

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

По дисциплине Объектно-ориентированное программирование

Тема работы Создание и использование классов

Обучающийся Буров Глеб Максимович

Факультет факультет инфокоммуникационных технологий

Группа K3223

Направление подготовки 11.03.02 Инфокоммуникационные технологии и системы связи

Образовательная программа Программирование в инфокоммуникационных системах

Обучающийся

(дата)

(подпись)

Буров Г.М.
(Ф.И.О.)

Руководитель

(дата)

(подпись)

Иванов С.Е.
(Ф.И.О.)

СОДЕРЖАНИЕ

Стр.

ВВЕДЕНИЕ	3
1 Ход работы	4
1.1 Упражнение 1	4
1.2 Упражнение 2	5
1.3 Упражнение 3	7
ЗАКЛЮЧЕНИЕ	9

ВВЕДЕНИЕ

Целью данной лабораторной работы является изучение понятия класса как пользовательского типа данных и приобретение навыков работы с классами.

1 Ход работы

1.1 Упражнение 1

В первом упражнении нужно было создать класс Book с соответствующими полями и методами.

В процессе работы был создан класс Book с полями типа string: author (автор книги), title (название), publisher (издатель); и полями типа int: pages (количество страниц) и year (год издания). Также были реализованы метод SetBook, позволивший присвоить значения приватным полям, метод Show, отображающий информацию о книге, и метод PriceBook, вычисляющий стоимость аренды книги на указанное число дней (рис. 1.1).

```
internal class Book
{
    private String author;
    private String title;
    private String publisher;
    private int pages;
    private int year;
    private static double price = 9;

    Ссылка: 1
    public static void SetPrice(double price)
    {
        Book.price = price;
    }

    Ссылка: 1
    public void Show()
    {
        Console.WriteLine("\nКнига:\n Автор: {0}\n Название: {1} \n Год издания: {2}\n " +
            "{3} стр. \n Стоимость аренды: {4} р.", author, title, year, pages, Book.price);
    }

    Ссылка: 1
    public double PriceBook(int s)
    {
        double cust = s * price;
        return cust;
    }
}
```

Рисунок 1.1 — Код упражнения 1

В методе Main был создан экземпляр класса Book, были вызваны его методы (рис. 1.2).

```
static void Main(string[] args)
{
    Book b1 = new Book();
    b1.SetBook("Пушкин А.С.", "Капитанская дочка", "Вильямс", 123, 2012);
    Book.SetPrice(12);
    b1.Show();
    Console.WriteLine("\nИтоговая стоимость аренды: {0} р.", b1.PriceBook(3));
}

C:\Windows\system32\cmd.exe

Книга:
Автор: Пушкин А.С.
Название: Капитанская дочка
Год издания: 2012
123 стр.
Стоимость аренды: 12 р.

} Итоговая стоимость аренды: 36 р.
Для продолжения нажмите любую клавишу . . .
```

Рисунок 1.2 — Работа программы из упражнения 1

1.2 Упражнение 2

Во втором упражнении нужно было добавить конструкторы для инициализации объекта.

В класс `Book` были добавлены конструктор, инициализирующий все поля объекта, перегруженный конструктор, принимающий в качестве аргументов только автора и название книги, статический конструктор. Также был переопределен конструктор по умолчанию (рис. 1.3).

```

Ссылка: 1
public Book()
{
    ...
}

Ссылка: 0
static Book()
{
    price = 10;
}

Ссылка: 0
public Book(String author, String title, String publisher, int pages, int year)
{
    this.author = author;
    this.title = title;
    this.publisher = publisher;
    this.pages = pages;
    this.year = year;
}

Ссылка: 0
public Book(String author, String title)
{
    this.author = author;
    this.title = title;
}

```

Рисунок 1.3 — Код упражнения 2

В методе Main были созданы два новых экземпляра класса Book (рис. 1.4).

```

Book b2 = new Book("Толстой Л.Н.", "Война и мир", "Наука и жизнь", 1234, 2013);
b2.Show();

Book b3 = new Book("Лермонтов М.Ю.", "Мцыри");
b3.Show();

```

C:\Windows\system32\cmd.exe

```

Книга:
Автор: Толстой Л.Н.
Название: Война и мир
Год издания: 2013
1234 стр.
Стоимость аренды: 10 р.

Книга:
Автор: Лермонтов М.Ю.
Название: Мцыри
Год издания: 0
0 стр.
Стоимость аренды: 10 р.
Для продолжения нажмите любую клавишу . . .

```

Рисунок 1.4 — Пример работы программы из упражнения 2

1.3 Упражнение 3

В последнем элементе требовалось создать класс `Triangle`, разработав следующие элементы класса:

- Поля: стороны треугольника.
- Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
- Методы, позволяющие:
 - вывести длины сторон треугольника на экран;
 - рассчитать периметр треугольника;
 - рассчитать площадь треугольника;
 - реализовать проверку, позволяющую установить, существует ли треугольник с данными длинами сторон.

В процессе работы был создан класс `Triangle`, в котором были приватные поля типа `double`: `a`, `b`, `c` — стороны треугольника. Публичный метод `Exists` подобно тому, который был в одной из прошлых лабораторных работ, проверял, выполняется ли неравенство треугольника, а также корректные ли длины сторон.

Метод `ShowInfo` выводил на экран консоли информацию о сторонах треугольника, если сам треугольник существует. В противном случае выводится сообщения о том, что такого треугольника не существует.

Методы `Perimeter` и `Square` рассчитывают соответственно периметр и площадь треугольника, если он существует. Иначе возвращается значение `-1`.

Методы класса `Triangle` приведены на рис. 1.5.

```

Ссылка: 1
public void ShowInfo()
{
    if (this.Exists())
    {
        Console.WriteLine("Треугольник:\n a = {0}\n b = {1}\n c = {2}", a, b, c);
    }
    else
    {
        Console.WriteLine("Треугольника со сторонами {0}, {1} и {2} не существует", a, b, c);
    }
}

Ссылка: 1
public double Perimeter()
{
    double perimeter = -1;
    if (this.Exists())
    {
        perimeter = a + b + c;
    }
    return perimeter;
}

Ссылка: 0
public double Square()
{
    double square = -1;
    if (this.Exists())
    {
        double p = this.Perimeter() / 2;
        square = Math.Sqrt(p * (p - a) * (p - b) * (p - c));
    }
    return square;
}

Ссылка: 3
public bool Exists()
{
    return a > 0 && b > 0 && c > 0 && (a + b > c) && (a + c > b) && (b + c > a);
}

```

Рисунок 1.5 — Код упражнения 3

В методе Main был создан экземпляр класса Triangle и вызваны его методы (рис. 1.6).

```

Triangle tr = new Triangle(3, 4, 5);
tr.ShowInfo();
Console.WriteLine("Периметр треугольника: {0}", tr.Perimeter());
Console.WriteLine("Площадь треугольника: {0}", tr.Square());

```

C:\Windows\system32\cmd.exe

```

Треугольник:
a = 3
b = 4
c = 5
Периметр треугольника: 12
Площадь треугольника: 6
Для продолжения нажмите любую клавишу . . .

```

Рисунок 1.6 — Пример работы программы из упражнения 3

ЗАКЛЮЧЕНИЕ

В ходе этой лабораторной работы я ознакомился с внутренним устройством классов, полями, методами, их модификаторами доступа, конструкторами, понял различия между обычным конструктором и статическим конструктором. Все задания были выполнены.