

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждение высшего  
образования «Национальный исследовательский университет  
Московский институт электронной техники»

Институт Микроприборов и систем управления

НАПРАВЛЕНИЕ: 27.03.04 «Управление в технических системах»

ДИСЦИПЛИНА: Объектно-ориентированное программирование

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

«Классы и Объекты»

Долучен	02.11.22	Бобков
Воталаш	03.11.22	Бобков
Зачислен	03.11.22	Бобков

Работу выполнил  
студент гр. УТС-21

  
(подпись студента)

Р. В. Закшевский  
(Ф.И.О. студента)

Преподаватель

  
(подпись преподавателя)

В. Д. Бобков  
(Ф.И.О. преподавателя)

5 листов

Москва, 2022 г.

Цель работы: Научиться создавать и работать с классами.

Задача работы: Aeroflot: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:

- список рейсов для заданного пункта назначения;
- список рейсов для заданного дня недели;
- список рейсов для заданного дня недели, время вылета для которых больше заданного.

### Теоретическая часть

#### 1. Создаем класс и описываем поля данных

Для полей фамилия, имя, отчество, адрес выбираем тип данных строка. Для поля «год рождения» - целое число. Для полей пол и образование лучше всего подходит тип «перечисление». Определяем нужные типы:

```
public enum Sex {  
    Male,  
    Female  
};
```

```
public enum Education {  
    Midschool,  
    Highschool  
};
```

Обратите внимание, если вы описываете перечисления внутри класса Person, то вы должны объявить их публичными, для того чтобы эти типы можно было использовать извне.

После этого описываем поля данных, пока без методов:

```
class Person {  
  
    string lastname;           //фамилия  
    string name;              //имя  
    string patronymic;        //отчество  
    string address;  
    int birthYear;  
    Sex sex;  
    Education education;  
}
```

#### 2. Создаем конструктор класса

Для того чтобы класс был проинициализирован корректными значениями мы должны описать конструктор класса (возможно и не один).

```
public Person (string n, string l, string p, Sex s, int b){  
    name = n;  
    lastname = l;
```

```

    patronymic = p;
    sex = s;
    if(birthYear < 1900)
        throw new Exception("Некорректный год рождения!");
    birthYear = b;
}

```

В конструкторе необходимо обеспечить проверку корректности данных, и в случае некорректных данных выбрасывать исключение.

### 3. Создаем методы для записи полей данных

Для того чтобы можно было задать значения полей, не инициализирующихся в конструкторе, необходимо задать методы для установки значений, так называемые сеттеры (setters).

```

    public void setEducation (Education e){
        education = e;
    }

    public void setAddress (string a){
        address = a;
    }

```

И так далее. В сеттерах, при необходимости, нужно проверять валидность входных параметров и выбрасывать исключения в случае ошибки.

### 4. Создаем методы для чтения полей данных

Так как все поля класса объявлены приватными, для чтения их значений необходимо создать методы для получения их значений – так называемые геттеры (getters). Исходя из задания, нам необходимо вывести всех мужчин, всех граждан старше определенного возраста и всех имеющих высшее образование. Для этого можно определить следующие методы:

```

    public int getAge() {
        return 2015 - birthYear;
    }

    public Education getEducation() {
        return education;
    }

    public bool isMale() {
        return sex == Sex.Male;
    }

```

Обратите внимание, что в геттерах необязательно просто возвращать значение поля. Можно произвести вычисления и из года рождения получить возраст, сравнить с заданным значением и вернуть bool.

## 5. Вывод полной информации о гражданине

Для того чтобы можно было вывести на экран информацию об объекте, можно описать метод для вывода строки с этой информацией:

```
public string info() {  
  
    return String.Format("{0} {1} {2}\n\tВозраст:{3}\n\tОбразование: {4}",  
        name, patronymic, lastname, getAge(), education);  
}
```

## Практическая часть

Задание:

Worker: Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив объектов. Вывести:

- список работников, стаж работы которых на данном предприятии превышает заданное число лет;
- список работников, зарплата которых больше заданной;

список работников, занимающих заданную должность.

\*Примеры работы программы для нескольких значений представлены на рисунке

```
using System;  
using System.Text;  
  
namespace Lab3  
{  
    class Programm  
    {  
        public class Worker  
        {  
            private string m_surname = "";  
            public string surname  
            {  
                set  
                {  
                    if ((value != ""))  
                    {  
                        m_surname = value;  
                    }  
                    else  
                    {  
                        throw new Exception();  
                    }  
                }  
                get { return m_surname; }  
            }  
            private string m_name = "123";  
            public string name  
            {  
                set  
                {  
                    if ((value != ""))  
                    {  
                        m_name = value;  
                    }  
                }  
                get { return m_name; }  
            }  
        }  
    }  
}
```

```

private string m_patronymic = "";
public string patronymic
{
    set
    {
        if ((value != ""))
        {
            m_patronymic += value;
        }
    }
    get { return m_patronymic; }
}
private string m_jobrang = "";
public string jobrang
{
    set
    {
        if ((value != ""))
        {
            m_jobrang += value;
        }
        else
        {
            throw new Exception();
        }
    }
    get { return m_jobrang; }
}
private int m_salary;
public int salary
{
    set
    {
        if (value >= 0)
        {
            m_salary = value;
        }
        else
        {
            throw new Exception();
        }
    }
    get { return m_salary; }
}
private int m_year;
public int year
{
    set
    {
        if ((value >= 2000) && (value <= 2022))
            m_year = value;
        else
        {
            throw new Exception();
        }
    }
    get { return m_year; }
}
public Worker(string s, string n, string p, string j, string sal, string
y)
{
    try
    {
        surname = s;
        name = n;
    }
}

```

```

        patronymic = p;
        jobrang = j;
        salary = int.Parse(sal);
        year = int.Parse(y);
    }
    catch (Exception)
    {
        Console.WriteLine("Некорректный ввод. Повторите действие,
позаязь");
    }
}

}

public static void PrintArrWorker(Worker[] ArrWorker)
{
    Console.Clear();
    for (int i = 0; i < ArrWorker.Length; i++)
    {
        Console.WriteLine("{0} {1} {2}\n\tДолжность: {3}\n\tЗарплата:
{4}\n\tГод: {5}", ArrWorker[i].surname, ArrWorker[i].name[0],
ArrWorker[i].patronymic[0], ArrWorker[i].jobrang,
ArrWorker[i].salary, ArrWorker[i].year);
    }
}

public static void PrintWorker(Worker ArrWorker)
{
    Console.WriteLine("{0} {1} {2}\n\tДолжность: {3}\n\tЗарплата:
{4}\n\tГод: {5}", ArrWorker.surname, ArrWorker.name[0],
ArrWorker.patronymic[0], ArrWorker.jobrang, ArrWorker.salary,
ArrWorker.year);
}

public static int CorrectInt(string b)
{
    int a;
    do
    {
        if (int.TryParse(b, out a))
        {
            break;
        }
        Console.WriteLine("Число должно быть целым и положительным");
        b = Console.ReadLine();
    } while (true);
    if (a < 0)
    {
        a = a * -1;
    }
    return a;
}

public static int Main()
{
    Console.OutputEncoding = Encoding.UTF8;
    Worker[] ArrWorker = new Worker[10];
    ArrWorker[0] = new Worker("Иванов", "Иван", "Иванович", "Продавец",
"20000", "2018");
    ArrWorker[1] = new Worker("Бобков", "Влад", "Сишарпович", "Программист",
"120000", "2016");
    ArrWorker[2] = new Worker("Хрустов", "Дима", "Олегович", "Маркетолог",
"80000", "2020");
    ArrWorker[3] = new Worker("Прикольный", "Прикол", "Приколыч", "Комик",
"210500", "2015");
    ArrWorker[4] = new Worker("Теслов", "Сергей", "СанФранцискович",
"ДезигнерФромСанФрансисько", "150000", "2016");
}

```

```

        ArrWorker[5] = new Worker("Хрусталёв", "Дима", "Олегович", "Маркетолог",
"85000", "2020");
        ArrWorker[6] = new Worker("Закшевский", "Родион", "Викторович", "Прогер-
бомж за 3 копейки", "0", "2022");
        ArrWorker[7] = new Worker("Кабанов", "Алексей", "Владимирович",
"Программист", "130000", "2014");
        ArrWorker[8] = new Worker("Кабанов", "Алексей", "Владимирович",
"Дизайнер", "50000", "2021");
        ArrWorker[9] = new Worker("Мудаков", "Мудила", "Идиотович", "Манагер",
"100000", "2020");
        PrintArrWorker(ArrWorker);
        Console.WriteLine("1. Сортировка по ЗП");
        Console.WriteLine("2. Сортировка по стажу");
        Console.WriteLine("3. Сортировка по должности");
        Console.WriteLine("0. Выход");
        string checkclient = Console.ReadLine();
        int client = CorrectInt(checkclient);
        do
        {
            while (true)
            {
                if ((client == 1) || (client == 2) || (client == 3) || (client
== 0))
                {
                    break;
                }
                Console.WriteLine("Некорректный ввод, повторите его");
                checkclient = Console.ReadLine();
                client = CorrectInt(checkclient);
            }
            if (client == 1)
            {
                Console.WriteLine("Введите ЗП: ");
                string checkSalary = Console.ReadLine();
                int salary = CorrectInt(checkSalary);
                for (int i = 0; i < ArrWorker.Length; i++)
                {
                    if (ArrWorker[i].salary >= salary)
                    {
                        PrintWorker(ArrWorker[i]);
                    }
                }
            }
            if (client == 2)
            {
                Console.WriteLine("Введите стаж: ");
                string checkYear = Console.ReadLine();
                int year = CorrectInt(checkYear);
                for (int i = 0; i < ArrWorker.Length; i++)
                {
                    if (2022 - ArrWorker[i].year >= year)
                    {
                        PrintWorker(ArrWorker[i]);
                    }
                }
            }
            if (client == 3)
            {
                Console.WriteLine("Введите должность: ");
                string jobrang = Console.ReadLine();
                for (int i = 0; i < ArrWorker.Length; i++)
                {
                    if (ArrWorker[i].jobrang == jobrang)
                    {
                        PrintWorker(ArrWorker[i]);
                    }
                }
            }
        }
    }
}

```

```

    }
}
Console.WriteLine("1. Сортировка по ЗП");
Console.WriteLine("2. Сортировка по стажу");
Console.WriteLine("3. Сортировка по должности");
Console.WriteLine("0. Выход");
checkclient = Console.ReadLine();
client = CorrectInt(checkclient);
} while (client != 0);
return 0;
}
}
}

```

```

D:\UEBA\Курс 2\ООП\Лабораторная №3\Lab3_Zakshevskij_UTS-22\bin\Debug\net6.0\Lab3_Zakshevskij_UTS-22.exe
Иванов И. И.
    Должность: Продавец
    Зарплата: 20000
    Год: 2018
Бобков В. С.
    Должность: Программист
    Зарплата: 120000
    Год: 2016
Хрустов Д. О.
    Должность: Маркетолог
    Зарплата: 80000
    Год: 2020
Прикольный П. П.
    Должность: Комик
    Зарплата: 210500
    Год: 2015
Теслов С. С.
    Должность: ДизайнерФромСанФрансисско
    Зарплата: 150000
    Год: 2016
Хрусталёв Д. О.
    Должность: Маркетолог
    Зарплата: 85000
    Год: 2020
Закшевский Р. В.
    Должность: Прогер-бомж за 3 копейки
    Зарплата: 0
    Год: 2022
Кабанов А. В.
    Должность: Программист
    Зарплата: 130000
    Год: 2014
Кабанов А. В.
    Должность: Дизайнер
    Зарплата: 50000
    Год: 2021
Фудаков М. И.
    Должность: Менеджер
    Зарплата: 100000
    Год: 2020
1. Сортировка по ЗП
2. Сортировка по стажу
3. Сортировка по должности

```

Рисунок 1. Пример работы программы



```
D:\UEBA\Курс 2\ООП\Лабораторная №3\Lab3_Zakshevskij_UTS-22\bin\Debug\net6.0\Lab3_Zakshevskij_UTS-22.exe
Должность: Менеджер
Зарплата: 100000
Год: 2020
1. Сортировка по ЗП
2. Сортировка по стажу
3. Сортировка по должности
0. Выход
1
Введите ЗП:
80000
Бобков В С
Должность: Программист
Зарплата: 120000
Год: 2016
Хрустов Д О
Должность: Маркетолог
Зарплата: 80000
Год: 2020
Трикольный П П
Должность: Комик
Зарплата: 210500
Год: 2015
Теслов С С
Должность: ДизайнерФромСанФрансисско
Зарплата: 150000
Год: 2016
Хрусталёв Д О
Должность: Маркетолог
Зарплата: 85000
Год: 2020
Хабанов А В
Должность: Программист
Зарплата: 130000
Год: 2014
Иудаков И И
Должность: Менеджер
Зарплата: 100000
Год: 2020
1. Сортировка по ЗП
2. Сортировка по стажу
3. Сортировка по должности
0. Выход
```

Рисунок 2. Пример взаимодействия с программой

**Вывод:** в данной работе научились создавать и работать с классами, объектами и методами