка программы расселения детей в лагере
- 3) Разработка программы расписания движения самолетов
- 4) Разработка программы продажи билетов на самолёт
- 5) Разработка программы учета переговоров работников такси
- 6) Разработка программы расчета заработной платы сотрудников Яндекс Такси
- 7) Разработка программы учета проживающих в общежитии
- 8) Разработка программы учета выполненных заказов Яндекс такси

- 9) Разработка программы учета научной деятельности сотрудников
- 10) Разработка программы учета выплат заработной платы работникам приюта
- 11) Разработка программы учета клиентов салона красоты
- 12) Разработка программы ассортимента машин в таксопарке
- 13) Разработка программы подбора материалов для ремонта квартиры
- 14) Разработка программы учета выполнения работы лаборанта
- 15) Разработка программы учета слушателей переподготовки
- 16) Разработка программы учета сведений о музыкальном конкурсе
- 17) Разработка программы учета сведений о пациентах медицинского центра
- 18) Разработка программы продажи железнодорожных билетов
- 19) Разработка программы планирования факультативных учебных дисциплин для студентов
- 20) Разработка программы учета сведений об игроках хоккейной команды
- 21) Разработка программы продажи авиабилетов
- 22) Разработка программы учета автомобилей автопарка
- 23) Разработка программы расчета стипендии
- 24) Разработка программы учета выполнения работы мастера по работе с волосами
- 25) Разработка программы планирования бюджета при ремонте машины
- 26) Разработка программы расчета оплаты лечения
- 27) Разработка программы подбора санатория
- 28) Разработка программы учета продаж билетов в аквапарк
- 29) Разработка программы учета продаж квартир
- 30) Разработка программы автоматизации учета кадров на предприятии

Задачи, решаемые при выполнении курсового проекта:

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

Курсовой проект 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».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-7	Знать – общие принципы построения, области и особенности применения языков программирования высокого уровня; – язык программирования высокого уровня (объектно-ориентированное программирование).	Активное участие в устных опросах на занятиях, защита лабораторных работ, ответы на теоретические и практические вопросы промежуточного контроля	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь – работать с интегрированной средой разработки программного обеспечения; – разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач.	Выполнение лабораторных работ, выполнение этапов курсового проекта.	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, 4 семестре для очной формы обучения по четырехбалльной системе:

«отлично»;  
«хорошо»;  
«удовлетворительно»;  
«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-7	Знать – общие принципы построения, области и особенности применения языков программирования высокого уровня; – язык программирования высокого уровня (объектно-ориентированное программирование).	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь – работать с интегрированной средой разработки программного	Решение стандартных практических задач	Задачи решены в полном объеме и получен	Продемонстрирован верный ход решения всех, но не получен	Продемонстрирован верный ход решения в большинстве	Задачи не решены

	обеспечения; – разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач.		ы верные ответы	верный ответ во всех задачах	е задач	
--	--	--	--------------------	------------------------------------	---------	--

## 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. Класс - это:

- а) любой тип данных, определяемый пользователем
- б) тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- в) структура, для которой в программе имеются функции работы с ней

2. Членами класса могут быть

- а) как переменные, так и функции, могут быть объявлены как private и как public
- б) только переменные, объявленные как private
- в) только функции, объявленные как private
- г) только переменные и функции, объявленные как private
- д) только переменные и функции, объявленные как public

3. Что называется конструктором?

- а) метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- б) метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- в) метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- г) метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

4. Объект - это

- а) переменная, содержащая указатель на класс
- б) экземпляр класса
- в) класс, который содержит в себе данные и методы их обработки

5. Отметьте правильные утверждения

- а) конструкторы класса не наследуются
- б) конструкторов класса может быть несколько, их синтаксис



определяется программистом

- в) конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций
- г) конструктор возвращает указатель на объект
- д) конструктор не возвращает значение

6. Что называется деструктором?

- а) метод, который уничтожает объект
- б) метод, который удаляет объект
- в) метод, который освобождает память, занимаемую объектом
- г) системная функция, которая освобождает память, занимаемую объектом

7. Выберите правильные утверждения

- а) у конструктора могут быть параметры
- б) конструктор наследуется, но должен быть перегружен
- в) конструктор должен явно вызываться всегда перед объявлением объекта
- г) конструктор вызывается автоматически при объявлении объекта
- д) объявление каждого класса должно содержать свой конструктор
- е) если конструктор не создан, компилятор создаст его автоматически

8. Выберите правильные утверждения

- а) деструктор - это метод класса, применяемый для удаления объекта
- б) деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом
- в) деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом
- г) деструктор не наследуется
- д) деструктор наследуется, но должен быть перегружен

9. Что называется наследованием?

- а) это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
- б) это механизм переопределения методов базового класса
- в) это механизм, посредством которого производный класс получает все поля базового класса
- г) это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

10. Выберите правильное объявление производного класса

- а) `class MoreDetails:: Details;`
- б) `class MoreDetails: public class Details;`
- в) `class MoreDetails: public Details;`

г) `class MoreDetails: class(Details);`

11. Выберите правильные утверждения:

а) если элементы класса объявлены как `private`, то они доступны только наследникам класса, но не внешним функциям

б) если элементы класса объявлены как `private`, то они недоступны ни наследникам класса, ни внешним функциям

в) если элементы объявлены как `public`, то они доступны наследникам класса, но не внешним функциям

г) если элементы объявлены как `public`, то они доступны и наследникам класса, и внешним функциям

12. Возможность и способ обращения производного класса к элементам базового определяется

а) ключами доступа: `private`, `public`, `protected` в теле производного класса

б) только ключом доступа `protected` в заголовке объявления производного класса

в) ключами доступа: `private`, `public`, `protected` в заголовке объявления производного класса

г) ключами доступа: `private`, `public`, `protected` в теле базового класса

13. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового

а) ключ доступа - `public`; в базовом классе: `private`; права доступа в производном классе - `protected`

б) ключ доступа - любой; в базовом классе: `private`; права доступа в производном классе - нет прав

в) ключ доступа - `protected` или `public`; в базовом классе: `protected`; права доступа в производном классе - `protected`

г) ключ доступа - `private`; в базовом классе: `public`; права доступа в производном классе - `public`

д) ключ доступа – любой; в базовом классе: `public`; права доступа в производном классе – такие же, как ключ доступа

14. Дружественная функция - это

а) функция другого класса, среди аргументов которой есть элементы данного класса

б) функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;

в) функция, являющаяся членом класса и объявленная с атрибутом `friend`;

г) функция, которая в другом классе объявлена как дружественная

данному.

### 7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какой тип переменной используется в коде: `int a = 5;`
  - а) Знаковое 8-бит целое
  - б) Знаковое 64-бит целое
  - в) Знаковое 32-бит целое
  
2. Что делает оператор «%»
  - а) Возвращает процент от суммы
  - б) Возвращает остаток от деления
  - в) Возвращает тригонометрическую функцию
  - г) Ни чего из выше перечисленного.
  
3. Что сделает программа выполнив следующий код:  
`Console.WriteLine(«Hello, World!»);`
  - а) Напишет на новой строчке Hello, World! (+)
  - б) Напишет Hello, World!
  - в) Удалит все значения с Hello, World!
  - г) Вырежет слово Hello, World! из всего текста
  
4. Как сделать инкрементацию числа
  - а) ++
  - б) —
  - в) %%
  - г) !=
  
5. Как сделать декрементация числа
  - а) %%
  - б) —
  - в) !=
  - г) ++
  
6. Чему будет равен c, если `int a = 10; int b = 4; int c = a % b;`
  - а) 11
  - б) 2
  - в) 3
  - г) 1
  
7. Чему будет равен c, если `int a = 10; int b = 4; bool c = (a == 10 && b == 4);`
  - а) True
  - б) False
  - в) Null
  - г) 14

8. Чему будет равен c, если `int a = 0; int c = a—;`

- а) Null
- б) -1
- в) 0
- г) 1

9. Чему будет равен c, если `int a = 0; int c = —a;`

- а) Null
- б) -1
- в) 0
- г) 1

10. Чему равен d, если `int a = 0; int b = a++; int c = 0; int d = a + b + c + 3;`

- а) 3
- б) True
- в) False
- г) 4

### 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**

Не предусмотрено учебным планом

#### **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) Механизм ввода/вывода в ООП. Манипуляторы. Формат ввода вывода.

5) Указатели и ссылки. Использование операторов new и delete.

6) Возникновение ООП. Фундаментальные характеристики ООП (инкапсуляция, наследование, полиморфизм).

7) Простые типы данных. Представление в памяти ЭВМ символов, целых и вещественных чисел.

8) Структурированные типы данных. Представление в памяти ЭВМ массивов, структур, битовых полей и т.д.

9) Классы в C++. Передача параметров в методы класса

- 10) Описания, типы и основные свойства конструкторов.
- 11) Описание и свойства деструктора.
- 12) Присвоение объектов. Указатель this.
- 13) Передача объектов функции. Объект в качестве возвращаемого
- 14) значения функции.
- 15) Конструктор копии.
- 16) Статические поля и методы класса.
- 17) Константные поля и методы класса.
- 18) Дружественные функции и классы.
- 19) Конструктор explicit
- 20) Структуры. Объявление. Инициализация.
- 21) Прямой доступ к элементам структуры.
- 22) Косвенный доступ к элементам структуры.
- 23) Массивы структур. Объявление.
- 24) Массивы структур. Инициализация.
- 25) Классы. Объявление открытых переменных.
- 26) Классы. Инициализация открытых переменных.
- 27) Классы. Объявление закрытых переменных.
- 28) Классы. Инициализация закрытых переменных.
- 29) Понятие наследования. Указатель на базовый и производный класс.
- 30) Многоуровневая иерархия классов.
- 31) Множественное наследование.
- 32) Спецификаторы доступа.
- 33) Индексаторы. Свойства индексаторов
- 34) Виртуальные методы и их переопределение
- 35) Применение абстрактных классов
- 36) Интерфейсы
- 37) Ссылочные типы и типы значения
- 38) Тип enum
- 39) Арифметические операторы
- 40) Операторы отношения и логические операторы
- 41) Управляющие операторы
- 42) Массивы в C++. Одномерные массивы. Многомерные массивы.

Ступенчатые массивы.

- 43) Строки. Постоянство строк
- 44) Основы работы платформы. NET. Понятие единой среды выполнения. Принципы организации выполнения процессов в .NET
- 45) Пространство имен
- 46) Виртуальные методы. Ранее и позднее связывание.
- 47) Чисто виртуальные функции, абстрактный класс
- 48) Тип System.CG
- 49) Финализация объектов
- 50) Создание высвобождаемых объектов
- 51) Основы обработки исключительных ситуаций
- 52) Применение пары ключевых слов try и catch

- 53) Генерирование исключений вручную
- 54) Градиентные методы
- 55) Понятие шаблона. Шаблоны функций.
- 56) Понятие шаблона. классов.
- 57) Шаблоны и наследование. Шаблоны и друзья. Шаблоны и статические члены.
- 58) Использование блока finally
- 59) Ключевые слова checked и unchecked
- 60) Реализация класса - обобщения
- 61) Интерфейсы необобщенных коллекций
- 62) Интерфейс ICollection
- 63) Интерфейс IList
- 64) Интерфейс IDictionary
- 65) Простой запрос
- 66) Общая форма запроса
- 67) Сортировка результатов запроса с помощью оператора orderby
- 68) Операторы group, into, let, join
- 69) Методы запроса
- 70) Формирование запроса с помощью методов запроса.
- 71) Стандартные потоки.
- 72) Файловые потоки.
- 73) Строковые потоки.

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

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» ставится в случае, если студент набрал менее 6 баллов.

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

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

4. Оценка «Отлично» 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 курсовому проекту
2	Основы структурного	ОПК-7	Тест, защита



	программирования		лабораторных работ, защита реферата, требования к курсовому проекту
3	Объектно-ориентированное программирование на языке C++	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовому проекту
4	Стандартная библиотека шаблонов в языке программирования C++	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовому проекту
5	Основы языка программирования C#	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовому проекту
6	Платформа .NET. основные понятия и принцип работы	ОПК-7	Тест, защита лабораторных работ, защита реферата, требования к курсовому проекту

### **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 С++ [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 С++ [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 С# : 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 курсовой проект.

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 курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометить 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	Готовиться к промежуточной аттестации следует систематически, в течение

к промежуточ ной аттестации	всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом с оценкой, зачетом с оценкой, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.
--------------------------------------	---