

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Наумова Наталия Александровна
Должность: Ректор
Дата подписания: 24.10.2024 14:21:41
Уникальный программный ключ:
6b5279da4e034bff679172803da5b7b559fc69e2

МИНИСТЕРСТВО ОБРАЗОВАНИЯ МОСКОВСКОЙ ОБЛАСТИ
Государственное образовательное учреждение высшего образования Московской области
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ОБЛАСТНОЙ УНИВЕРСИТЕТ
(МГОУ)

Физико-математический факультет
Кафедра вычислительной математики
и методики преподавания информатики

УТВЕРЖДЕН на заседании кафедры
Протокол от «20» мая 2020 г., № 10
Зав. кафедрой Шевчук М.В. / Шевчук М.В./

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине
Языки и методы программирования

Направление подготовки
44.03.01 Педагогическое образование

Профиль
Информатика

Мытищи
2020

Автор - составитель:

Грамаков Дмитрий Анатольевич
доцент, кандидат педагогических наук, доцент кафедры вычислительной математики и
методики преподавания информатики

Рабочая программа дисциплины «Языки и методы программирования» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 44.03.01 Педагогическое образование профиль «Информатика» утвержденная приказом МИНОБРНАУКИ РОССИИ от 22.02.18 № 121.

Дисциплина входит в обязательную часть блока Б1 «Дисциплины (модули)» и является обязательной для изучения.

Год начала подготовки 2020

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Изучение дисциплины «Языки и методы программирования» позволяет сформировать у бакалавров следующие компетенции.

Код и наименование компетенции	Этапы формирования
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	1.Работа на учебных занятиях. 2. Самостоятельная работа.

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

Оцениваемые компетенции	Уровень сформированности	Этап формирования	Описание показателей	Критерии оценивания	Шкала оценивания
УК-1	Пороговый	1.Работа на учебных занятиях. 2.Самостоятельная работа.	Знает и понимает: • закономерности поиска, критического анализа и синтеза информации для решения широкого круга задач, связанных с программированием; • особенности системного подхода для решения широкого спектра задач в области программирования; Умеет: • осуществлять поиск, критический анализ и синтез методов для задач обучения в области программирования; • применять системный подход для решения задач в области программирования.	Посещение, тестовые задания, задания по сам работе, домашние задания, экзамен	41-60
	Продвинутый	1.Работа на учебных занятиях. 2.Самостоятельная работа.	Знает и понимает: • закономерности поиска, критического анализа и синтеза информации для решения широкого круга задач, связанных с программированием; • особенности системного подхода для решения	Посещение, тестовые задания, задания по сам работе, домашние задания,	61-100

			широкого спектра задач в области программирования; Умеет: • осуществлять поиск, критический анализ и синтез методов для задач обучения в области программирования; • применять системный подход для решения задач в области программирования. <i>Владеет (навыками и/или опытом деятельности):</i> • навыками осуществления поиска, критического анализа и синтеза информации для решения широкого круга задач обучения программированию; • навыками применения системного подхода для решения широкого круга задач в области программирования и обучения программированию.	экзамен	
--	--	--	--	---------	--

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Тестовые задания для текущего контроля и промежуточного контроля:

- 1. Вы пишете код для класса с именем Book. Вы должны быть в состоянии получить список всех книг, отсортированных по фамилии автора. Вам нужно написать код, чтобы определить такое поведение класса. Какой из следующих элементов класса следует использовать?**

- a) метод
- b) свойство
- c) событие
- d) подпрограмма

- 2. Вы разрабатываете код для метода, который вычисляет скидку для проданных товаров. Имя метода CalculateDiscount. Этот метод определяет переменную percentValue типа double. Вам нужно убедиться, что переменная percentValue доступна только в методе CalculateDiscount. Какой модификатор доступа следует использовать при определении переменной percentValue?**

- a) private
- b) protected
- c) internal
- d) public

- 3. Вы пишете код для бизнес-приложений с использованием C#. Вы пишете следующий оператор объявления массива:**
`int[] numbers = { 1, 2, 3, 4, 5 };`

Теперь вам нужно получить доступ ко второму элементу в этом массиве. Какие из следующих выражений следует использовать?

- a) numbers[0]
- b) numbers[1]
- c) numbers[2]
- d) numbers[3]

- 4. Вы разрабатываете программы на C# и написали следующий код:**

```
int x = 10;  
int y = x++;  
int z = y++;
```

Какое будет значение переменной z, после того, как все вышеописанные операторы выполнятся?

- a) 10
- b) 11
- c) 12
- d) 13

5. Вы пишете программы на C#, которые используются для итерации по коллекциям (массивам и спискам). Вы должны убедиться, что вы обрабатываете каждый элемент коллекции один раз. Вы также должны убедиться, что ваш код легко читать и отлаживать. Какое из следующих выражений на C# обеспечит лучшее решение для этого требования?

- a) while
- b) for
- c) foreach
- d) do-while

6. Вы разрабатываете программы на C# и написали рекурсивный метод для вычисления факториала числа. Какой из следующий фрагментов кода следует использовать, чтобы генерировать правильные результаты?

- a)

```
public static int Factorial(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return n * Factorial(n - 1);
    }
}
```
- b)

```
public static int Factorial(int n)
{
    if (n == 0)
    {
        return 1;
    }
    else
    {
        return (n - 1) * Factorial(n);
    }
}
```
- c)

```
public static int Factorial(int n)
{
    if (n == 0)
    {
        return n;
    }
    else
    {
        return Factorial(n - 1);
    }
}
```
- d)

```
public static int Factorial(int n)
{
    return n * Factorial(n - 1);
}
```

7. Вы пишете программу на C#, в которой необходимо выполнить итерацию фиксированное количество раз. Вам нужно убедиться, что ваш код легко

понимать и поддерживать, даже когда тело цикла содержит сложный код. Какие из следующих утверждений C# обеспечивают лучшее решение для этого требования

- a) while
- b) for
- c) foreach
- d) do-while

8. Вам необходимо предоставить сложные многоходовые ветвление в программе C#. Вы должны убедиться, что ваш код легко читать и понимать. Какое из следующих выражений на C# вы должны использовать?

- a) case
- b) break
- c) if-else
- d) switch

9. Вы пишете программы на C# и нужно выбрать соответствующую структуру повторения для вашего требования: «Проверка в условии завершения должна выполняться в нижней части цикла, а не в верхней». Какой оператор цикла следует использовать?

- a) Оператор while
- b) Оператор for
- c) Оператор foreach
- d) Оператор do-while

10. Вы разрабатываете программы на C# и написали следующий код:

```
int x = 10;  
int y = ++x;  
int z = --y;
```

Какое будет значение переменной z после того, как все вышеописанные операторы выполнятся?

- a) 10
- b) 11
- c) 12
- d) 13

11. Какие из следующих утверждений верны для фрагмента кода на # C ?

```
class Sample  
{  
    private int i;  
    public Single j;  
    private void DisplayData()  
    {  
        Console.WriteLine(i + " " + j);  
    }  
}
```

```

public void ShowData()
{
    Console.WriteLine(i + " " + j);
}
}

```

- a) j не может быть объявлен как public
- b) DisplayData() не может быть объявлен как private.
- c) DisplayData() не может получить доступ к j.
- d) ShowData() не может получить доступ к i
- e) Ошибки нет в этом классе.

12. Что из следующего используется при описании 8-байтового целого числа?

- a) char
- b) long
- c) short
- d) byte
- e) integer

13. Каков размер переменных, описанных с помощью типа decimal?

- a) 4 байт
- b) 8 байт
- c) 16 байт
- d) 32 байт

14. Вы анализируете программу на C#, содержащую следующий класс

```

public class Rectangle
{
    public double Length {get; set;}
    public double Width { get; set; }
}

```

Программа выполняет следующий код в рамках метода Main

```

Rectangle r1, r2;
r1 = new Rectangle(Length = 10.0, Width = 20.0);
r2 = r1;
r2.Length = 30;
Console.WriteLine(r1.Length);

```

Каким будет результат, если этот код выполнится?

- a) 10
- b) 20
- c) 30
- d) 40

15. Предположим, что вы пишете код для класса с именем Product. Вам нужно убедиться, что члены данного класса инициализированы их правильными значениями, как только вы создаете объект класса Product. Код инициализации всегда должен быть выполнен. Что делать?

- a) Создать статический метод в класс Product для инициализации членов данных.

- b) Создать конструктор в классе Product для инициализации членов данных
- c) Создать статическое свойство в классе Product для инициализации членов данных
- d) Создать событие в классе Product для инициализации членов данных.

16. Создан класс с именем GeoShape. Определен метод Area в этом классе, он вычисляет площадь геометрической фигуры. Вы хотите, чтобы производные классы GeoShape могли заменить его функциональность для поддержки расчетов дополнительных геометрических фигур. Когда метод Area вызывается в объекте GeoShape, площадь должна быть рассчитана, основываясь на используемом в текущий момент объекте GeoShape. Ключевое слово, которое вы должны использовать с определением площади методом в GeoShape классе?

- a) abstract
- b) virtual
- c) new
- d) overrides

17. Вы разрабатываете программы на C# и написали следующий код:

```
int x = 10;  
int y = ++x;  
int z = y++;
```

Какое будет значение переменной z после того, как все вышеописанные операторы выполнятся?

- a) 10
- b) 11
- c) 12
- d) 13

18. 23. Вы разрабатываете программы на C# и написали следующий код:

```
int count = 0;  
while (count < 5)  
{  
    if (count == 3)  
        break;  
    count++;  
}
```

Сколько раз будет выполнен цикл while?

- a) 5
- b) 4
- c) 3
- d) 2

19. Вы пишете программы на C# и написали следующий метод

```

public static void TestSwitch(int op1, int op2, char opr)
{
    int result;
    switch (opr)
    {
        case '+':
            result = op1 + op2;
        case '-':
            result = op1 - op2;
        case '*':
            result = op1 * op2;
        case '/':
            result = op1 / op2;
        default:
            Console.WriteLine("Неизвестный оператор");
            return;
    }
    Console.WriteLine("Результат: {0}", result);
    return;
}

```

Однако, когда вы компилируете этот код, вы получаете следующее сообщение об ошибке:
Управление не проваливается из одной метки инструкции case к другой

Как нужно модифицировать код, чтобы убедиться, что он успешно компилируется?

- a) после каждого case добавьте следующую строку кода: `break;`
- b) после каждого case добавьте следующую строку кода: `continue;`
- c) после каждого case добавьте следующую строку кода: `goto default;`
- d) после каждого case добавьте следующую строку кода: `return;`

20. Вы пишете метод с именем `PrintReport`, который не возвращает значение в вызывающий код. Какое ключевое слово следует использовать в вашем объявлении метода?

- a) `void`
- b) `private`
- c) `int`
- d) `string`

21. Вы разрабатываете программы на C# и написали следующий код:

```

int i = 6;
do
{
    if (i == 3)
        break;
    Console.WriteLine("Значение i = {0}", i);
    i++;
}
while (i <= 5);

```

Сколько раз будет выполнен цикл `while`?

- a) 0
- b) 1
- c) 2
- d) 3

22. Вы разрабатываете программу на C# и написали следующую строку кода:

```
int x = 6 + 4 * 4 / 2 - 1;
```

Каково будет значение переменной x после того, как этот оператор выполнится?

- a) 19
- b) 13
- c) 20
- d) 14

23. Какие из следующих утверждений верны?

1. Члены экземпляра класса доступны только через объект этого класса.
2. Класс может содержать только данные экземпляра и экземпляр функции.
3. Все объекты, созданные из класса будут занимать равное количество байт в памяти.
4. Класс является проектом или шаблоном, согласно которого создаются объекты.

- a) 1, 3, 4
- b) 2, 4
- c) 3, 4
- d) 2, 4
- e) Ничего из вышеперечисленного

24. Что из нижеперечисленного не является целым числом?

- a) char
- b) byte
- c) integer
- d) short
- e) long

25. Какие из следующих утверждений верны о типах данных в C#?

1. Каждый тип данных является или типом значения или ссылочным типом.
2. Типы значений всегда создаются в куче.
3. Ссылочные типы всегда создаются в стеке.
4. Сопоставление каждого типа значения в языке C# с типами в общей системе типов облегчает взаимодействие в платформе .NET.
5. Каждый ссылочный тип имеет сопоставленный тип в общей системе типов.

- a) 1, 3
- b) 2, 5
- c) 1, 4
- d) 3, 4

26. Вы создаете новый класс с именем Sphere, производный от класса Shape. Класс Shape имеет следующий код:

```
class Shape
```

```

{
    public virtual void Area()
    {
        // additional code...
    }
}

```

Метод Area в классе Sphere должен предоставлять новые функциональные возможности, но также скрыть реализацию метода Area класса Shape. Какой сегмент кода следует использовать для этого?

a)

```
class Sphere : Shape
{
    public override void Area()
    {
        // additional code ...
    }
}
```

b)

```
class Sphere : Shape
{
    public new void Area()
    {
        // additional code ...
    }
}
```

c)

```
class Sphere : Shape
{
    public virtual void Area()
    {
        // additional code ...
    }
}
```

d)

```
class Sphere : Shape
{
    public static void Area()
    {
        // additional code ...
    }
}
```

27. Создан класс с именем GeoShape. Определен метод Area в этом классе, он вычисляет площадь геометрической фигуры. Вы хотите, чтобы производные классы GeoShape могли заменить его функциональность для поддержки расчетов дополнительных геометрических фигур. Когда метод Area вызывается в объекте GeoShape, площадь должна быть рассчитана, основываясь на используемом в текущий момент объекте GeoShape. Ключевое слово, которое вы должны использовать с определением площади методом в GeoShape класс?

- a) abstract
- b) virtual
- c) new
- d) overrides

28. Вы разрабатываете программы на C# и написали следующий код:

```
int x = 16;  
int y = --x;  
int z = y++;
```

Какое будет значение переменной z, после того, как все вышеописанные операторы выполнятся?

- a) 14
- b) 15
- c) 16
- d) 17

29. Вы пишете метод с именем PrintReport, который не возвращает значение в вызывающий код. Какое ключевое слово следует использовать в вашем объявлении метода?

- a) void
- b) private
- c) int
- d) string

30. Вы разрабатываете программу на C# и написали следующую строку кода:

```
int x = 6 + 4 * 4 / 2 - 1;
```

Каково будет значение переменной x после того, как этот оператор выполнится?

- a) 19
- b) 13
- c) 20
- d) 14

31. Вы разрабатываете программы на C# и написали следующий код:

```
int i = 6;  
do  
{  
    if (i == 3)  
        break;  
    Console.WriteLine("Значение i = {0}", i);  
    i++;  
}  
while (i <= 5);
```

Сколько раз будет выполнен цикл while?

- a) 0
- b) 1
- c) 2
- d) 3

32. Вы пишете программы на C#, которые используются для итерации по коллекциям (массивам и спискам). Вы должны убедиться, что вы обрабатываете каждый элемент коллекции один раз. Вы также должны

убедиться, что ваш код легко читать и отлаживать. Какое из следующих выражений на C# обеспечить лучшее решение для этого требования?

- a) while
- b) for
- c) foreach
- d) do-while

33. Какие из следующих описаний типов относятся к типам значений?

- 1. integer
- 2. array
- 3. float
- 4. string
- 5. long

- a) 1, 2, 5
- b) 1, 3, 5
- c) 2, 4
- d) 3, 5

34. Предположим, что вы новый C# - разработчик и изучаете основы объектно-ориентированного программирования. Какое из следующих утверждений не верно?

- a) Класс-это конкретный экземпляр объекта.
- b) Класс определяет шаблон для объекта.
- c) Класс — это определение нового типа данных.
- d) Конструктор используется для инициализации членов данных объекта.

35. Вы анализируете программу на C#, содержащую следующий класс

```
public class Rectangle
{
    public double Length {get; set;}
    public double Width { get; set; }
}
```

Программа выполняет следующий код в рамках метода Main

```
Rectangle r1, r2;
r1 = new Rectangle(Length = 10.0, Width = 20.0);
r2 = r1;
r2.Length = 30;
Console.WriteLine(r1.Length);
```

Каким будет результат, если этот код выполнится?

- a) 10
- b) 20
- c) 30
- d) 40

36. Вы разрабатываете программы на C# и написали следующий код:

```
int x = 10;
int y = ++x;
int z = --y;
```

Какое будет значение переменной z, после того, как все вышеописанные операторы выполнятся?

- a) 10
- b) 11
- c) 12
- d) 13

37. Какие из следующих утверждений являются правильным способом создания объекта класса Sample?

1. Sample s = new Sample();
2. Sample s;
3. Sample s; s = new Sample();
4. s = new Sample();

- a) 1, 3
- b) 2, 4
- c) 1, 2, 3
- d) 1, 4
- e) Ничего из вышеперечисленного

38. Какое из следующих описаний не применимо к знаковым типам данных?

- a) short
- b) integer
- c) long
- d) byte
- e) single

39. Какое из нижеперечисленных утверждений правильно объявляет двумерный целочисленный массив в C#?

- a) int[,] myArray;
- b) int[][] myArray;
- c) int[2] myArray
- d) System.Array[2] myArray;
- e) System.Array[,] myArray;

40. Если значение a равно 4 и значение b равно 5 , тогда значение выражения a - ++b равно -1

Да

Нет

Какого типа будет результат

```
Console.WriteLine(5 % 2);           // _____
Console.WriteLine(-5 % 2);          // _____
Console.WriteLine(5.0 % 2.2);       // _____
Console.WriteLine(5.0m % 2.2m);     // _____
Console.WriteLine(-5.2 % 2.0);      // _____
```

41. Результат выполнения следующего кода

```
int a = 5;
a -= 6;
Console.WriteLine(a);
```

42. Результат выполнения следующего кода

```
int a = 6;
a %= 3;
Console.WriteLine(a);
```

43. Результат выполнения следующего кода

```
double x;
x = 1.5;
Console.WriteLine(x++);
Console.WriteLine(x);
```

44. Программы на языке C# должны сохраняться с расширением _____.

45. Значение выражения $26 \% 4 \% 3$ равно _____

46. Результат выполнения следующего кода

```
int a = 5;
a += 6;
Console.WriteLine(a);
```

47. Результат выполнения следующего кода

```
double x;
x = 1.5;
Console.WriteLine(--x);
Console.WriteLine(x);
```

48. Какого типа будет результат

```
Console.WriteLine(12 % 2);    // _____
Console.WriteLine(-15 % 2);   // _____
Console.WriteLine(1555.0f % 2.2f); // _____
Console.WriteLine(235.0m % 2.2m); // _____
Console.WriteLine(-65.2 % 2.0); // _____
```

49. Результат выполнения следующего кода

```
int a = 5;
a /= 6;
Console.WriteLine(a);
```

50. Результат выполнения следующего кода

```
double x;
x = 1.5;
Console.WriteLine(++x);
Console.WriteLine(x);
```

51. Можно ли создать программу на языке C# без метода Main

Да
Нет

52. Какой вариант ответа является правильным результатом для программы C#, приведенной ниже?

```
int i = 20 ;  
for( ; ; )  
{  
    Console.Write(i + " ");  
    if (i >= -10)  
        i -= 4;  
    else  
        break;  
}
```

- A. 20 16 12 8 4 0 -4 -8
- B. 20 16 12 8 4 0
- C. 20 16 12 8 4 0 -4 -8 -12
- D. 16 12 8 4 0
- E. 16 8 0 -8

53. Каков результат приведенного ниже фрагмента кода C#?

```
namespace ConsoleApplication  
{  
    public enum color  
    { red, green, blue };  
  
    class SampleProgram  
    {  
        static void Main (string[ ] args)  
        {  
            color c = color.blue;  
            switch (c)  
            {  
                case color.red:  
                    Console.WriteLine(color.red);  
                    break;  
  
                case color.green:  
                    Console.WriteLine(color.green);  
                    break;  
  
                case color.blue:  
                    Console.WriteLine(color.blue);  
                    break;  
            }  
        }  
    }  
}
```

- A. red
- B. blue
- C. 0
- D. 1
- E. 2

54. Какой из следующих вариантов ответа является верным способом переписания приведенного фрагмента кода?

```

int i = 0;
do
{
    Console.WriteLine(i);
    i+ = 1;
}
while (i <= 10);

A. int i = 0;
    do
    {
        Console.WriteLine(i);
    }
    until (i <= 10);

B. int i;
    for (i = 0; i <= 10 ; i++)
        Console.WriteLine(i);

C. int i = 0;
    while (i <= 11)
    {
        Console.WriteLine(i);
        i += 1;
    }

D. int i = 0;
    do while ( i <= 10)
    {
        Console.WriteLine(i);
        i += 1;
    }

E. int i = 0;
    do until (i <= 10)
    {
        Console.WriteLine(i);
        i+=1;
    }

```

55. Какой из приведенных ниже фрагментов кода является верным способом определения, является ли параметр нечетным или четным?

1.

```
int a = Int32.Parse(Console.ReadLine());
String res;
if (a % 2 == 0)
    res = "Четный";
else
    res = "Нечетный";
```
2.

```
int a = Int32.Parse(Console.ReadLine());
String res;
if (a Mod 2 == 0)
    res = "Четный";
```

```

else
    res = "Нечетный";

3. int a = Int32.Parse(Console.ReadLine());
   Console.WriteLine(a Mod 2 == 0 ? "Четный": "Нечетный");

4. int a = Int32.Parse(Console.ReadLine());
   String res;
   a % 2 == 0 ? res = "Четный" : res = "Нечетный";
   Console.WriteLine(res);

```

- A. 1, 3
- B. Только 1
- C. 2, 3
- D. Только 4
- E. Ничего из вышеперечисленного

56. Какое из следующих утверждений является верным в приведенном ниже фрагменте кода C#?

```

if (age > 18 && no < 11)
    a = 25;

```

- 1. Условие *no < 11* будет оценено только в том случае, если *age > 18* будет равен *True*.
- 2. Выражение *a = 25* будет выполнено, если какое-либо из условий истинно.
- 3. Условие *no < 11* будет оценено, только если *age > 18* будет оценен как *False*.
- 4. Выражение *a = 25* будет выполнено, если оба условия равны *True*.
- 5. *&&* известен как логический оператор с коротким замыканием.

- A. 1, 3
- B. 2, 5
- C. 1, 4, 5
- D. 3, 4, 5
- E. Ничего из вышеперечисленного

57. Какие из следующих утверждений верны?

- 1. Оператор *switch* может использовать как *числовые* типы, так и *булевский* тип.
- 2. Оператор *switch* может использовать *символы*, *строки* и *перечисления*.
- 3. Нельзя объявлять переменные в операторе *case*, если он не заключен в *{}*.
- 4. Оператор *foreach* используется для итерации по коллекции для получения необходимой информации и может использоваться для изменения содержимого коллекции.

- A. 1, 2
- B. 2, 3
- C. 3, 4
- D. Ничего из вышеперечисленного

58. Каким будет результат выполнения фрагмента кода, приведенного ниже?

```

int i;
for(i = 0; i<=10; i++)
{
    if(i == 4)

```

```

    {
        Console.Write(i + " "); continue;
    }
    else if (i != 4)
        Console.Write(i + " "); else
        break;
}

```

- A. 1 2 3 4 5 6 7 8 9 10
- B. 1 2 3 4
- C. 0 1 2 3 4 5 6 7 8 9 10
- D. 4 5 6 7 8 9 10
- E. 4

59. Какие из следующих выражений являются правильными способами увеличения значения переменной a на 1?

- 1. ++a++;
- 2. a += 1;
- 3. a ++ 1;
- 4. a = a +1;
- 5. a = +1;

- A. 1, 3
- B. 2, 4
- C. 3, 5
- D. 4, 5
- E. Ничего из вышеперечисленного

60. Что из нижеперечисленного НЕ является арифметическим оператором в C#?

- A. **
- B. +
- C. /
- D. %
- E. *

61. Какие из следующих операторов НЕ относятся к C#?

- 1. >=
- 2. !=
- 3. Not
- 4. <=
- 5. <>=

- A. 1, 3
- B. 2, 4
- C. 3, 5
- D. 4, 5
- E. Ничего из вышеперечисленного

62. Какое из следующих утверждений является правильным результатом для фрагмента кода C#, приведенного ниже?

```
Console.WriteLine(13/2 + "" + 13% 2);
```

- A. 6,5 1
- B. 6,5 0
- C. 6 0
- D. 6 1
- E. 6.5 6.5

63. Что из нижеперечисленного НЕ является оператором присваивания в C#?

- A. $\forall =$
- B. $/=$
- C. $*=$
- D. $+=$
- E. $\%=$

Домашние задания (самостоятельная работа) по дисциплине «Языки и методы программирования»

1. Разработать консольное приложение, позволяющее вводить сторону квадрата a и вычислять его периметр P . Предусмотреть две возможности ввода: через командную строку и используя класс Console.
2. Разработать консольное приложение, позволяющее вводить сторону квадрата a и вычислять его площадь S . Предусмотреть две возможности ввода: через командную строку и используя класс Console.
3. Разработать консольное приложение, позволяющее вводить стороны прямоугольника a и b , и вычислять его площадь S . Предусмотреть две возможности ввода: через командную строку и используя класс Console.
4. Разработать консольное приложение, позволяющее вводить стороны прямоугольника a и b , и вычислять его периметр P и площадь $S = a^2$. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
5. Разработать консольное приложение, позволяющее вводить диаметр окружности d и вычислять ее длину L . Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций. Для ввода числа π , использовать стандартную библиотеку
6. Разработать консольное приложение, позволяющее вводить длину ребра куба a и вычислять объем куба V и площадь его поверхности S . Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций. Для ввода числа π , использовать стандартную библиотеку
7. Разработать консольное приложение, позволяющее вводить длины ребер a , b , c прямоугольного параллелепипеда и вычислять его объем V и площадь его поверхности S . Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
8. Разработать консольное приложение. Вводятся два числа a и b . Найти их среднее арифметическое и среднее геометрическое значение. Предусмотреть возможность работы с различными типами данных и две возможности ввода: через командную строку и используя класс Console
9. Разработать консольное приложение. Вводятся два ненулевых числа a и b . Найти сумму, разность, произведение и частное их квадратов. Предусмотреть возможность работы с различными типами данных и две возможности ввода: через командную строку и используя

класс Console

10. Разработать консольное приложение. Имеются два круга с общим центром и радиусами R_1 и R_2 ($R_1 > R_2$). Найти площади этих кругов S_1 и S_2 , а также площадь S_3 кольца. Предусмотреть возможность работы с различными типами данных и две возможности ввода: через командную строку и используя класс Console
11. Разработать консольное приложение, позволяющее вводить длину окружности L и находить радиус R и площадь S круга, ограниченного этой окружностью. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
12. Разработать консольное приложение, позволяющее по введенному значению площади S круга, находить его диаметр D и длину L окружности. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
13. Разработать консольное приложение, позволяющее вычислять корни квадратного уравнения. Программа должна вводить коэффициенты a , b и c в виде целых чисел. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
14. Разработать консольное приложение, позволяющее при вводе целого числа, определять является ли это число положительным или отрицательным, равным нулю, а также четным или нечетным. Должны выводиться сообщения типа: "Введено отрицательное четное число", "Введено положительное нечетное число", "Введено нулевое значение" и т.д. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
15. Разработать консольное приложение, позволяющее при вводе целого числа, лежащего в диапазоне 1-999, определять является ли это число положительным или отрицательным, равным нулю, четным или нечетным, а также однозначным, двухзначным или трехзначным. Должны выводиться сообщения типа: "Введено отрицательное четное двухзначное число", "Введено положительное нечетное трехзначное число", "Введено нулевое значение" и т.д. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
16. Разработать консольное приложение, позволяющее при вводе целого числа, лежащего в диапазоне 1-7, выводить строку с названием дня недели, соответствующее данному числу (1 — «Понедельник», 2 — «Вторник» и т. д.). Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
17. Разработать консольное приложение, позволяющее при вводе целого числа, лежащего в диапазоне 1-12, выводить строку с названием месяца, соответствующего данному числу (1 — «Январь», 2 — «Февраль» и т. д.). Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
18. Разработать консольное приложение, позволяющее при вводе целого числа, лежащего в диапазоне 1-12, выводить строку с названием времени года, соответствующего данному месяцу. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций
19. Разработать консольное приложение, позволяющее при вводе трех целых чисел, которые являются сторонами некоторого треугольника, проверить является ли треугольник; равнобедренным. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
20. Разработать консольное приложение, позволяющее при вводе трех целых чисел, которые являются сторонами некоторого треугольника, проверить является ли треугольник; равносторонним. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
21. Разработать консольное приложение, позволяющее при вводе трех целых чисел, которые являются сторонами некоторого треугольника, проверить является ли треугольник; прямоугольным. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
22. Разработать консольное приложение, позволяющее при вводе трех целых чисел, которые

являются сторонами некоторого треугольника, определить к какому типу относится данный треугольник; равнобедренный, равносторонний, прямоугольный, с произвольными сторонами или набор введенных чисел не может относиться к треугольнику. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.

23. Разработать консольное приложение выводящее название карты по введенным двум числам: масти и достоинству карты. Мастям игральным карт присвоить порядковые номера: 1 — пики, 2 — трефы, 3 — бубны, 4 — червы. Достоинству карт, старших десятки, присвоить номера: 11 — валет, 12 — дама, 13 — король, 14 — туз. Приложение должно выводить названия типа «шестерка бубен», «дама червей», «туз треф» и т. п. Предусмотреть две возможности ввода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.

Вопросы для самоконтроля студентов по время самостоятельной работы и при подготовке к экзамену или зачету

1. Объяснить, что такое тип.
2. Как можно определить понятие «переменная»?
3. Приведите примеры классов и объектов.
4. Перечислите признаки объектов.
5. В чём отличия членов класса от членов объекта?
6. Дайте определение идентификатора.
7. Объясните назначение отдельных частей простейшей программы на C#.
8. Каково назначение статического метода Main()?
9. Возможно ли написать программу на C#, не применяя классов?
10. Что такое тип void?
11. Какие методы класса Console применяются для ввода и вывода данных?
12. В какой момент выполняется чтение вводимых с клавиатуры данных?
13. В чём различие методов Console.Write() и Console.WriteLine()?
14. Что такое пространство имён?
15. Какое из слов конструкции System.Console.ReadLine() является именем пространства имён?
16. Для каких целей применяется директива using?
17. Чем решение (solution) в Visual Studio отличается от проекта (Project)?
18. Перечислите имеющиеся в языке C# средства объявления новых типов.
19. Как вы понимаете термин «инкапсуляция»?
20. В чём состоит различие статических и нестатических членов класса?
21. Перечислите известные вам виды членов класса.
22. Объясните смысл (назначение) каждого элемента в составном имени System.Console.WriteLine.
23. Что обозначает операция «+» в бинарном выражении, одним из операндов которого является строка (объект класса string)?
24. Чем отличаются типы знаковые от беззнаковых.
25. Приведите примеры констант-литералов числовых (арифметических) типов.
26. Укажите назначение десятичного типа и правила записи его констант.
27. Назовите способы записи символьных констант.
28. Назовите размеры (в битах) представления в памяти констант базовых типов.
29. Какие символы допустимы в идентификаторах C#?
30. Приведите примеры служебных слов языка C#.
31. Является ли идентификатор Main служебным словом?
32. Что такое инициализация переменной?
33. Чем именованная константа отличается от константы литерала?
34. Перечислите группы (категории) операций языка C#.

35. Перечислите названия групп операций в порядке возрастания их приоритетов (рангов).
36. Знаки каких бинарных операций могут использоваться в составных операциях присваивания?
37. В чём отличия префиксных форм операций декремента и инкремента от постфиксных.
38. К каким операциям применимы операции ++ и --?
39. К каким операндам применима операция %?
40. В чём особенность операции деления целочисленных операндов?
41. Назовите правила выполнения операций %.
42. Объясните механизм возникновения переполнения при вычислениях с целочисленными операндами
43. Назовите две основные части платформы .NET Framework.
44. Что такое CIL(Common Intermediate Language)?
45. Укажите назначение общезыковой спецификации CLS – Common Language Specification?
46. Какие члены могут присутствовать в классе в соответствии с требованиями общей системы типов CTS?
47. Приведите названия типов CTS, которые представлены в языке C# базовыми типами, например, double.
48. Какие базовые типы языка C# не соответствуют CLS?
49. Каково назначение оператора в программах на C#?
50. Перечислите операторы языка C#.
51. Каков обязательный признак отличного от блока оператора в C#?
52. Что такое оператор-выражение?
53. Дайте определение блока.
54. Какими правилами определяются вход в блок и выход из него?
55. Назовите операторы выбора (ветвлений).
56. Какие операторы не могут входить в условный оператор?
57. Что такое сокращённая форма условного оператора?
58. Как устанавливается соответствие между if и else при вложениях условных операторов?
59. Назовите виды операторов циклов в C#.
60. Какой тип имеет выражение-условие в операторе цикла?
61. Сколько элементов в заголовке цикла общего вида (цикла for) и как они разделяются?
62. Что такое инициализатор цикла общего вида (цикла for)?
63. Когда вычисляется завершающее выражение цикла for?
64. Как выполняется вложение циклов?
65. Какие операторы могут прервать выполнение цикла до его завершения, запланированного выражением-условием?
66. Каково минимальное количество итераций в цикле с постусловием?
67. Назовите назначение оператора break. Где его можно применять?
68. Укажите возможности оператора goto при вложениях циклов.
69. Где и когда употребляется оператор continue?
70. Какого типа может быть значение переключающего выражения в переключателе?
71. Каким оператором должна завершиться ветвь переключателя?
72. Какая конструкция вводит ветвь переключателя?
73. В каких случаях выполняется ветвь переключателя, введённая меткой default?
74. Являются ли типы массивов типами значений?
75. Какое значение имеет индексирующее выражение при обращении к первому элементу одномерного массива?

76. Какой тип может иметь индексирующее выражение?
77. Где размещается ссылка на массив: в стеке или в управляемой памяти (в куче)?
78. Какие значения принимают элементы массива при отсутствии в его определении инициализатора?
79. Какова структура инициализатора массива?
80. Чем определяется количество инициализирующих выражений в инициализаторе массива?
81. Объясните назначение всех элементов цикла `foreach`.
82. Каково назначение и возможности переменной цикла `foreach`.
83. Можно ли изменить размер массива после его создания?
84. Можно ли изменить размер массива в процессе выполнения программы?
85. Назовите свойства массивов, унаследованные ими от класса `Array`.
86. Приведите примеры нестатических методов одномерных массивов.
87. Приведите примеры статических методов одномерных массивов.
88. В чём различия методов `Copy()` и `Clone()`?
89. Что такое размерность массива?
90. Допустимо ли динамическое определение размеров многомерных массивов?
91. Чему равно свойство `Length` для многомерного массива?
92. С помощью каких средств можно получить размер многомерного массива по нужному измерению?
93. Сколько спецификаторов размерности в объявлении типа четырёхмерного массива?
94. Перечислите синтаксические отличия массива/массивов от двумерного массива.
95. Сколько операций `new` в определении объекта трёхмерного массива?
96. Чему равно свойство `Rank` массива массивов?
97. Какие элементы входят в заголовок метода?
98. Что такое тело метода?
99. Назовите особенности метода-процедуры.
100. В каком случае в теле метода может отсутствовать оператор `return`?
101. Перечислите возможные источники данных, получаемых методом при его выполнении.
102. Перечислите модификаторы параметров методов.
103. Укажите область видимости параметра метода.
104. Назовите виды параметров.
105. В чём особенности статических методов?
106. Может ли статический метод играть роль процедуры?
107. Назовите требования к аргументам метода, заменяющим фиксированные параметры.
108. В чём отличия передачи параметров по значениям от передачи по ссылкам?
109. Какие ограничения накладываются на аргументы, заменяющие передаваемые по ссылке параметры. В чём отличия модификаторов `out` и `ref`?
110. Может ли параметр с типом ссылки передаваться методу по значению?
111. Может ли параметр с типом ссылки снабжён модификатором `ref`?
112. Какой параметр представляет в теле метода список аргументов нефиксированной длины?
113. Как в теле метода выполняются обращения к аргументам, количество которых переменное?
114. Можно ли за счёт выполнения метода изменить значения аргументов, представляемых в методе параметром с модификатором `params`?
115. Приведите примеры полиморфизма.
116. Что определяет в сигнатуру метода?
117. Что такое перегрузка методов?

118. Какой метод называют рекурсивным?
119. В чём отличие косвенной рекурсии от прямой?
120. Перечислите члены класса, которые могут быть объявлены статическими.
121. Какие члены класса являются статическими без применения модификатора static?
122. Перечислите модификаторы доступа.
123. Можно ли в объявлении члена класса использовать два модификатора доступа?
124. Что такое поле класса?
125. Можно ли в объявлении статического поля использовать модификатор доступа?
126. Как объявляются константы, принадлежащие классу?
127. Сформулируйте правила инициализации констант класса.
128. В чём отличие статических констант от статических полей с модификатором readonly.
129. Перечислите возможные применения и ограничения статических методов.

Примерные вопросы к экзамену (проводится в устной форме) в 6 семестре.

На экзамене важно не только сказать словами, но в большинстве случаев записать то или иное понятие в виде синтаксической конструкции.

Особенно важно приводить примеры программ.

1. Общая характеристика C#.
2. Типы приложений, которые можно разрабатывать с помощью C#.
3. Назначение компонентов в архитектуре .Net.
4. Платформа .NET и ее базовые составляющие.
5. Парадигмы программирования.
6. Прimitивные типы данных. Синтаксис.
7. Тип данных булевский.
8. Символьный тип данных.
9. Тип данных String.
10. Тип Object.
11. Определение класса и членов. Синтаксис.
12. Модификаторы доступа.
13. Ключевое слово this.
14. Использование классов и объектов.
15. Конструкторы.
16. Определение и использование свойств.
17. Перечисления (Enums). Назначение.
18. Методы. Общая форма определения метода.
19. Сигнатура метода.
20. Параметры методов. Передача по значению, по ссылке.
21. Ключевые слова ref или out.
22. Передача параметров ссылочного типа. Out. Params. Ref.
23. Основополагающие принципы ООП.
24. Наследование. Типы наследования. Терминология наследования.
25. Наследование в C#. Ключевое слово base.

26. Абстрактные классы
27. Интерфейсы в C#.
28. Полиморфизм. Определение и назначение.
29. Классический полиморфизм.
30. Полиморфизм, посредством перегрузки методов.
31. Связывание. Раннее связывание. Позднее связывание.
32. Виртуальные методы.
33. Суть парадигмы объектно-ориентированного программирования.
34. Массивы. Определение.
35. Общая форма объявления.
36. Общая форма инициализации.
37. Одномерные массивы.
38. Инициализация массива.
39. Массивы типов значений и ссылочных типов.
40. Многомерные массивы.
41. Массивы массивов.
42. Структуры в C#.
43. Сравнение структур и классов.
44. Различия между передачей структуры и ссылки класса в метод.
45. Однострочные и многострочные комментарии.
46. Метод Main. Способы объявления метода Main.
47. Операции арифметические, сравнения, логические.
48. Область видимости переменных. Правила определения области видимости.
49. Консольный ввод-вывод.
50. Спецификаторы стандартных числовых форматов.
51. Класс Math из пространства имен System. Основные методы и константы.
52. Служебные слова virtual и override. Назначение и примеры использования.
53. Свойства. Метод доступа get. Метод доступа set.
54. Оператор if.
55. Оператор switch. Оператор break.
56. Цикл while. Цикл do ... while. Цикл for. Цикл foreach.
57. Специальные операторы для циклов: break, continue, goto.
58. Статические методы и методы экземпляра.
59. Виртуальные, переопределяемые и абстрактные методы.
60. Рекурсивные методы.
61. Класс Array. Свойства. Методы.
62. Константы и литералы.

Примерные вопросы к экзамену (проводится в устной форме) в 7 семестре.

1. HTML5. Общая характеристика.
2. Подход HTML 4 к веб-разметки.
3. Подход HTML5 к веб-разметке. Общая характеристика. Семантическая разметка.
4. Элементы header и footer. Назначение и использование. Пример.
5. Элемент <section>. Назначение и использование. Пример.
6. Элемент <article>. Назначение и использование. Пример.
7. Элемент <aside>. Назначение и использование. Пример.
8. Универсальные селекторы. Селекторы атрибутов элементов. Селекторы нижележащих элементов.
9. Псевдо-классы в CSS.
10. Псевдо-элементы в CSS.
11. Боксы. Типы боксов.
12. Поток и расположение элементов
13. Выравнивание блока
14. Поток и расположение элементов
15. Позиционирование. Типы позиционирования. z-index
16. Границы и отступы. Выступ за границы родительского элемента. Плавающие блоки
17. Очистка float. Методы очистки float.
18. Flexbox. Основные преимущества flexbox
19. Типы данных JavaScript.
20. Объявление и использование переменных.
21. Арифметические операторы и операторы сравнения
22. Условные операторы: if, if-else. Оператор switch-case
23. Функции в JavaScript.
24. Анонимные функции
25. Способы определения функции в JS
26. Область действия переменных и функций
27. Четыре способа вызова функций в JavaScript
28. Объекты в JavaScript. Создание объекта. Способы.
29. Прототип. Цепь прототипов
30. Классическое наследование и безклассовое наследование (Prototypal)
31. Массивы в JavaScript. Общая характеристика.
32. Методы для работы с массивами
33. Введение в библиотеку jQuery. Общая характеристика.
34. Ненавязчивый JavaScript.
35. AngularJS. Общая характеристика
36. Модель MVC.

- 37.Bootstrap. Основные инструменты Bootstrap.
- 38.Адаптивный дизайн. Назначение и использование
- 39.Элементы header, nav, section, aside, footer, article, hgroup, figure и figcaption
- 40.Элементы mark, small, cite, address, time,
- 41.Традиционная блочная модель.
- 42.Выравнивание тела документа по центру
- 43.Навигационная полоса
- 44.Раздел и боковая врезка
- 45.Создание двух столбцов с помощью свойства float
- 46.Нижний колонтитул.
- 47.Свойства border-radius, box-shadow, text-shadow, @font-face
- 48.Линейный и радиальный градиенты. RGBA(). HSLA()
- 49.Свойства outline, border-image, transform и transition
- 50.Функции transform: scale, transform: rotate, transform: skew, transform: translate
- 51.Одновременное использование всех видов трансформации
- 52.Формы HTML и ввод данных. Общая характеристика. Базовые элементы.
- 53.Фиксированный, резиновый дизайн и адаптивный дизайны.
- 54.Макеты на основе обтекаемых элементов..
- 55.Логические операции.
- 56.Циклы while, do while, for, for ... in. Операторы break и continue
- 57.Основы работы с DOM. DOM.
- 58.Работа с событиями в DOM.
- 59.Знакомство и работа с элементом canvas. Общая характеристика.
- 60.Знакомство с SVG. Общая характеристика.

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

Оценивание степени освоения обучающимися дисциплины осуществляется на основе «Положение о балльно-рейтинговой системе оценки успеваемости студентов МГОУ».

Шкала соответствия рейтинговых оценок пятибалльным оценкам

Оценка по 5-балльной системе		Оценка по 100-балльной системе
5	отлично	81 – 100

4	хорошо	61 - 80
3	удовлетворительно	41 - 60
2	неудовлетворительно	21 - 40
1	необходимо повторное изучение	0 - 20

В зачетно-экзаменационную ведомость и зачетную книжку выставляются оценки по пятибалльной шкале и рейтинговые оценки в баллах.

При получении студентом на зачёте неудовлетворительной оценки в ведомость выставляется рейтинговая оценка в баллах (<40 баллов), соответствующая фактическим знаниям (ответу) студента.

Общее количество баллов по дисциплине - 100 баллов.

Максимальное количество баллов, которое можно набрать в течение семестра за посещаемость, выполнение лабораторных, практических и домашних работ, тестирование и реферат и т.д. – 80 баллов.

За посещение лекционных занятий и написание конспектов обучающийся может набрать максимально 15 баллов.

За выполнение домашних заданий обучающийся может набрать максимально 10 баллов (5 заданий по 2 балла).

За выполнение рефератов по дисциплине обучающийся набрать максимально 10 баллов.

За выполнение лабораторных работ обучающийся может набрать максимально 30 баллов (15 работ по 2 балла).

За тестирование обучающийся может набрать максимально 15 баллов (15 тестовых вопросов по 1 баллу за каждый).

Максимальная сумма баллов, которые обучающийся может набрать при сдаче зачета, составляет 20 баллов.

Для сдачи зачета по дисциплине необходимо выполнить все требуемые лабораторные работы (получить допуск к зачету у преподавателя, проводившего лабораторные работы). Существенным моментом является посещаемость занятий (в случае пропусков занятий предполагается более подробный опрос по темам пропущенных занятий). На зачет выносятся материал, излагаемый в лекционном курсе и рассматриваемый на практических и лабораторных занятиях. Для получения зачета надо правильно ответить на несколько поставленных вопросов.

Критерии и шкала оценивания домашней работы

Критерий	Баллы
Решение логически выстроено и точно изложено, ясен весь ход рассуждения	0,5
Представлено решение задач несколькими способами (если это возможно)	0,5
Ответ на каждый вопрос (задание) заканчивается выводом	0,5
Оформление соответствует образцу. Представлены необходимые таблицы и схемы	0,5

По результатам оценивания обучающийся может получить:

Пороговый уровень – до 1 балла;

Продвинутый уровень – 1,5-2 балла.

Критерии и шкала оценивания конспекта

Критерий	Баллы
----------	-------

Текст конспекта логически выстроен и точно изложен, ясен весь ход рассуждения	0,5
Даны ответы на все поставленные вопросы, изложены научным языком, с применением терминологии	0,5
Ответ на каждый вопрос заканчивается выводом, сокращения слов в тексте отсутствуют (или использованы общепринятые)	0,5
Оформление соответствует образцу. Представлены необходимые таблицы и схемы	0,5

По результатам оценивания обучающийся может получить:

Пороговый уровень – до 1 балла;

Продвинутый уровень – 1,5-2 балла.

Критерии и шкала оценивания реферата (доклада)

Критерий	Баллы
Обзор источников информации	0,5
Логика изложения материала	0,5
Убедительность сформулированных выводов	0,5
Качество оформления	0,5

По результатам оценивания обучающийся может получить:

Пороговый уровень – до 1 балла;

Продвинутый уровень – 1,5-2 балла.

Шкала оценивания теста, контрольной работы

Показатель	отметка
Выполнено до 40% заданий	2
Выполнено 41-60% заданий	3
Выполнено 61-80% заданий	4
Выполнено более 81% заданий	5

Критерии и шкала оценивания работы студентов на лекциях и практических занятиях

Шкала	Показатели степени обученности
0,5 балл	Присутствовал на занятии, слушал, смотрел, записывал под диктовку, переписывал с доски и т.п. Отличает какой-либо процесс, объект и т.п. от их аналогов только тогда, когда ему их предъявляют в готовом виде.
1 балла	Запомнил большую часть текста, правил, определений, формулировок, законов и т.п., но объяснить ничего не может (механическое запоминание). Демонстрирует полное воспроизведение изученных правил, законов, формулировок, математических и иных формул и т.п., однако затрудняется что-либо объяснить.
1,5 баллов	Объясняет отдельные положения усвоенной теории, иногда выполняет такие мыслительные операции, как анализ и синтез. Отвечает на большинство вопросов по содержанию теории, демонстрируя осознанность усвоенных теоретических знаний, проявляя способность к

	самостоятельным выводам и т.п.
2 балла	Четко и логично излагает теоретический материал, свободно владеет понятиями и терминологией, способен к обобщению изложенной теории, хорошо видит связь теории с практикой, умеет применить ее в простейших случаях. Демонстрирует полное понимание сути изложенной теории и свободно применяет ее на практике. Выполняет почти все практические задания, иногда допуская незначительные ошибки, которые сам и исправляет. Легко выполняет практические задания на уровне переноса, свободно оперируя усвоенной теорией в практической деятельности. Оригинально, нестандартно применяет полученные знания на практике, формируя самостоятельно новые умения на базе полученных ранее знаний и сформированных умений и навыков.

Шкала оценивания отчета по самостоятельной работе

Критерий	Баллы
Полнота и глубина ответа. Наличие методических комментариев и примеров.	1
Содержательность и объем выполненного задания. Рассмотрение вопроса во всех сторон.	1
Знание и рациональное использование средств ИКТ.	1
Определение достоинств и недостатков различных явлений, процессов	1
Наличие выводов	1

По результатам оценивания обучающийся может получить:

Пороговый уровень – до 3 баллов;

Продвинутый уровень – 4-5 баллов.

Шкала оценивания домашней работы

Показатель	Отметка, балл
Выполнено до 80% заданий	1
Выполнено более 81% заданий	2

Шкала оценивания решения задач

Показатель	Отметка, балл
Выполнено до 80% заданий	1
Выполнено более 81% заданий	2

Критерии оценивания презентаций (баллы)

Параметры оценивания презентации	баллы
Связь темы презентации с программой и учебным планом	0,5
Содержание презентации.	0,5
Заключение презентации	0,5
Подача материала проекта – презентации	0,5

Графическая информация (иллюстрации, графики, таблицы, диаграммы и т.д.)	0,5
Наличие импортированных объектов из существующих цифровых образовательных ресурсов и приложений Microsoft Office	0,5
Графический дизайн	0,5
Техническая часть	0,5
Эффективность применения презентации в учебном процессе	0,5
Итоговое количество баллов:	

Курсовая работа рассматривается как самостоятельный вид учебной работы и оценивается по 100-бальной рейтинговой шкале.

Для оценки курсовых работ используется следующая схема рейтингового расчета:

Раздел	Критерии	Рейтинговая оценка
1. Самостоятельность выполнения работы	Работа написана самостоятельно	15
	Работа носит частично самостоятельный характер	10
	Работа носит не самостоятельный характер	0
2. Содержание работы	Полностью соответствует выбранной теме	15
	Частично соответствует выбранной теме	10
	Не соответствует теме	0
3. Элементы исследования	Определены цели и задачи исследования, сформулированы объект и предмет исследования, показана история и теория вопроса	15
	Определены цели и задачи исследования, не четко определены объект и предмет исследования, частично показана история и теория вопроса	10
	Не определены цели и задачи исследования, не сформулированы объект и предмет исследования, не показана история и теория вопроса	0
4. Цитирование и наличие ссылочного материала	Достаточно	10
	Частично	5
	Не использовались	0
5. Наличие собственных выводов, рекомендаций и предложений, собственной позиции и ее аргументации	Да	15
	Нет	0
6. Оформление работы	Соответствует полностью требованиям	10
	Соответствует частично требованиям	5
	Не соответствует требованиям	0
7. Библиография по теме работы	Актуальна и составлена в соответствии с требованиями	10
	Актуальна и частично соответствует требованиям	5
	Не соответствует требованиям	0
8. Оценка на защите	Владеет материалом	10
	Частично владеет материалом	5
	Не владеет материалом	0

