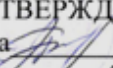


**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

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

УТВЕРЖДАЮ

Декан факультета  С.М. Пасмурнов
«31» августа 2017 г.

РАБОЧАЯ ПРОГРАММА

дисциплины

«Языки программирования»

Специальность 10.05.03 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ
АВТОМАТИЗИРОВАННЫХ СИСТЕМ

Специализация

Квалификация выпускника специалист по защите информации

Нормативный период обучения 5 лет

Форма обучения очная

Год начала подготовки 2016

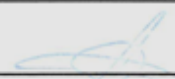
Автор программы

 /Карпеев Д.О./

Заведующий кафедрой
Систем информационной
безопасности

 / А.Г. Остапенко /

Руководитель ОПОП

 / А.Г. Остапенко /

Воронеж 2017

1.ЦЕЛИИЗАДАЧИДИСЦИПЛИНЫ

1.1.Целидисциплины

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.Задачiosoвoениядисциплины

- 1) Изучение стандартов языков программирования.
- 2) Изучение концепций программирования.
- 3) Изучение стандартных библиотек и платформ языков программирования.

2.МЕСТОДИСЦИПЛИНЫВСТРУКТУРЕОПОП

Дисциплина«Языкипрограммирования»относитсякдисциплинамбазово
йчастиблокаБ1.

3.ПЕРЕЧЕНЬПЛАНИРУЕМЫХРЕЗУЛЬТАТОВОБУЧЕНИЯПОДИСЦИ ПЛИНЕ

Процессизучениядисциплины«Языкипрограммирования»направленна
формированиеследующихкомпетенций:

ОПК-5-способностьюприменятьметодынаучныхисследованийвпрофесс
иональнойдеятельности,втомчислевработенадмеждисциплинарнымииинова
ционнымипроектами;

ОПК-6-способностьюприменятьнормативныеправовыеактывпрофессио
нальнойдеятельности;

ПК-1-способностьюосуществлятьпоиск,изучение,обобщениеисистемат
изациюнаучно-техническойинформации,нормативныхиметодическихматериа
ловвсферепрофессиональнойдеятельности,втомчислеанаиностранномязыке;

Компетенция	Результатыобучения,характеризующие сформированностькомпетенции
ОПК-5	Знать требования к структуре исходного кода программного обеспечения
	Уметь реализовывать консольный и графический пользовательский интерфейс программного обеспечения
	Владеть методиками проектирования программного обеспечения
ОПК-6	Знать стандарты языков программирования C, C++, C#
	Уметь применять возможности языков программирования C, C++, C# для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере

ПК-1	информационной безопасности
	Владеть синтаксисом языков программирования C, C++, C#
	Знать концепции и основные конструкции функционального и объектно-ориентированного программирования
	Уметь разрабатывать и реализовывать алгоритмы для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности
	Владеть методами оценки сложности алгоритмов для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Языки программирования» составляет 10 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры	
		2	3
Аудиторные занятия (всего)	162	90	72
В том числе:			
Лекции	54	36	18
Практические занятия (ПЗ)	36	18	18
Лабораторные работы (ЛР)	72	36	36
Самостоятельная работа	162	108	54
Курсовой проект	+		+
Часы на контроль	36	-	36
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость:			
академические часы	360	198	162
зач.ед.	10	5.5	4.5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№	Наименование темы	Содержание раздела	Лекц	Прак	Лаб.	СРС	Всего,
---	-------------------	--------------------	------	------	------	-----	--------

п/п				зан.	зан.		час
1	Основы языка программирования С	Общие принципы построения и использования языков Программирования. Современные интегрированные среды разработки программного обеспечения. Синтаксис языка С. Простые и структурные типы данных в языке С. Простые и составные операторы языка С. Функциональное программирование	10	6	12	26	54
2	Основы языка программирования С++	Синтаксис языка С. Система типов данных в языке С++. Простые и составные операторы языка С++. Массивы. Функции и рекурсия. Работа со строками. Работа с файлами. Обработка исключений	10	6	12	26	54
3	Объектно-ориентированное-программирование на языке С++	Концепция ООП. Классы и объекты класса. Механизм инкапсуляции. Особые случаи организации доступа к компонентам класса. Механизм наследования. Полиморфизм. Виртуальное наследование. Виртуальные методы. Абстрактные методы и классы. Интерфейсы. Шаблоны.	10	6	12	26	54
4	Стандартная библиотека шаблонов в языке программирования С++	Контейнеры. Итераторы. Алгоритмы. Адаптеры. Функциональные объекты	8	6	12	28	54
5	Основы языка программирования С#	Синтаксис языка С#. Система типов данных в языке С#. Простые и составные операторы языка С++. Инкапсуляция, наследований полиморфизм в языке С#	8	6	12	28	54
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.	8	6	12	28	54
Итого			54	36	72	162	324

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.
10. Рекурсивные функции.
11. Проектирование и создание классов.

12. Переопределение операций.
13. Построение иерархии классов.
14. Простой и сложный полиморфизм.
15. Абстрактные и дружественные функции и классы.
16. Работа с последовательными контейнерами.
17. Работа с ассоциативными контейнерами.
18. Применение итераторов.
19. Применение алгоритмов.
20. Создание функциональных объектов.
21. Применение условного оператора и операторов цикла.
22. Обработка одномерных и двумерных массивов.
23. Проектирование иерархии классов, полиморфизм.
24. Шаблоны классов.
25. Работа с классами, реализующими хеширование.
26. Работа с классами, реализующими симметричное шифрование.
27. Работа с классами, реализующими асимметричное шифрование.
28. 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 курсового проекта в 4 семестре для очной формы обучения.

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

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 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 шкалы оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются последующей системой:

«аттестован»;

«неаттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Неаттестован
-------------	---	---------------------	------------	--------------

ОПК-5	Знать требования к структуре исходного кода программного обеспечения	Знает требования к структуре исходного кода программного обеспечения	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь реализовывать консольный и графический пользовательский интерфейс программного обеспечения	Умеет реализовывает консольный и графический пользовательский интерфейс программного обеспечения	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть методиками проектирования программного обеспечения	Владеет методиками проектирования программного обеспечения	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	Знать стандарты языков программирования С, С++, С#	Знает стандарты языков программирования С, С++, С#	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь применять возможности языков программирования С, С++, С# для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Умеет применять возможности языков программирования С, С++, С# для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть синтаксисом языков программирования С, С++, С#	Владеет синтаксисом языков программирования С, С++, С#	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-1	Знать концепции и основные конструкции функционального и объектно-ориентированного программирования	Знает концепции и основные конструкции функционального и объектно-ориентированного программирования	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Уметь разрабатывать и реализовывать алгоритмы для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Умеет разрабатывает и реализовывает алгоритмы для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	Владеть методами оценки сложности алгоритмов для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Владеет методами оценки сложности алгоритмов для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	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»

«незачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Незачтено
ОПК-5	Знать требования к структуре исходного кода программного обеспечения	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь реализовывать консольный и графический пользовательский интерфейс программного обеспечения	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть методиками проектирования программного обеспечения	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	Знать стандарты языков программирования C, C++, C#	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь применять возможности языков программирования C, C++, C# для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть синтаксисом языков программирования C, C++, C#	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-1	Знать концепции и основные конструкции функционального и объектно-ориентированного программирования	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	Уметь разрабатывать и реализовывать алгоритмы для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	Владеть методами оценки сложности алгоритмов для решения профессиональных, исследовательских и прикладных задач, в	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

	том числе в сфере информационной безопасности			
--	---	--	--	--

или

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ОПК-5	Знать требования к структуре исходного кода программного обеспечения	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь реализовывать консольный и графический пользовательский интерфейс программного обеспечения	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 методиками проектирования программного обеспечения	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
ОПК-6	Знать стандарты языков программирования C, C++, C#	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь применять возможности языков программирования C, C++, C# для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	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 синтаксисом языков программирования C, C++, C#	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

ПК-1	Знать концепции и основные конструкции функционального и объектно-ориентированного программирования	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70-80%	В тесте менее 70% правильных ответов
	Уметь разрабатывать и реализовывать алгоритмы для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	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 методами оценки сложности алгоритмов для решения профессиональных, исследовательских и прикладных задач, в том числе в сфере информационной безопасности	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 (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что такое ООП:

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

2. Visual Studio 2013/2015/2017 это:

- а) язык программирования
- б) среда разработки приложений на различных языках программирования, включая C++
- в) добавление стандарта языка C++

3. Выберите недопустимые имена названия переменных:

- а) `_zarplata`
- б) `IopataSovkovaya_SSSR`
- в) `myDog`
- г) `КотПушокCrazy`
- д) `j5432_123stepanIvanovich_`
- е) `15Count`
- ж) `_1`
- з) `double`

4. 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

5. Приведён фрагмент кода:

```
signed char param1 = 0;
unsigned short param2 = 0;
long param3 = 0;
param1--;
param2--;
param3--;
std::cout << (int) param1 << param2 << param3;
```

Напишите через запятую, три значения, которые выведутся на экран (известно, что sizeof(unsigned char) = 1, sizeof(signed short) = 2, sizeof(long) = 4).

6. В какой из перечисленных строк компилятор выдаст ошибку:

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

7. Чему будет равна переменная x:

```
int temp = 0;
int count = 0;
for (temp = 10; temp < 20; temp++)
{
    count++;
}
int x = (temp/count)*0.8;
```

8. Чему будут равны переменные x -? y-? z-?:

```
int temp = 13;
double x = 0.0;
double y = 0.0;
int z = 0;
x = temp / 2;
y = temp / 4.0;
z = temp / 2.0
```

9. Какое значение вернёт функция:

```
int randomXX()
{
    int value = 5;
    return value + 3;
    value++;
    return value;
}
```

10. 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:

- A) `A[i]:=strtoint(datagridview1[0,i-1].Value.ToString);`
- Б) `a(i) = Val(DataGridView1.Item(0, i).Value);`
- B) `a[i]:=strtoint(stringgrid1.cells[i,1]);`
- Г) `a[i] = Convert.ToSingle(dataGridView1[0, i].Value).`

10. Датчик случайных чисел описывается командами:

- A) `srand(time(NULL)); a= rand() % 10;`
- Б) `a= int(random()*10);`
- B) `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

Этапы развития технологий программирования

2. Введение в ООП
3. Понятие класса и объекта класса. Создание объектов. Данные и методы класса
4. Передача параметров в методы класса
5. Конструкторы и деструкторы
6. Инкапсуляция
7. Индексаторы
8. Свойства индексаторов
9. Основное понятие наследования. Наследование методов и данных
10. Виртуальные методы и их переопределение

11. Применение абстрактных классов
12. Интерфейсы
13. Ссылочные типы и типы значения
14. Тип enum
15. Типы структур
16. Арифметические операторы
17. Операторы отношения и логические операторы
18. Управляющие операторы
19. Массивы в C++. Одномерные массивы. Многомерные массивы.

Ступенчатые массивы.

20. Строки. Постоянство строк

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. Этапы развития технологий программирования
 2. Введение в ООП
 3. Понятие класса и объекта класса. Создание объектов. Данные и методы класса
 4. Передача параметров в методы класса
 5. Конструкторы и деструкторы
 6. Инкапсуляция
 7. Индексаторы
 8. Свойства индексаторов
 9. Основное понятие наследования. Наследование методов и данных
 10. Виртуальные методы и их переопределение
 11. Применение абстрактных классов
 12. Интерфейсы
 13. Ссылочные типы и типы значения
 14. Тип enum
 15. Типы структур
 16. Арифметические операторы
 17. Операторы отношения и логические операторы
 18. Управляющие операторы
 19. Массивы в C++. Одномерные массивы. Многомерные массивы.
- Ступенчатые массивы.
20. Строки. Постоянство строк
 21. Основы работы платформы. NET. Понятие единой среды выполнения. Принципы организации выполнения процессов в .NET
 22. Пространство имен
 23. Сборки
 24. Тип System.CG
 25. Финализация объектов
 26. Создание высвобождаемых объектов
 27. Основы обработки исключительных ситуаций
 28. Применение пары ключевых слов try и catch
 29. Генерирование исключений вручную

30. Градиентные методы
31. Использование блока finally
32. Ключевые слова checked и unchecked
33. Реализация класса – обобщения
34. Интерфейсы необобщенных коллекций
35. Интерфейс ICollection
36. Интерфейс IList
37. Интерфейс IDictionary
38. Простой запрос
39. Общая форма запроса
40. Сортировка результатов запроса с помощью оператора orderby
41. Операторы group, into, let, join
42. Методы запроса
43. Формирование запроса с помощью методов запроса.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов по задаче. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, за задачу оценивается в 10 баллов (5 баллов за 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	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту.
2	Основы языка программирования C++	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту.
3	Объектно-ориентированное программирование на языке C++	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, защита лабораторных работ,

			защита реферата, требования к курсовому проекту.
4	Стандартная библиотека шаблонов в языке программирования C++	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту.
5	Основы языка программирования C#	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, защита лабораторных работ, защита реферата, требования к курсовому проекту.
6	Платформа .NET. основные понятия и принцип работы	ОПК-5, ОПК-6, ПК -1	Тест, контрольная работа, 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 методике 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 методике 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 методике выставления оценки при проведении промежуточной аттестации.

Защита курсовой работы, курсового проекта или отчета по всем видам практики 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. Языки программирования С и С++ [Электронный ресурс] . - 2001. - Компьют. файл (на 1 дискете). - 20.00.

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

Дополнительная литература:

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

2. Методические рекомендации к выполнению лабораторных работ по дисциплине "Языки программирования" для студентов специальностей 090301 "Компьютерная безопасность", 090303 "Информационная безопасность автоматизированных систем" очной формы обучения [Электронный ресурс] / Каф. систем информационной безопасности; Сост. Д. О. Карпеев. - Электрон. текстовые, граф. дан. (73 Кб). - Воронеж : ФГБОУ ВПО "Воронежский государственный технический университет", 2012. - 1 файл. - 00-00.

3. Методические указания к самостоятельным работам по дисциплине «Языки программирования» для студентов специальностей 090301 «Компьютерная безопасность», 090302 «Информационная безопасность телекоммуникационных систем», 090303 «Информационная безопасность автоматизированных систем» очной формы обучения [Электронный ресурс] / Каф. систем информационной безопасности; Сост. Д. О. Карпеев. - Электрон. текстовые, граф. дан. (563 Кб). - Воронеж : ФГБОУ ВПО "Воронежский государственный технический университет", 2015. - 1 файл. - 00-00.

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

Наличие среды разработки программного обеспечения на языках C++ и C#.

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Компьютерных класс с количеством персональных компьютеров из расчета 1 персональный компьютер на 2 обучающихся.

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

По дисциплине «Языки программирования» читаются лекции, проводятся практические занятия и лабораторные работы, выполняется курсовой проект.

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

Практические занятия направлены на приобретение практических навыков разработки программного обеспечения. Занятия проводятся путем решения конкретных задач в аудитории.

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 в свое время 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; 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.
Практическое занятие	Конспектирование рекомендуемых источников. Работа с конспектом лекций, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы. Прослушивание аудио- и видеозаписей по заданной теме, выполнение расчетно-графических заданий, решение задач по алгоритму.
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: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;

	- выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
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.