# Объектно-ориентированное программирование

# Лабораторная работа №3

В данной лабораторной работе необходимо использовать полученные знания по созданию собственных классов, использованию конструкторов, методов и исключений. Разработать классы для описанных ниже объектов. Включить в класс методы set (…), get (…), info (…). Определить другие методы.

## Задание:

### Варианты:

#### Вариант 1

Student: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс. Создать массив объектов. Вывести:

* список студентов заданного факультета;
* списки студентов для каждого факультета и курса;
* список студентов, родившихся после заданного года.

#### Вариант 2

Abiturient: Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив объектов. Вывести:

* список абитуриентов, имеющих неудовлетворительные оценки;
* список абитуриентов, сумма баллов у которых не меньше заданной;
* выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.

#### Вариант 3

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.

#### Вариант 4

Book: Автор, Название, Издательство, Год, Количество страниц.

Создать массив объектов. Вывести:

* список книг заданного автора;
* список книг, выпущенных заданным издательством;
* список книг, выпущенных после заданного года.

#### Вариант 5

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.

#### Вариант 6

Train: Пункт назначения, Номер поезда, Время отправления,

Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести:

* список поездов, следующих до заданного пункта назначения;
* список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
* список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

#### Вариант 7

Product: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести:

* список товаров для заданного наименования;
* список товаров для заданного наименования, цена которых не превышает указанной;
* список товаров, срок хранения которых больше заданного.

#### Вариант 8

Patient: Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив объектов. Вывести:

* список пациентов, имеющих данный диагноз;
* список пациентов, номер медицинской карты которых находится в заданном интервале.

#### Вариант 9

Bus: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив объектов. Вывести:

* список автобусов для заданного номера маршрута;
* список автобусов, которые эксплуатируются больше 10 лет;
* список автобусов, пробег у которых больше 10 000 км.

#### Вариант 10

Customer: Фамилия, Имя, Отчество, Адрес, Телефон, Номер кредитной карточки, Номер банковского счета. Создать массив объектов. Вывести:

* список покупателей в алфавитном порядке;
* список покупателей, номер кредитной карточки которых находится в заданном интервале.

#### Вариант 11

File: Имя файла, Размер, Дата создания, Количество обращений. Создать массив объектов. Вывести:

* список файлов, упорядоченный в алфавитном порядке;
* список файлов, размер которых превышает заданный;
* список файлов, число обращений к которым превышает заданное.

#### Вариант 12

Word: Слово, Номера страниц, на которых слово встречается (от 1 до 10), Число страниц. Создать массив объектов. Вывести:

* слова, которые встречаются более чем на N страницах;
* слова в алфавитном порядке;
* для заданного слова номера страниц, на которых оно встречается.

#### Вариант 13

House: Адрес, Этаж, Количество комнат, Площадь. Создать массив объектов. Вывести:

* список квартир, имеющих заданное число комнат;
* список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в определенном промежутке;
* список квартир, имеющих площадь, превосходящую заданную.

#### Вариант 14

Phone: Фамилия, Имя, Отчество, Адрес, Номер, Время внутригородских разговоров, Время междугородних разговоров. Создать массив объектов. Вывести:

* сведения об абонентах, время внутригородских разговоров которых превышает заданное;
* сведения об абонентах, воспользовавшихся междугородней связью;
* сведения об абонентах, выведенные в алфавитном порядке.

## Ход решения

## Задание:

Person: Фамилия, Имя, Отчество, Адрес, Пол, Образование, Год рождения. Создать массив объектов. Вывести:

* список граждан, возраст которых превышает заданный;
* список граждан с высшим образованием;
* список граждан мужского пола.

### 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;  
}

### Создаем конструктор класса

Для того чтобы класс был проинизиализирован 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.

### 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 необходимо проверять валидность входных параметров, и выбрасывать исключения в случае ошибки.

### 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 назывемые 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.

### Вывод полной информации о гражданине

Для того чтобы можно быть 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);  
}

### Основная программа

В основной программе используем все созданные методы и знания полученные на лекциях и домашних работах. Создаем массив значений, заполняем его, и производим поиск по заданным критериям.