



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

*Факультет Информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки
09.03.02 «Информационные системы и технологии»

ЛАБОРАТОРНАЯ РАБОТА № 1

Дисциплина: «Объектно-ориентированное программирование»

Тема: «Создание программы с использованием классов»

Выполнил: студент группы 241-334

Зотов Михаил Павлович

(Фамилия И.О.)

Дата, подпись 25.09.2025

(Дата)

(Подпись)

Проверил:

(Фамилия И.О., степень, звание)

(Оценка)

Дата, подпись

(Дата)

(Подпись)

Замечания:

Москва

2025

Цель: получить практические навыки в создании программ, содержащих определения классов и использующих объекты на языке программирования C#.

Задание. Создание классов, реализация методов, объявление классов, запуск программы и вывод полученного на экран.

Дан следующий класс:

```
class Person
{
    public string name = "Ben";
    public int age = 18;
    public string email = "ben@gmail.com";
    public Person(string name)
    {
        this.name = name;
    }
    public Person(string name, int age) : this(name)
    {
        this.age = age;
    }
    public Person(string name, int age, string email) : this("Bob", age)
    {
        this.email = email;
    }
}
```

Какие значения будут иметь поля name, age и email после выполнения следующего кода и почему? В каком порядке будут вызываться конструкторы класса Person?

Листинг:

MainWindow

— □ ×

Создать Человечка

Информация о Человечке:

Name: Bob

Age: 18

Email: ben@gmail.com

Анализ вызова конструкторов:

При создании объекта: `Person person = new Person("Ben", 18, "ben@gmail.com");`

Порядок вызова конструкторов:

Вызывается конструктор с тремя параметрами: `Person("Ben", 18, "ben@gmail.com")`

Конструктор с тремя параметрами вызывает конструктор с двумя параметрами: `this("Bob", age) → Person("Bob", 18)`

Конструктор с двумя параметрами вызывает конструктор с одним параметром: `this(name) → Person("Bob")`

Выполняется конструктор с одним параметром: `this.name = "Bob"`

Возврат в конструктор с двумя параметрами: `this.age = 18`

Возврат в конструктор с тремя параметрами: `this.email = "ben@gmail.com"`

Инициализация полей

Начальные значения полей:

`name = "Ben"`

`age = 18`

`email = "ben@gmail.com"`

Процесс изменения значений:

Поля инициализируются значениями по умолчанию

Конструктор `Person("Bob")` устанавливает `name = "Bob"`

Конструктор `Person("Bob", 18)` устанавливает `age = 18`

Конструктор `Person("Ben", 18, "ben@gmail.com")` устанавливает `email = "ben@gmail.com"`

Итоговые значения полей

name: "Bob" (перезаписан в цепочке конструкторов)

age: 18 (переданное значение 18)

email: "ben@gmail.com" (переданное значение)

Ответ

Значения полей после выполнения кода:

`name = "Bob"`

`age = 18`

`email = "ben@gmail.com"`

Порядок вызова конструкторов:

`Person(string name)` - с параметром "Bob"

`Person(string name, int age)` - с параметрами "Bob", 18

`Person(string name, int age, string email)` - с параметрами "Ben", 18, "Ben@gmail.com"

Ключевой момент: несмотря на то, что в последний конструктор передано имя "Ben", цепочка вызовов `this("Bob", age)` переопределяет имя на "Bob".