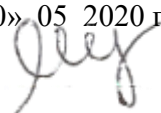


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Наумова Наталия Александровна
Должность: Ректор
Дата подписания: 24.10.2020 14:21:41
Уникальный программный ключ:
6b5279da4e034bffa679172803da5b7b559fc69e2

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» 05 2020 г., № 10
Зав. Кафедрой  / Шевчук М.В./

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ**

по дисциплине
Языки и методы программирования

Направление подготовки
44.03.05 Педагогическое образование

Профиль
Физика и информатика

Мытищи
2020

Автор-составитель:

Бычкова Дарья Дмитриевна
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 44.03.05 Педагогическое образование профиль «Физика и информатика», утвержденного приказом МИНОБРНАУКИ РОССИИ от 22.02.2018 г. № 125.

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
ОПК-8 «Способность осуществлять педагогическую деятельность на основе специальных научных знаний»	1. Работа на учебных занятиях. 2. Самостоятельная работа.

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

Оцениваемые компетенции	Уровень сформированности	Этапы формирования	Описание показателей	Критерии оценивания	Шкала оценивания
ОПК-8	Пороговый	1. Работа на учебных занятиях. 2. Самостоятельная работа.	<i>Знать:</i> -особенности планирования учебного процесса в соответствии соответствующей предметной области; -содержание, формы и методы технологии обучения и диагностики; -основные понятия из предметной области; <i>уметь:</i> -планировать и осуществлять отдельные составляющие учебного процесса в соответствии в соответствующей предметной области; - использовать некоторые формы технологии обучения и диагностики; - применять некоторые методы технологии обучения и диагностики; -применять основные понятия из предметной области; <i>владеть:</i>	Текущий контроль (выполнение лабораторных работ и заданий для самостоятельной работы) Промежуточный контроль (зачет в виде контрольной работы)	41-60

			-некоторыми методами технологии обучения и диагностики; -особенностями планирования учебного процесса в соответствии в соответствующей предметной области.		
Продвинутый	1. Работа на учебных занятиях. 2. Самостоятельная работа.	<i>Знать:</i> -особенности планирования и осуществления учебного процесса в соответствии в соответствующей предметной области; -содержание, формы и методы технологии обучения и диагностики; -основные понятия и сопряженные с ними из предметной области; <i>уметь:</i> -планировать и осуществлять все составляющие учебного процесса в соответствии в соответствующей предметной области; - использовать различные формы технологии обучения и диагностики; - применять различные методы технологии обучения и диагностики; -применять основные понятия и сопряженные с ними из предметной области; <i>владеть:</i> -особенностями планирования учебного процесса в соответствии в соответствующей предметной области; -различными формами и методами технологии обучения и диагностики.	Текущий контроль (выполнение лабораторных работ и заданий для самостоятельной работы) Промежуточный контроль (зачет в виде контрольной работы)	61-100	

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 разрабатываете код, который определяет метод InitFields. Метод принимает два параметра с типом данных Double и не возвращает никакого значения в вызывающий код. Какие из следующих сегментов кода вы бы использовали для определения метода InitFields?

- a)

```
public double InitFields(double l, double w)
{
    length = l;
    width = w;
    return length * width;
}
```
- b)

```
public void InitFields(double l, double w)
{
    length = l;
    width = w;
}
```
- c)

```
public void InitFields(double l)
{
    length = l;
    width = l;
    return;
}
```
- d)

```
public double InitFields(double l, double w)
{
    length = l;
    width = w;
}
```

2. 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

3. 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

4. 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]

5. 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

6. 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

7. 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);
}

```

8. 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

9. 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

10. 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`

11. 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

1

12. 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.

13. Какой из нижеприведенных результатов будет получен при выполнении следующего кода

```
namespace ConsoleApplication  
{  
    class Sample  
    {  
        int i;  
        Single j;
```



```

        public void SetData(int i, Single j)
        {
            i = i;
            j = j;
        }
        public void Display()
        {
            Console.WriteLine(i + " " + j);
        }
    }
}
class MyProgram
{
    static void Main(string[ ] args)
    {
        Sample s1 = new Sample();
        s1.SetData(10, 5.4f);
        s1.Display();
    }
}

```

- a) 0 0
- b) 10 5.4
- c) 10 5.400000
- d) 10 5
- e) Ничего из вышеперечисленного

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

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

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

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

16. Необходимо предоставить возможность печати в нескольких классах. В каждом классе алгоритм печати скорее всего будет отличаться. Кроме того, не все классы имеют отношения «is-a» друг с другом. Как вы должны поддержать эту функциональность?

- a) Добавить возможность печати в базовый класс с модификатором доступа public.
- b) Сделать так, чтобы все классы наследовались от базового класса и переопределить метод базового класса, чтобы обеспечить собственные функции печати в других классах.

- c) Сделать так, чтобы все классы наследовались от базового класса, который обеспечивает функции печати.
- d) Создать общий интерфейс, который все классы реализуют.

17. 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

18. 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.

19. 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

20. 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

21. 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

22. Вы разрабатываете программу на C#, которая должна выполнить 5 повторов. Вы написали следующий код:

```
01: int count = 0;
02: while (count <= 5)
03: {
04:     Console.WriteLine("Значение count = {0}", count);
05:     count++;
06: }
```

При запуске программы, вы заметили, что цикл не повторяется пять раз. Что нужно сделать, чтобы цикл выполнялся ровно пять раз?

- a) Изменить код в строке 01 на `int count = 1;`
- b) Изменить код в строке 02 на `while (count == 5)`
- c) Изменить код в строке 02 на `while (count >= 5)`
- d) Изменить код в строке 05 на `++count;`

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

24. 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`;

25. 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`

26. 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

27. 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

28. 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

29. Что из нижеперечисленного не является целым числом?

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

30. 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

31. 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 ...
    }
}
```

32. Вы разрабатываете приложение на C#. Вы должны решить, следует ли объявить член класса как статический. Какое из следующих утверждений верно для статических членов классов?

- a) Вы можете использовать ключевое слово `this` в качестве ссылки на статический метод или свойство.
- b) Только одна копия статического поля совместно используется всеми экземплярами класса.
- c) Статические члены класса можно использовать только после создания экземпляра класса.
- d) Ключевое слово `static` используется для объявления членов, которые не относятся к отдельным объектам, а только к самому классу.

33. 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`

34. 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

35. 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`

36. 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

37. 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

38. 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

39. 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]

40. 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);
}

```

41. 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?

- e) после каждого case добавьте следующую строку кода: `break`;
- f) после каждого case добавьте следующую строку кода: `continue`;
- g) после каждого case добавьте следующую строку кода: `goto default`;
- h) после каждого case добавьте следующую строку кода: `return`;

42. Какие из следующих утверждений являются верными?

- a) Процедурная парадигма программирования отличается от парадигмы структурного программирования.
- b) Парадигма объектно-ориентированного программирования направлена на разделение логики на более мелкие части и написание процедур для каждой части.
- c) Классы и объекты - краеугольные камни парадигмы структурного программирования.
- d) В парадигме объектно-ориентированного программирования одинаково важны данные и процедуры, которые работают с этими данными.
- e) C# на платформе .NET — это язык структурного программирования.

43. 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);
    }
}
```

- f) j не может быть объявлен как public
- g) DisplayData() не может быть объявлен как private.
- h) DisplayData() не может получить доступ к j.
- i) ShowData() не может получить доступ к i
- j) Ошибки нет в этом классе.

44. 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

45. Какой .Net тип соответствует типу int в языке C#?

- a) System.Int16
- b) System.Int32
- c) System.Int64
- d) System.Int128

46. Вы создали новый класс с именем Polygon и написали следующий код

```
class Polygon : IComparable
{
    public double Length { get; set; }
    public double Width { get; set; }

    public double GetArea()
    {
        return Length * Width;
    }

    public int CompareTo(object obj)
    {
        // должен быть завершен
    }
}
```

Вам нужно завершить определение метода CompareTo для сравнения объектов Polygon. Какие из следующих сегментов кода следует использовать?

- a)

```
public int CompareTo(object obj)
{
    Polygon target = (Polygon)obj;
    double diff = this.GetArea() - target.GetArea();

    if (diff == 0)
        return 0;
    else if (diff > 0)
        return 1;
    else return -1;
}
```
- b)

```
public int CompareTo(object obj)
{
    Polygon target = (Polygon)obj;
    double diff = this.GetArea() - target.GetArea();

    if (diff == 0)
        return 1;
    else if (diff > 0)
        return -1;
    else return 0;
}
```

```
}
```

c)

```
public int CompareTo(object obj)
{
    Polygon target = (Polygon)obj;

    if (this == target)
        return 0;
    else if (this > target)
        return 1;
    else return -1;
}
```

d)

```
public int CompareTo(object obj)
{
    Polygon target = (Polygon)obj;

    if (this == target)
        return 1;
    else if (this > target)
        return -1;
    else return 0;
}
```

47. 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.

48. 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

49. 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

50. 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

51. Вы пишете программы на C#, который должен оперировать с очень большими целочисленными значениями, которые могут превышать 12 цифр. Значения могут быть положительными или отрицательными. Какой тип данных следует использовать для хранения таких переменных?

- a) int
- b) float
- c) double
- d) long

52. 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

53. Вы разрабатываете программу на C#, которая должна выполнить 5 повторов. Вы написали следующий код:

```
01: int count = 0;  
02: while (count <= 5)  
03: {  
04:     Console.WriteLine("Значение count = {0}", count);  
05:     count++;  
06: }
```

При запуске программы, вы заметили, что цикл не повторяется пять раз. Что нужно сделать, чтобы цикл выполнялся ровно пять раз?

- a) Изменить код в строке 01 на `int count = 1;`
- b) Изменить код в строке 02 на `while (count == 5)`
- c) Изменить код в строке 02 на `while (count >= 5)`
- d) Изменить код в строке 05 на `++count;`

54. 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]`

55. 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

56. 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

57. 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

58. 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);
    }
}
```

- k) `j` не может быть объявлен как `public`
- l) `DisplayData()` не может быть объявлен как `private`.
- m) `DisplayData()` не может получить доступ к `j`.
- n) `ShowData()` не может получить доступ к `i`
- o) Ошибки нет в этом классе.

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

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

60. 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;`

61. C# поддерживает _____ (количество) целочисленных типов данных.

62. Если значение `a` равно 4 и значение `b` равно 5 , тогда значение выражения `a – ++b` равно -1

Да

Нет

63. 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);       // _____
```

64. Результат выполнения следующего кода

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

65. Результат выполнения следующего кода

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

66. Результат выполнения следующего кода

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

67. Записать метод Main, который возвращает значение void.

68. К целочисленным типам без знака относятся

69. Модификатор _____ обеспечивает доступ к типу или члену из любого кода в той же сборке, но не из другой сборки.

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

71. C# поддерживает _____ (количество) типа чисел с плавающей точкой.

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

73. Результат выполнения следующего кода

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

74. Результат выполнения следующего кода


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

75. Записать метод Main, который возвращает целое значение.

76. К целочисленным типам со знаком относятся

77. Модификатор _____ обеспечивает доступ к типу или члену из любого другого кода в той же сборке или другой сборке, ссылающейся на него.

78. В .NET Framework имеются переменные двух типов _____ и _____

79. Формальное описание языка программирования называется

80. Тип ushort занимает _____ байт.

81. Если a равно 7.5 и b равно 7, тогда значение $a \geq b$ равняется false

Да

Нет

82. 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); // _____
```

83. Результат выполнения следующего кода

```
int a = 5;
```

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

84. Результат выполнения следующего кода

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

85. Записать метод Main, который возвращает значение void и использует один формальный параметр.

86. Модификатор _____ обеспечивает доступ к типу или члену только из кода в том же классе или структуре, или в производном классе.

87. Промежуточный язык (MSIL) (определение)

88. Чем компиляция отличается от интерпретации?

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

Да

Нет

90. Двумя основными компонентами среды .Net Framework являются _____ и _____

91. Если a равно 7.5 и b равно 7, тогда значение a >= b равняется false

Да

Нет

92. 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); // _____
```

93. Результат выполнения следующего кода

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

94. Результат выполнения следующего кода

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

95. Записать метод Main, который возвращает значение void и использует один формальный параметр.

96. Модификатор _____ обеспечивает доступ к типу или члену только из кода в том же классе или структуре, или в производном классе.

97. Промежуточный язык (MSIL) (определение) _____

98. Чем компиляция отличается от интерпретации?

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

Да
Нет

100. Двумя основными компонентами среды .Net Framework являются _____ и _____

101. Что напечатает следующий фрагмент кода C # .NET?

```
int i = 0, j = 0;
label:
    i++;
    j+=i;
if (i < 10)
{
    Console.Write(i + " ");
    goto label;
}
```

- A. 1 2 3 4 5 6 7 8 9
- B. 0 1 2 3 4 5 6 7 8
- C. 2 3 4 5 6 7 8
- D. 2 3 4 5 6 7 8 9
- E. Ошибка компиляции на *label*..

102. 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 84 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

103. 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

104. 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;`
`}`

105. Каким будет результат фрагмента кода C#, приведенного ниже?

```

int val;
for (val = -5; val <= 5; val++)
{
    switch (val)
    {
        case 0:
            Console.Write ("Sun");
            break;
    }

    if (val > 0)
        Console.Write ("B");
    else if (val < 0)
        Console.Write ("X");
}

```

- A. XXXXXSun
- B. SunBBBBB
- C. XXXXXSunBBBBB
- D. BBBBBSunXXXXX
- E. Zero

106. Каким будет результат выполнения фрагмента кода C#, приведенного ниже?

```

char ch = Convert.ToChar ('a' | 'b' | 'c');
switch (ch)

```

```

{
    case 'A':
    case 'a':
        Console.WriteLine ("case A | case a");
        break;

    case 'B':
    case 'b':
        Console.WriteLine ("case B | case b");
        break;

    case 'C':
    case 'c':
    case 'D':
    case 'd':
        Console.WriteLine ("case D | case d");
        break;
}

```

- A. case A | case a
- B. case B | case b
- C. case D | case d
- D. Ошибка компиляции
- E. Ничего из вышеперечисленного

107. Какой из следующих вариантов оператора управления if является не правильным?

- A. if (Condition1)
 {// Исполняемый код }
- B. if (Condition1)
 {// Исполняемый код }
else
 {// Исполняемый код }
- C. if (Condition1)
 {// Исполняемый код }
else
 {// Исполняемый код }
else if (Condition2)
 {// Исполняемый код }
- D. if (Condition1)
 {// Исполняемый код }
else if (Condition2)
 {// Исполняемый код }
else
 {// Исполняемый код }
- E. if (Condition1)

```

    { // Исполняемый код }
else if ( Condition2 )
    { // Некоторая инструкция }
else if ( Condition3 )
    { // Исполняемый код }
else
    { // Исполняемый код }

```

108. 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

109. Что из нижеперечисленного может быть использовано для завершения цикла while и передачи управления вне цикла?

1. exit while
2. continue
3. exit statement
4. break
5. goto

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

110. 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

111. 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

112. Каким будет результат выполнения фрагмента кода C#?

```
int i = 2, j = i;
if (Convert.ToBoolean((i | j & 5) & (j - 25 * 1)))
    Console.WriteLine(1);
else
    Console.WriteLine(0);
```

- A. 0
- B. 1
- C. Ошибка компиляции
- D. Ошибка времени выполнения

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

```
switch (id)
{
    case 6:
        grp = "Grp B";
        break;

    case 13:
        grp = "Grp D";
        break;

    case 1:
```



```

        grp = "Grp A";
        break;

    case ls > 20:
        grp = "Grp E";
        break ;

    case Else:
        grp = "Grp F";
        break;
}

```

- A. Компилятор сообщит об ошибке в *case ls > 20*, а также в *case Else*.
- B. В этой инструкции *switch* нет ошибок.
- C. Компилятор сообщит об ошибке только в *case Else*.
- D. Компилятор будет сообщать об ошибке так как отсутствует ветвь *default*.
- E. Порядком первых трех случаев должен быть *case 1*, *case 6*, *case 13* (по возрастанию).

114. 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

115. 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

116. Каким будет результат выполнения фрагмента кода C#, приведенного ниже?

```
byte b1 = 0xF7;  
byte b2 = 0xAB;  
byte temp;  
temp = (byte)*(b1 & b2);  
Console.Write(temp + " ");  
temp = (byte)*(b1^b2);  
Console.WriteLine(temp);
```

- A. 163 92
- B. 92 163
- C. 192 63
- D. 0 1

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

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

118. 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

119. Что из нижеперечисленного НЕ является побитовым оператором в C#?

- A. &
- B. |
- C. <<
- D. ^
- E. ~

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

```
int d;  
d = Convert.ToInt32 (! (30<20) );
```

- A. Переменной d будет присвоено значение 0.
- B. Переменной d будет присвоено значение 1.
- C. Переменной d будет присвоено значение -1.
- D. Будет выдано сообщение об ошибке.
- E. Фрагмент кода будет работать правильно, если ! будет заменен на *Not*.

121. 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

122. Что из нижеперечисленного является логическими операторами в C#?

- 1. &&
- 2. ||
- 3. !
- 4. Xor
- 5. %

- A. 1, 2, 3
- B. 1, 3, 4
- C. 2, 4, 5

D. 3, 4, 5

E. Ничего из вышеперечисленного

123. Каким будет результат выполнения фрагмента кода C#?

```
int num = 1, z = 5;  
if (!(num <= 0))  
    Console.WriteLine( ++num + z++ + " " + ++z );  
else  
    Console.WriteLine( --num + z-- + " " + --z );
```

A. 5 6

B. 6 5

C. 6 6

D. 7 7

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

A. W=

B. /=

C. *=

D. +=

E. %=

125. Каким будет результат выполнения фрагмента кода C#?

```
int i, j = 1, k;  
for (i = 0; i < 5; i++)  
{  
    k = j++ + ++j;  
    Console.Write(k + " ");  
}
```

A. 8 4 16 12 20

B. 4 8 12 16 20

C. 4 8 16 32 64

D. 2 4 6 8 10

126. Каким будет результат выполнения фрагмента кода C#?

```
int a = 10, b = 20, c = 30;  
  
int res = a < b ? a < c ? c : a : b;  
  
Console.WriteLine(res);
```

- A. 10
- B. 20
- C. 30.
- D. Ошибка компиляции / ошибка синтаксиса

127. Какое из следующих утверждений является правильным относительно следующего фрагмента кода?

```
int a = 10;  
int b = 20;  
bool c;  
c = !(a > b);
```

- 1. В фрагменте кода нет ошибки.
- 2. Будет выведено сообщение об ошибке, ! Может работать только с *int*.
- 3. c будет присвоено значение 1.
- 4. c будет присвоено значение True.
- 5. c будет присвоено значению False.

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

128. 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

129. Каким будет результат выполнения фрагмента кода C#, приведенного ниже?

```
byte b1 = 0xF7;  
byte b2 = 0xAB;  
byte temp;  
temp = (byte)*(b1 & b2);  
Console.Write(temp + " ");  
temp = (byte)*(b1^b2);  
Console.WriteLine(temp);
```

- A. 163 92
- B. 92 163
- C. 192 63
- D. 0 1

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

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

131. 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

132. Что из нижеперечисленного НЕ является побитовым оператором в C#?

- A. &
- B. |
- C. <<
- D. ^
- E. ~

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

```
int d;  
d = Convert.ToInt32 (! (30<20)) ;
```

- A. Переменной d будет присвоено значение 0.
- B. Переменной d будет присвоено значение 1.
- C. Переменной d будет присвоено значение -1.
- D. Будет выдано сообщение об ошибке.
- E. Фрагмент кода будет работать правильно, если ! будет заменен на *Not*.

134. 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

135. Что из нижеперечисленного является логическими операторами в C#?

- 1. &&
- 2. ||
- 3. !
- 4. Xor
- 5. %

- A. 1, 2, 3
- B. 1, 3, 4
- C. 2, 4, 5

D. 3, 4, 5

E. Ничего из вышеперечисленного

136. Каким будет результат выполнения фрагмента кода C#?

```
int num = 1, z = 5;  
if (!(num <= 0))  
    Console.WriteLine( ++num + z++ + " " + ++z );  
else  
    Console.WriteLine( --num + z-- + " " + --z );
```

A. 5 6

B. 6 5

C. 6 6

D. 7 7

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

A. \=

B. /=

C. *=

D. +=

E. %=

138. Каким будет результат выполнения фрагмента кода C#?

```
int i, j = 1, k;  
for (i = 0; i < 5; i++)  
{  
    k = j++ + ++j;  
    Console.Write(k + " ");  
}
```

A. 8 4 16 12 20

B. 4 8 12 16 20

C. 4 8 16 32 64

D. 2 4 6 8 10

140. Каким будет результат выполнения фрагмента кода C#?

```
int a = 10, b = 20, c = 30;  
  
int res = a < b ? a < c ? c : a : b;  
  
Console.WriteLine(res);
```


- A. 10
- B. 20
- C. 30.
- D. Ошибка компиляции / ошибка синтаксиса

141. Какое из следующих утверждений является правильным относительно следующего фрагмента кода?

```
int a = 10;  
int b = 20;  
bool c;  
c = !(a > b);
```

- 1. В фрагменте кода нет ошибки.
- 2. Будет выведено сообщение об ошибке, ! Может работать только с *int*.
- 3. *c* будет присвоено значение 1.
- 4. *c* будет присвоено значение True.
- 5. *c* будет присвоено значению False.

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

**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 по введенному трех- или четырехзначному положительному числу, представляющему собой год, определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются). 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, 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
16. 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
17. 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

18. 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
19. 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
20. 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.
21. 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.
22. 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.
23. Разработать консольное приложение, которое по номеру года показывает название соответствующего года по восточному календарю. Вывод должен осуществляться по правилам русского языка. Например: "Год чёрной змеи", "Год чёрного тигра" и т.д. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных.¹
24. Разработать консольное приложение, позволяющее при вводе трех целых чисел, которые являются сторонами некоторого треугольника, определить к какому типу относится данный треугольник; равнобедренный, равнобедренный, равнобедренный, равнобедренный, 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

¹ Описание восточного календаря см. <http://www.whitemouse.ru/calendar/oriental.wmb>

возможность обработки исключительных ситуаций.

25. Разработать консольное приложение выводящее название карты по введенным двум числам: масти и достоинству карты. Мастям игральных 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.
26. Разработать консольное приложение, которое по номеру года показывает название соответствующего года по восточному календарю. Вывод должен осуществляться по правилам русского языка. Например: "Год чёрной змеи", "Год чёрного тигра" и т.д. Выводимые данные должны формироваться из массивов названий животных и названий цвета года. В массиве с названиями цвета года допускается иметь все значения необходимые и для формирования правильного вывода. Например: "красной", "красного" т.д. В массиве с названиями животных допускается иметь все значения необходимые и для формирования правильного вывода. Например: "змея", "змеи", "тигр", "тигра и т.д. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных.

	Дерево (сине-зелёный)	Огонь (красный)	Земля (жёлтый)	Металл (белый)	Вода (черный)
крыса	1924, 1984, 2044	1936, 1996, 2056	1888, 1948, 2008	1900, 1960, 2020	1912, 1972, 2032
бык	1925, 1985, 2045	1937, 1997, 2057	1889, 1949, 2009	1901, 1961, 2021	1913, 1973, 2033
тигр	1914, 1974, 2034	1926, 1986, 2046	1938, 1998, 2058	1890, 1950, 2010	1902, 1962, 2022
кролик	1915, 1975, 2035	1927, 1987, 2047	1939, 1999, 2059	1891, 1951, 2011	1903, 1963, 2023
дракон	1904, 1964, 2024	1916, 1976, 2036	1928, 1988, 2048	1940, 2000, 2060	1892, 1952, 2012
змея	1905, 1965, 2025	1917, 1977, 2037	1929, 1989, 2049	1941, 2001, 2061	1893, 1953, 2013
лошадь	1894, 1954, 2014	1906, 1966, 2026	1918, 1978, 2038	1930, 1990, 2050	1942, 2002, 2062
овца	1895, 1955, 2015	1907, 1967, 2027	1919, 1979, 2039	1931, 1991, 2051	1943, 2003, 2063
обезьяна	1944, 2004, 2064	1896, 1956, 2016	1908, 1968, 2028	1920, 1980, 2040	1932, 1992, 2052
петух	1945, 2005, 2065	1897, 1957, 2017	1909, 1969, 2029	1921, 1981, 2041	1933, 1993, 2053
собака	1934, 1994, 2054	1946, 2006, 2066	1898, 1958, 2018	1910, 1970, 2030	1922, 1982, 2042
свинья	1935, 1995, 2055	1947, 2007, 2067	1899, 1959, 2019	1911, 1971, 2031	1923, 1983, 2043

Рис. 1. Упрощённая таблица соответствия годов (1888...2067) и ветвей (<http://ru.wikipedia.org>).

27. Разработать консольное приложение, которое по названию животного из восточного календаря, показывает все года, которые соответствуют этому животному. Предусмотреть два варианта вывода: по три значения года в строке и по пять значений года в строке. Для

- определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть две возможности ввода названия животного и варианта вывода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
28. Разработать консольное приложение, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному и цвет соответствующего года. Предусмотреть три варианта вывода: значения года в строке и рядом цвета года; по два значения года в строке и снизу вывод цвета года; по три значения года в строке и снизу вывод цвета года. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть две возможности ввода названия животного и варианта вывода: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
29. Разработать консольное приложение, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному, также начало и конец соответствующего года по восточному (лунному) календарю. Вариант вывода результатов: название животного в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть две возможности ввода названия животного: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.

Первые две цифры года: 19

31.01.00	19.02.01	08.02.02	29.01.03	16.02.04	04.02.05	25.01.06	13.02.07	02.02.08	22.01.09
10.02.10	30.01.11	18.02.12	06.02.13	26.01.14	14.02.15	03.02.16	23.01.17	11.02.18	01.02.19
20.02.20	08.02.21	28.01.22	16.02.23	05.02.24	24.01.25	13.02.26	02.02.27	23.01.28	10.02.29
30.01.30	17.02.31	06.02.32	26.01.33	14.02.34	04.02.35	24.01.36	11.02.37	31.01.38	19.02.39
08.02.40	27.01.41	15.02.42	05.02.43	25.01.44	13.02.45	02.02.46	22.01.47	10.02.48	29.01.49
17.02.50	06.02.51	27.01.52	14.02.53	03.02.54	24.01.55	12.02.56	31.01.57	18.02.58	08.02.59
28.01.60	15.02.61	05.02.62	25.01.63	13.02.64	02.02.65	21.01.66	09.02.67	30.01.68	17.02.69
06.02.70	27.01.71	15.02.72	03.02.73	23.01.74	11.02.75	31.01.76	18.02.77	07.02.78	10.02.79
16.02.80	05.02.81	25.01.82	13.02.83	02.02.84	20.02.85	09.02.86	29.01.87	17.02.88	06.02.89
27.01.90	15.02.91	04.02.92	23.01.93	10.02.94	31.01.95	19.02.96	07.02.97	28.01.98	16.02.99

Рис.2 Точные даты начала года по лунному календарю для XIX века. Даты приведены в формате ДД.ММ.ГГ, первые 2 цифры года не указаны.

30. Разработать консольное приложение, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному) календарю. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть две возможности ввода названия животного цвета года: через командную строку и используя класс Console, а также возможность обработки исключительных ситуаций.
31. Разработать приложение Windows Forms, 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 возможность обработки исключительных ситуаций.
32. Разработать приложение Windows Forms, 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 возможность обработки исключительных ситуаций.
33. Разработать приложение Windows Forms, позволяющее по введенному трех- или четырехзначному положительному числу, представляющему собой год, определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые

- делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются). Предусмотреть возможность обработки исключительных ситуаций.
34. Разработать приложение Windows Forms, позволяющее по введенному трех- или четырехзначному положительному числу, представляющему собой год, определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются). Предусмотреть возможность обработки исключительных ситуаций.
35. Разработать приложение Windows Forms, позволяющее при вводе целого числа, лежащего в диапазоне 1-12, выводить строку с названием времени года, соответствующего данному месяцу. Предусмотреть возможность обработки исключительных ситуаций.
36. Разработать приложение Windows Forms, 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 возможность обработки исключительных ситуаций.
37. Разработать приложение Windows Forms, выводящее название карты по введенным двум числам: масти и достоинству карты. Для ввода чисел использовать текстовое поле. Мастям игральных 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.
38. Разработать приложение Windows Forms, которое позволяет ввести сторону квадрата, вычислить периметр этого квадрата и отобразить его графически. Предусмотреть возможность изменения цвета линий построенного квадрата.
39. Разработать приложение Windows Forms, которое позволяет ввести сторону квадрата, вычислить площадь этого квадрата и отобразить его графически. Предусмотреть возможность изменять цвет области заполнения квадрата.
40. Разработать приложение Windows Forms, которое позволяет ввести длину ребра куба a и вычислить объем куба V и площадь его поверхности S , отобразить куб графически. Размер отображаемого куба должен изменяться с изменением длины ребра.
41. Разработать приложение Windows Forms, которое позволяет вводить длины ребер a , b , c прямоугольного параллелепипеда и вычислять его объем V и площадь его поверхности S и отобразить параллелепипед графически. Размер отображаемого параллелепипеда должен

изменяться с изменением длины ребер.

42. Разработать приложение Windows Forms, позволяющее вводить длину окружности L и находить радиус R и площадь S круга, ограниченного этой окружностью. Приложение должно рисовать окружность на основе вычисленного радиуса и радиус этой окружности (30 баллов).
43. Разработать приложение Windows Forms, позволяющее по введенному значению площади S круга, находить его диаметр D и длину L окружности. Приложение должно рисовать окружность на основе вычисленного диаметра и диаметр для этой окружности.
44. Разработать приложение Windows Forms, 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 возможность обработки исключительных ситуаций.
45. Разработать приложение Windows Forms, 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 возможность обработки исключительных ситуаций.
46. Разработать приложение Windows Forms, выводящее название карты по введенным двум числам: масти и достоинству карты. Для ввода чисел использовать раскрывающиеся списки. Мастям игровых карт присвоить порядковые номера: 1 — червы, 2 — бубны, 3 — трефы, 4 — пики. 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.
47. Разработать приложение Windows Forms, выводящее название карты. Для выбора масти использовать кнопки, с соответствующим изображением масти, а достоинство карты вводить, используя раскрывающийся список. Мастям игровых карт присвоить порядковые номера: 1 — трефы, 2 — пики, 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.

	Дерево (сине-зелёный)	Огонь (красный)	Земля (жёлтый)	Металл (белый)	Вода (чёрный)
крыса	1924, 1984, 2044	1936, 1996, 2056	1888, 1948, 2008	1900, 1960, 2020	1912, 1972, 2032
бык	1925, 1985, 2045	1937, 1997, 2057	1889, 1949, 2009	1901, 1961, 2021	1913, 1973, 2033
тигр	1914, 1974, 2034	1926, 1986, 2046	1938, 1998, 2058	1890, 1950, 2010	1902, 1962, 2022
кролик	1915, 1975, 2035	1927, 1987, 2047	1939, 1999, 2059	1891, 1951, 2011	1903, 1963, 2023
дракон	1904, 1964, 2024	1916, 1976, 2036	1928, 1988, 2048	1940, 2000, 2060	1892, 1952, 2012
змея	1905, 1965, 2025	1917, 1977, 2037	1929, 1989, 2049	1941, 2001, 2061	1893, 1953, 2013
лошадь	1894, 1954, 2014	1906, 1966, 2026	1918, 1978, 2038	1930, 1990, 2050	1942, 2002, 2062
овца	1895, 1955, 2015	1907, 1967, 2027	1919, 1979, 2039	1931, 1991, 2051	1943, 2003, 2063
обезьяна	1944, 2004, 2064	1896, 1956, 2016	1908, 1968, 2028	1920, 1980, 2040	1932, 1992, 2052
петух	1945, 2005, 2065	1897, 1957, 2017	1909, 1969, 2029	1921, 1981, 2041	1933, 1993, 2053
собака	1934, 1994, 2054	1946, 2006, 2066	1898, 1958, 2018	1910, 1970, 2030	1922, 1982, 2042
свинья	1935, 1995, 2055	1947, 2007, 2067	1899, 1959, 2019	1911, 1971, 2031	1923, 1983, 2043

Рис. 1. Упрощённая таблица соответствия годов (1888...2067) и ветвей (<http://ru.wikipedia.org>).

48. Разработать приложение Windows Forms, которое по номеру года показывает название соответствующего года по восточному календарю. Ввод номера года выполнить, используя текстовое поле. Вывод должен осуществляться по правилам русского языка. Например: "Год чёрной змеи", "Год чёрного тигра" и т.д. Для определения типа соответствующего года использовать данные из рис. 1. Для каждого вывода должна показываться картинка с изображением соответствующего животного.
49. Разработать приложение Windows Forms, которое по номеру года показывает название соответствующего года по восточному календарю. Для ввода номера года использовать элемент управления Windows, который позволяет выбрать дату с помощью визуального календаря на месяц. Вывод должен осуществляться по правилам русского языка. Например: "Год чёрной змеи", "Год чёрного тигра" и т.д. Для определения типа соответствующего года использовать данные из рис. 1. Для каждого вывода должна показываться картинка с изображением соответствующего животного.
50. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, показывает все года, которые соответствуют этому животному. Для ввода названия животного использовать текстовое поле. Предусмотреть два варианта вывода: по три значения года в строке и по пять значений года в строке. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
51. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, показывает все года, которые соответствуют этому животному. Для ввода названия животного использовать раскрывающийся список. Предусмотреть два варианта

- вывода: по три значения года в строке и по пять значений года в строке. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
52. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, показывает все года, которые соответствуют этому животному. Для выбора названия животного использовать именованные радиокнопки. Предусмотреть два варианта вывода: по три значения года в строке и по пять значений года в строке. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
53. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, показывает все года, которые соответствуют этому животному. Для выбора названия животного использовать кнопки с изображениями животных. Предусмотреть два варианта вывода: по три значения года в строке и по пять значений года в строке. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
54. Разработать приложение Windows Forms, выводящее название карты. Для выбора масти использовать радиокнопки, а достоинство карты вводить, используя регулятор Windows (элемент управления "вверх-вниз"), отображающий числовые значения. Мастям игровых карт присвоить порядковые номера: 0 — трефы, 1 — бубны, 2 — пики, 3 — 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.
55. Разработать приложение Windows Forms, выводящее название карты. Для выбора масти использовать кнопки, с соответствующим изображением масти, а достоинство карты вводить, используя регулятор Windows (элемент управления "вверх-вниз"), отображающий числовые значения. Мастям игровых карт присвоить порядковые номера: 0 — трефы, 1 — червы, 2 — бубны, 3 — пики. 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.
56. Разработать приложение Windows Forms, выводящее название карты. Для выбора масти использовать радиокнопки, а достоинство карты вводить, используя раскрывающийся

- список. Мастям игральных карт присвоить порядковые номера: 0 — бубны, 1 — трефы, 2 — пики, 3 — 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 возможность обработки исключительных ситуаций или аргументировать, что для данной задачи исключительных ситуаций возникать не может.
57. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному и цвет соответствующего года. Для ввода названия животного использовать текстовое поле. Предусмотреть три варианта вывода: значения года в строке и рядом цвета года; по два значения года в строке и снизу вывод цвета года; по три значения года в строке и снизу вывод цвета года. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
58. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному и цвет соответствующего года. Для ввода названия животного использовать раскрывающийся список. Предусмотреть три варианта вывода: значения года в строке и рядом цвета года; по два значения года в строке и снизу вывод цвета года; по три значения года в строке и снизу вывод цвета года. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
59. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному и цвет соответствующего года. Для выбора названия животного использовать именованные радиокнопки. Предусмотреть три варианта вывода: значения года в строке и рядом цвета года; по два значения года в строке и снизу вывод цвета года; по три значения года в строке и снизу вывод цвета года. Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.
60. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному и цвет соответствующего года. Для выбора названия животного использовать кнопки с изображениями животных. Предусмотреть три варианта вывода: значения года в строке и рядом цвета года; по два значения года в строке и снизу вывод цвета года; по три значения года в строке и снизу вывод цвета года. Для определения типа соответствующего года

использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. Предусмотреть возможность обработки исключительных ситуаций.

Первые две цифры года: 19

31.01.00	19.02.01	08.02.02	29.01.03	16.02.04	04.02.05	25.01.06	13.02.07	02.02.08	22.01.09
10.02.10	30.01.11	18.02.12	06.02.13	26.01.14	14.02.15	03.02.16	23.01.17	11.02.18	01.02.19
20.02.20	08.02.21	28.01.22	16.02.23	05.02.24	24.01.25	13.02.26	02.02.27	23.01.28	10.02.29
30.01.30	17.02.31	06.02.32	26.01.33	14.02.34	04.02.35	24.01.36	11.02.37	31.01.38	19.02.39
08.02.40	27.01.41	15.02.42	05.02.43	25.01.44	13.02.45	02.02.46	22.01.47	10.02.48	29.01.49
17.02.50	06.02.51	27.01.52	14.02.53	03.02.54	24.01.55	12.02.56	31.01.57	18.02.58	08.02.59
28.01.60	15.02.61	05.02.62	25.01.63	13.02.64	02.02.65	21.01.66	09.02.67	30.01.68	17.02.69
06.02.70	27.01.71	15.02.72	03.02.73	23.01.74	11.02.75	31.01.76	18.02.77	07.02.78	10.02.79
16.02.80	05.02.81	25.01.82	13.02.83	02.02.84	20.02.85	09.02.86	29.01.87	17.02.88	06.02.89
27.01.90	15.02.91	04.02.92	23.01.93	10.02.94	31.01.95	19.02.96	07.02.97	28.01.98	16.02.99

Рис.2 Точные даты начала года по лунному календарю для XIX века.
Даты приведены в формате ДД.ММ.ГГ, первые 2 цифры года не указаны.

Первые две цифры года: 20

05.02.00	24.01.01	12.02.02	01.02.03	22.01.04	09.02.05	29.01.06	18.02.07	07.02.08	26.01.09
14.02.10	03.02.11	23.01.12	10.02.13	31.01.14	19.02.15	08.02.16	28.01.17	16.02.18	05.02.19
25.01.20	12.02.21	01.02.22	22.01.23	10.02.24	29.01.25	17.02.26	06.02.27	26.01.28	13.02.29
03.02.30	23.01.31	11.02.32	31.01.33	19.02.34	08.02.35	28.01.36	15.02.37	04.02.38	24.01.39
12.02.40	01.02.41	22.01.42	10.02.43	30.01.44	17.02.45	06.02.46	26.01.47	14.02.48	02.02.49
23.01.50	11.02.51	01.02.52	19.02.53	08.02.54	28.01.55	15.02.56	04.02.57	24.01.58	12.02.59
02.02.60	21.01.61	09.02.62	29.01.63	17.02.64	05.02.65	26.01.66	14.02.67	03.02.68	23.01.69
11.02.70	31.01.71	19.02.72	07.02.73	27.01.74	15.02.75	05.02.76	24.01.77	12.02.78	02.02.79
22.01.80	09.02.81	29.01.82	17.02.83	06.02.84	26.01.85	14.02.86	03.02.87	24.01.88	10.02.89
30.01.90	18.02.91	07.02.92	27.01.93	15.02.94	05.02.95	25.01.96	12.02.97	01.02.98	21.01.99

Рис. 3. Точные даты начала года по лунному календарю для XX века.
Даты приведены в формате ДД.ММ.ГГ, первые 2 цифры года не указаны.

61. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному, также начало и конец соответствующего года по восточному (лунному) календарю. Для ввода названия животного использовать текстовое поле. Вариант вывода результатов: название животного в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же

рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.

62. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному, также начало и конец соответствующего года по восточному (лунному) календарю. Для ввода названия животного использовать раскрывающийся список. Вариант вывода результатов: название животного в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис.1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.
63. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному, также начало и конец соответствующего года по восточному (лунному) календарю. Для выбора названия животного использовать кнопки с изображениями животных. Вариант вывода результатов: название животного в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.
64. Разработать приложение Windows Forms, которое по названию животного из восточного календаря, определяет все года, которые соответствуют этому животному, также начало и конец соответствующего года по восточному (лунному) календарю. Для выбора названия животного использовать именованные радиокнопки. Вариант вывода результатов: название животного в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.
65. Разработать приложение Windows Forms, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному)

календарю. Для ввода названия животного использовать текстовое поле, а для цвета использовать именованные радиокнопки. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.

66. Разработать приложение Windows Forms, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному) календарю. Для ввода названия животного использовать раскрывающийся список, а для цвета использовать именованные радиокнопки. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.
67. Разработать приложение Windows Forms, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному) календарю. Для выбора названия животного использовать кнопки с изображениями животных, а для цвета использовать раскрывающийся список. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.
68. Разработать приложение Windows Forms, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному) календарю. Для выбора названия животного и цвета использовать именованные радиокнопки. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю

(ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.

69. Разработать приложение Windows Forms, которое по названию животного из восточного календаря и цвету года, определяет все года, которые соответствуют этому животному и заданному цвету, также начало и конец соответствующего года по восточному (лунному) календарю. Для ввода названия животного и цвета использовать раскрывающийся список. Вариант вывода результатов: название животного и цвет года в начале вывода, далее строки с номером года (ГГГГ), началом этого года по восточному календарю (ДД.ММ.ГГГГ) и окончанием года по восточному календарю (ДД.ММ.ГГГГ). Для определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных. На рис. 2 и 3 приведены точные даты начала года по лунному календарю для XIX и XX веков. Предусмотреть возможность обработки исключительных ситуаций.

Первые две цифры года: 20

05.02.00	24.01.01	12.02.02	01.02.03	22.01.04	09.02.05	29.01.06	18.02.07	07.02.08	26.01.09
14.02.10	03.02.11	23.01.12	10.02.13	31.01.14	19.02.15	08.02.16	28.01.17	16.02.18	05.02.19
25.01.20	12.02.21	01.02.22	22.01.23	10.02.24	29.01.25	17.02.26	06.02.27	26.01.28	13.02.29
03.02.30	23.01.31	11.02.32	31.01.33	19.02.34	08.02.35	28.01.36	15.02.37	04.02.38	24.01.39
12.02.40	01.02.41	22.01.42	10.02.43	30.01.44	17.02.45	06.02.46	26.01.47	14.02.48	02.02.49
23.01.50	11.02.51	01.02.52	19.02.53	08.02.54	28.01.55	15.02.56	04.02.57	24.01.58	12.02.59
02.02.60	21.01.61	09.02.62	29.01.63	17.02.64	05.02.65	26.01.66	14.02.67	03.02.68	23.01.69
11.02.70	31.01.71	19.02.72	07.02.73	27.01.74	15.02.75	05.02.76	24.01.77	12.02.78	02.02.79
22.01.80	09.02.81	29.01.82	17.02.83	06.02.84	26.01.85	14.02.86	03.02.87	24.01.88	10.02.89
30.01.90	18.02.91	07.02.92	27.01.93	15.02.94	05.02.95	25.01.96	12.02.97	01.02.98	21.01.99

Рис. 3. Точные даты начала года по лунному календарю для XX века. Даты приведены в формате ДД.ММ.ГГ, первые 2 цифры года не указаны.

70. Разработать консольное приложение, которое по номеру года показывает название соответствующего года по восточному календарю. Выводимые данные должны формироваться из массивов с названиями животных и названиями цвета года. Массивы должны содержать названия животных и цвета года только в именительном падеже. Например, "крыса", "тигр", "красный", "белый" и т.д. Вывод должен осуществляться по правилам русского языка. Например: "Год чёрной змеи", "Год чёрного тигра" и т.д. Для

определения типа соответствующего года использовать данные из рис. 1. Этот же рисунок определяет диапазон доступных для ввода данных.

71. Разработать класс Параллелепипед, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении.
72. Разработать класс Конус, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении.
73. Разработать класс Шар, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении.
74. Разработать класс Куб, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении.
75. Разработать класс Фигура3D и создать дочерние классы Куб и Параллелепипед. Учесть все возможные действия, которые могут выполняться с данными фигурами. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этих классов в консольном приложении.
76. Разработать класс Треугольник и создать дочерние классы ПрямоугольныйТреугольник, РавностороннийТреугольник и РавнобедренныйТреугольник. Учесть все возможные действия, которые могут выполняться с данными фигурами. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этих классов в консольном приложении.
77. Разработать класс Окружность, который является потомком класса Фигура2D, определить все необходимые методы для класса Окружность. Продемонстрировать использование созданного класса Окружность.
78. Разработать класс Лодка, который позволит решать все возможные задачи на движение лодки. Варианты таких задач можно найти в учебниках по математике для 5-6 класса.
79. Разработать класс КвадратноеУравнение, позволяющий решать любое квадратное уравнение. Разработать тесты позволяющие проверить правильность данного класса.

Задачи по мотивам русских народных сказок

80. Разработать класс Репка, используя для разработки одноименную сказку. Продемонстрировать работу данного класса. Результат вывода сказки должен удовлетворять правилам русского языка.

```

Посадил дед репку
Выросла репка большая-пребольшая
Дед тянул, тянул – не вытянул репку
Позвал дед бабку
Дед и бабка тянули, тянули – не вытянули репку
Позвали внучку
Дед, бабка и внучка тянули, тянули – не вытянули репку
Позвали Жучку
Дед, бабка, внучка и Жучка тянули, тянули – не вытянули репку
Позвали кошку
Дед, бабка, внучка, Жучка и кошка тянули, тянули – не вытянули репку
Позвали мышку
Дед, бабка, внучка, Жучка и кошка тянули, тянули – вытянули репку
Вот и сказки конец
Для продолжения нажмите любую клавишу . . .

```

Рис. 4. Результат работы класса Репка.

81. Разработать класс Колобок, используя для разработки одноименную сказку.

Продемонстрировать работу данного класса. Результат вывода сказки должен удовлетворять правилам русского языка.

82. Разработать класс Яичко, используя для разработки сказку Курочка Ряба.

Продемонстрировать работу данного класса. Результат вывода сказки должен удовлетворять правилам русского языка.

83. Разработать класс Репка, используя для разработки одноименную сказку. Предусмотреть возможность участия в сказке других героев (см. рис.5 и 6). Продемонстрировать работу данного класса. Результат вывода сказки должен удовлетворять правилам русского языка.

```

Посадил дед репку
Выросла репка большая-пребольшая
Дед тянул, тянул – не вытянул репку
Позвал дед бабку
Дед и бабка тянули, тянули – не вытянули репку
Позвали внучку
Дед, бабка и внучка тянули, тянули – не вытянули репку
Позвали Жучку
Дед, бабка, внучка и Жучка тянули, тянули – не вытянули репку
Позвали кошку
Дед, бабка, внучка, Жучка и кошка тянули, тянули – не вытянули репку
Позвали мышку
Дед, бабка, внучка, Жучка и кошка и мышка тянули, тянули – вытянули репку
Вот и сказки конец
Для продолжения нажмите любую клавишу . . .

```

Рис. 5. Результат работы класса Репка, учитывающего возможность ввода любых данных.

```

Посадил Дима репку
Выросла репка большая-пребольшая
Дима тянул, тянул – не вытянул репку
Позвал Дима Наташу
Дима и Наташа тянули, тянули – не вытянули репку
Позвали Машу
Дима, Наташа и Маша тянули, тянули – не вытянули репку
Позвали Дашу
Дима, Наташа, Маша и Даша тянули, тянули – не вытянули репку
Позвали Мишу
Дима, Наташа, Маша, Даша и Миша тянули, тянули – не вытянули репку
Позвали Вову
Дима, Наташа, Маша, Даша и Миша и Вова тянули, тянули – вытянули репку
Вот и сказки конец
Для продолжения нажмите любую клавишу . . .

```

Рис. 6. Результат работы класса Репка, учитывающего возможность ввода любых данных.

84. Разработать класс Pascal, который содержит методы, формирующие треугольник Паскаля, а также методы выполняющие вывод этого треугольника.

В книге Мартина Гарднера «Математические новеллы» есть такие слова о треугольнике Паскаля:

Треугольник Паскаля так прост, что выписать его сможет даже десятилетний ребенок. В тоже время он таит в себе неисчерпаемые сокровища и связывает воедино различные аспекты математики, не имеющие на первый взгляд между собой ничего общего. Столь необычные свойства позволяют считать треугольник Паскаля одной из наиболее изящных схем во всей математике.

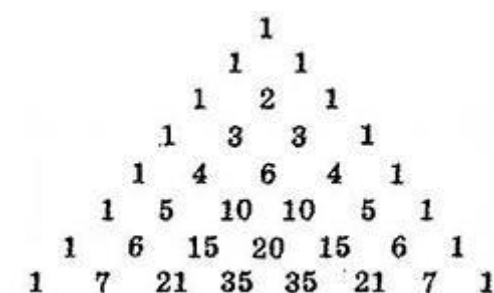


Рис. 7. Изображение треугольника Паскаля.

85. Разработать класс Репка, используя для разработки одноименную сказку. Предусмотреть возможность участия в сказке других героев (см. рис.5 и 6). Цепочку Дед – баба – внука и т.д. реализовать в виде связанного списка, разработав соответствующий класс. Показывать

- значения этого списка перед каждым его изменением. Продемонстрировать работу класса Репка. Результат вывода сказки должен удовлетворять правилам русского языка.
86. Разработать Web-приложение с использованием HTML5, CSS3 и JavaScript позволяющее ввести сторону квадрата, вычислить периметр этого квадрата, вывести его на странице и отобразить квадрат графически (30 баллов).
 87. Разработать Web-приложение с использованием HTML5, CSS3 и JavaScript, позволяющее ввести стороны прямоугольника, вычислить периметр и площадь этого прямоугольника, вывести эти данные на странице и отобразить прямоугольник графически (30 баллов).
 88. Разработать класс Параллелепипед, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении (15 баллов).
 89. Разработать класс Куб, учитывающий все возможные действия, которые могут выполняться с данной фигурой. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этого класса в консольном приложении (15 баллов).
 90. Разработать класс Фигура3D и создать дочерние классы Куб и Параллелепипед. Учесть все возможные действия, которые могут выполняться с данными фигурами. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этих классов в консольном приложении (20 баллов).
 91. Разработать класс Треугольник и создать дочерние классы ПрямоугольныйТреугольник, РавностороннийТреугольник и РавнобедренныйТреугольник. Учесть все возможные действия, которые могут выполняться с данными фигурами. Предусмотреть возможность работы с различными типами данных. Продемонстрировать использование этих классов в консольном приложении (20 баллов).
 92. Разработать класс Окружность, который является потомком класса Фигура2D, определить все необходимые методы для класса Окружность. Продемонстрировать использование созданного класса Окружность (35 баллов).
 93. Разработать класс Лодка, который позволит решать все возможные задачи на движение лодки. Варианты таких задач можно найти в учебниках по математике для 5-6 класса.
 94. Разработать класс КвадратноеУравнение, позволяющий решать любое квадратное уравнение. Разработать тесты позволяющие проверить правильность данного класса.
 95. Используя стандартные методы класса String и структуры Char написать программу, которая после введения некоторого предложения, определяет все гласные буквы в данном предложении и подсчитывает их количество. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением

соответствующего комментария вокруг некоторого теста.

96. Используя стандартные методы класса String и структуры Char написать программу, которая после введения некоторого предложения, определяет все согласные буквы в данном предложении и подсчитывает их количество. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
97. Используя стандартные методы класса String и структуры Char написать программу, которая после введения некоторого предложения, определяет количество слов в данном предложении. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
98. Используя стандартные методы класса String и структуры Char написать программу, которая после введения некоторого предложения и ввода определенной согласной буквы, определяет количество вхождений введенной согласной буквы. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария.
99. Используя стандартные методы класса String и структуру Char написать программу, которая после введения некоторого предложения и ввода определенной гласной буквы, определяет количество вхождений введенной гласной буквы. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
100. Используя стандартные методы класса String и структуры Char написать программу, которая после введения некоторого предложения и ввода определенных гласной и согласной букв, определяет количество вхождений введенной гласной буквы. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
101. Разработать программу, которая выводит букву и рядом ее кодовое значение. Программа должна вывести все буквы русского алфавита. Используемые типы данных должны быть выбраны оптимально.
102. Разработать программу, которая выводит букву и рядом ее кодовое значение. Программа должна вывести все буквы английского алфавита. Используемые типы данных должны быть

выбраны оптимально.

103. Написать программу работы с классом String, которая показывает основные возможности данного класса:
- определять длину введенной строки;
 - переводить введенную строку в верхний регистр (прописные символы) или в нижний регистр (строчные буквы);
 - определять значение символа по определенному индексу;
 - выводить текущую строку;
 - добавлять к введенной строке другую строку;
 - извлекать из введенной строки подстроку;
 - определять находится ли указанный элемент в строке;
 - заменить вхождение некоторого символа в строке на другой символ;
 - и т.д..
104. Разработать программу, которая позволяет ввести адрес электронной почты (например, myemail@postemail.com) и выделяет имя пользователя (myemail), имя почтового сервера (postemail) и имя домена первого уровня (com) и выводит эти данные. Предусмотреть работу с со следующими доменами первого уровня: com, edu, ru, net, info. Программа должна обрабатывать ошибки:
- в именах пользователя и почтового сервера должны быть только алфавитно-цифровые символы;
 - другие имена доменов первого уровня, кроме com, edu, ru, net, info, не допускаются;
 - отсутствует символ @ в адресе электронной почты;
 - отсутствует точка между именем почтового сервера и доменом первого уровня.
105. Используя стандартные методы класса String и структуры Char разработать программу, которая подсчитывает количество слов во введенном предложении, определяет наиболее длинное слово, подсчитывает количество гласных и согласных букв в этом слове. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
106. Используя стандартные методы класса String и структуры Char разработать программу, которая подсчитывает количество цифр во введенном предложении, состоящем из чисел различной длины, определяет наиболее длинное число, записывает это число в обратном порядке. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария

- вокруг некоторого теста.
107. Используя стандартные методы класса String и структуры Char разработать программу, которая подсчитывает количество цифр во введенном предложении, состоящем из чисел различной длины, определяет сколько раз каждая из цифр (0 ... 9) входит во введенное предложение. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
108. Используя стандартные методы класса String и структуры Char разработать программу, которая для пяти последовательно введенных строк с отдельными словами, выводит их алфавитном порядке. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
109. Используя стандартные методы класса String и структуры Char разработать программу, которая подсчитывает количество слов во введенном предложении, определяет слова, которые имеют длину не меньше заданной. Разработать несколько тестов, которые проверяют работоспособность разработанной программы, тесты должны располагаться внутри разработанной программы и быть закомментированы. Проверка осуществляется удалением соответствующего комментария вокруг некоторого теста.
110. Разработать класс Таблица_Пифагора, предусматривающий возможность работы с **различными типами целых данных**. Продемонстрировать использование этого класса в консольном приложении.
111. Разработать класс Наименьшее_Общее_Кратное, предусматривающий возможность работы с **различными типами целых данных**. Продемонстрировать использование этого класса в консольном приложении.
112. Разработать класс Найти_Совершенное_Число, предусматривающий возможность работы с **различными типами целых данных**. Поиск чисел должен осуществляться на интервале заданном двумя числами M и N. Продемонстрировать использование этого класса в консольном приложении.
113. Разработать класс Поиск_Простых_Чисел, предусматривающий возможность работы с **различными типами целых данных**. Продемонстрировать использование этого класса в консольном приложении.
114. Разработать класс Разложение_Числа_на_простые_множители, предусматривающий

возможность работы с **различными типами целых данных**. Продемонстрировать использование этого класса в консольном приложении.

115. Разработать класс Поиск_Простых_Чисел, предусматривающий возможность работы с **различными типами целых данных**. Продемонстрировать использование этого класса в консольном приложении.
116. Вычислить константу Эйлера. Для вычисления использовать цикл While. Вычисления выполнять при $n = 1000, 2000, 10000, 32000$. Проанализировать полученные результаты.
117. Разработать класс След_Квадратичной_Матрицы, предусматривающий возможность работы с матрицами, содержащими **различные типы данных**. Предусмотреть два варианта вычисления. Продемонстрировать использование этого класса в консольном приложении.
118. Разработать класс Определитель_Треугольной_Матрицы, предусматривающий возможность работы с матрицами, содержащими **различные типы данных**. Продемонстрировать использование этого класса в консольном приложении.
119. Пусть функция $f(x)$ задана в виде формулы:

$$f(x) = \begin{cases} x + x^2/2! + x^3/3! + \dots + x^N/N! , & x \leq 0 , \\ \sin x + \sin^3 x/3 + \dots + \sin^{2M+1} x/(2M + 1) , & x > 0 , \end{cases}$$

Разработать программу, которая для заданных параметров N и M вычисляет значение функции для любого аргумента.

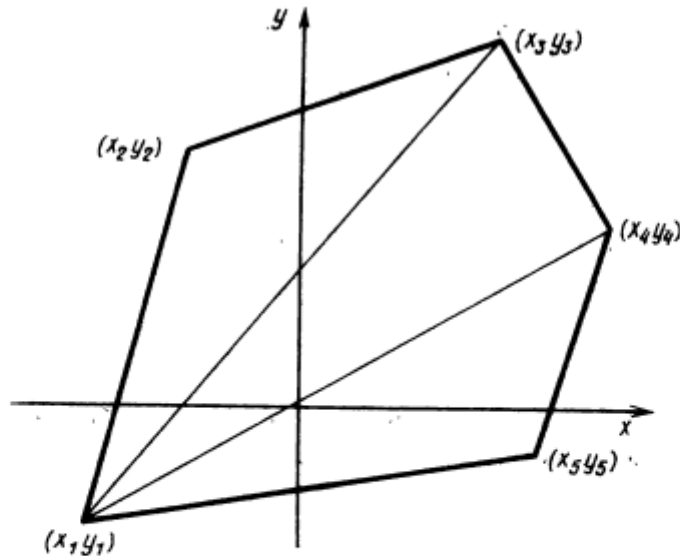


Рис. 1..

На плоскости заданы N точек с декартовыми координатами $X(I)$, $Y(I)$, $I=1, \dots, N$, $3 \leq N \leq 50$, которые являются последовательными вершинами выпуклого многоугольника, рис. 1. Написать программу, вычисляющую площадь этого многоугольника.

120. Разработать программу для вычисления и печати значений функции $y=2 \sin(x) + 1/\cos(x)$ при x , изменяющемся от 0 до 3,1415, с шагом 0,2. Вычисленные значения x и y сохранить в массивах. Организовать красивый и компактный вывод полученных данных.
121. Разработать программу для заполнения одномерного массива с 10 элементами, каждый элемент массива определяется по формуле

$$x_j = \begin{cases} \arctg \frac{\sqrt{j}+2}{n+3}, & \text{если } \lg j \leq 3, \\ e^j + \cos n, & \text{если } \lg j > 3, \end{cases}$$

1. $j = 0, 1, \dots, 10$; $n = 10$. Организовать красивый и компактный вывод полученного массива.

- 122.** Разработать программу для вычисления значений массива Z , равного сумме двух массивов X и Y по формуле $Z_i = X_i + Y_i$. Массив X задан и имеет следующие значения 0,1; 0,2; 0,3; 0,8; 1,6; -4,2; -0,4; 0,5; 0,8; 5,3. Значения массива Y вычисляются по формуле:
- $$Y_i = 0,7 \sin(0,5i) + 1,3 \cos(i+1).$$

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.WriteLine()` и `Console.Write()`?
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 зачету с оценкой (проводится в устной форме) в 8 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 9 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 позиционирований. 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	0,5

схемы	
-------	--

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