ПРАКТИЧЕСКАЯ РАБОТА № 12

(4 академических часа)

Тема: Классы и объекты

Цель задания:

Изучение понятия класса как пользовательского типа данных и приобретение навыков работы с классами.

Теоретическая часть.

C# является полноценным объектно-ориентированным языком, следовательно, программу на C# можно представить в виде взаимосвязанных взаимодействующих между собой объектов.

Описанием объекта является класс, а объект представляет экземпляр этого класса.

Класс определяется с помощью ключевого слова class:

```
class Book
{

}
```

Вся функциональность класса представлена его членами - полями (полями называются переменные класса), свойствами, методами, событиями.

```
class Book
{
    public string name;
   public string author;
   public int year;
   public void Info()
   {
      Console.WriteLine("Книга '{0}' (автор {1}) была издана в {2} году", name,
author, year);
   }
}
```

Кроме обычных методов в классах используются также и специальные методы, которые называются конструкторами. Конструкторы вызываются при создании нового объекта данного класса. Отличительной чертой конструктора является то, что его название должно совпадать с названием класса:

```
class Book
{
     public string name;
        public string author;
       public int year;

     public Book()
  { }    public Book(string name, string author, int year)
  {       this.name = name;
        this.author = author;
      this.year = year;
  }

  public void Info()
  {
```

```
        Console.WriteLine("Книга '{0}' (автор {1}) была издана в {2} году", name,
    author, year);
      }
    }
```

Одно из назначений конструктора - начальная инициализация членов класса. В данном случае мы использовали два конструктора. Один пустой. Второй конструктор наполняет поля класса начальными значениями, которые передаются через его параметры.
Поскольку имена параметров и имена полей (name, author, year) в данном случае совпадают, то мы используем ключевое слово this. Это ключевое слово представляет ссылку на текущий экземпляр класса. Поэтому в выражении this.name = name; первая часть this.name означает, что name - это поле текущего класса, а не название параметра name. Если бы у нас параметры и поля назывались по-разному, то использовать слово this было бы необязательно.
Теперь используем класс в программе. Создадим новый проект. Затем нажмем правой кнопкой мыши на название проекта в окне Solution Explorer (Обозреватель решений) и в появившемся меню выберем пункт Class.

В появившемся диалоговом окне дадим новому классу имя Book и нажмем кнопку Add (Добавить). В проект будет добавлен новый файл Book.cs, содержащий класс Book.
Изменим в этом файле код класса Book на следующий:

```
class Book
{
   public string name;
   public string author;
   public int year;

     public Book()
   {
      name = "неизвестно";
      author = "неизвестно";
       year = 0;
     }
    public Book(string name, string author, int year)
   {
      this.name = name;
      this.author = author;
      this.year = year;
   }

   public void GetInformation()
   {
      Console.WriteLine("Книга '{0}' (автор {1}) была издана в {2} году", name,
author, year);
   }
}
```

Теперь перейдем к коду файла Program.cs и изменим метод Main класса Program следующим образом:

```
class Program
{
  static void Main(string[] args)
   {
```

```
        Book b1 = new Book("Война и мир", "Л. Н. Толстой",
1869);
  b1.GetInformation();
  Book b2 = new Book();
  b2.GetInformation();
      Console.ReadLine();
   }
 }
```

Если мы запустим код на выполнение, то консоль выведет нам информацию о книгах b1 и b2. Обратите внимание, что чтобы создать новый объект с использованием конструктора, нам надо использовать ключевое слово new. Оператор new создает объект класса и выделяет для него область в памяти.

Инициализаторы объектов

Для нашего класса Book мы могли бы установить последовательно значения для всех трех полей класса:

```
Book b1 = new Book();
 b1.name = "Война и мир";
 b1.author = "Л. Н. Толстой";
b1.year = 1869;
b1.GetInformation();
```

Но можно также использовать инициализатор объектов:

```
Book b2 = new Book { name = "Отцы и дети", author = "И. С. Тургенев", year = 1862 };
b2.GetInformation();
```

С помощью инициализатора объектов можно присваивать значения всем доступным полям и свойствам объекта в момент создания без явного вызова конструктора.

ЗАДАНИЕ

Разработать класс, инкапсулирующий двумерный массив. Класс должен содержать поля и методы, необходимые для реализации индивидуального задания в соответствии с вариантом.

Вариант 1

Дана вещественная матрица размером 4 строки, 5 столбцов. Переставляя ее строки и столбцы, добейтесь того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.

Вариант 2

Определите, является ли заданная целочисленная квадратная матрица порядка 5 симметричной относительно главной диагонали.

Вариант 3

Дана вещественная матрица размером 4 строки, 5 столбцов. Поменяйте местами максимальный и минимальный элементы матрицы.

Вариант 4

Дана целочисленная квадратная матрица порядка 5. Найдите максимальный элемент среди элементов, лежащих ниже главной диагонали, и максимальный элемент среди элементов, лежащих выше главной диагонали, поменяйте их местами.

Вариант 5

Дана целочисленная квадратная матрица порядка 5. Найдите максимальный элемент среди элементов, лежащих левее вспомогательной диагонали, и максимальный элемент среди элементов, лежащих правее вспомогательной диагонали, поменяйте их местами.

Вариант 6

Среди строк целочисленной квадратной матрицы порядка 5 найдите строку с минимальной суммой элементов.

Вариант 7

Дана целочисленная квадратная матрица порядка 5. Найдите минимальный элемент среди элементов, лежащих ниже главной диагонали, и максимальный элемент среди элементов, лежащих выше главной диагонали, вычислите их среднее арифметическое.

Вариант 8

Дана целочисленная квадратная матрица порядка 5. Найдите минимальный элемент среди элементов, лежащих левее вспомогательной диагонали, и максимальный элемент среди элементов, лежащих правее вспомогательной диагонали, и вычислить их среднее геометрическое.

Вариант 9

Среди столбцов целочисленной квадратной матрицы порядка 5 найдите столбец с максимальной суммой элементов.

Вариант 10

Среди тех столбцов целочисленной матрицы размером 3 строки, 5 столбцов, которые содержат только такие элементы, значения которых по модулю не превышают 10, найдите столбец с минимальным произведением элементов.

Вариант 11

Даны целые числа a1,...,a10, целочисленная квадратная матрица порядка 5. Замените нулями в матрице те элементы, для которых имеются равные числа среди $a_1,...,a_{10}$.

Вариант 12

В двумерном целочисленном массиве размером 4 строки, 5 столбцов поменяйте местами строки, симметричные относительно середины массива (горизонтальной линии).

Вариант 13

В двумерном целочисленном массиве размером 4 строки, 5 столбцов поменяйте местами столбцы, симметричные относительно середины массива (вертикальной линии).

Вариант 14

Даны две вещественные квадратные матрицы порядка 4. Получите новую матрицу прибавлением к элементам каждого столбца первой матрицы минимального элемента соответствующего столбца второй матрицы.

Вариант 15

В целочисленной квадратной матрице порядка 4 найдите наибольший по модулю элемент. Получите квадратную матрицу порядка 3 путем выбрасывания из исходной матрицы строки и столбца, на пересечении которых расположен элемент с найденным значением.

Контрольные вопросы

1. Какие модификаторы доступа есть в C#?
2. Если классы и члены класса не имеют никаких модификаторов, какие модификаторы доступа к ним применяются по умолчанию?
3. Что выведет на консоль следующая программа и почему?

```
class Person
{
    int age = 26;
    string name = "Tom";

    public Person(int age, string name)
    {
        this.age = age;
        this.name = name;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Person person = new Person(19, "Bob");
        Console.WriteLine(person.name);

        Console.ReadKey();
    }
}
```

4. Дан следующий класс:

```
class Person
{
    public string name = "Sam";
    public int age;

    public Person(string name, int age)
    {
        this.name = name;
        this.age = age;
    }
}
```

Какое значение поле name будет иметь при выполнение следующего кода и почему?
Person tom = new Person("Tom", 34) { name = "Bob", age = 29 };