

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Учреждение образования «Полоцкий государственный университет»

Факультет информационных технологий  
Кафедра технологий программирования

**Лабораторная работа №1**  
по дисциплине: **«Объектно-ориентированные технологии программирования и  
стандарты проектирования»**  
на тему: **«Конструкторы и деструкторы»**

ВЫПОЛНИЛ

студент группы 16 ИТ-3  
Яблонский А.С.

ПРОВЕРИЛ

преподаватель  
Ярошевич П.В.

Полоцк, 2018 г.

# Вариант №11

## Постановка задачи:

По условию задания варианта №11, было необходимо

1. Реализовать класс «Фирма» с не менее чем шестью полями.
2. Реализовать конструкторы класса
  - a. без параметров
  - b. с параметрами
  - c. конструктор копирования
3. Реализовать методы ввода полей класса
4. Реализовать методы вывода полей класса

## Реализация:

1. **Поля.** В результате работы был описан класс `Factory` со следующими полями:
  - `Integer id` - ID фирмы
  - `String name` - название фирмы
  - `String address` - физический адрес
  - `String tel` - номер телефона
  - `String owner` - имя владельца
  - `Long createDate` - дата основания в миллисекундах

2. **Ввод вывод полей.** Для ввода полей класса использовался объект класса `Scanner` и его методы `nextLine()`, `nextInt()`, `nextLong()`.

Пример метода ввода поля ID:

```
fun readId() {
    val scanner = Scanner(System.`in`)
    print("Enter company ID: ")
    this.id = scanner.nextInt()
}
```

Для вывода полей использовался методы `print()` и `println()`. Пример вывода поля `address`:

```
fun printAddress() {
    println("Company address: ${this.address}")
}
```

3. **Конструкторы.**

- 1) В конструктор класса с параметрами, в качестве параметров передавались значения ID и название фирмы. В результате конструктор принял следующий вид:

```
constructor(id: Int, name: String) {
    this.id = id
    this.name = name
}
```

В теле конструктора, переданные ему значения присваиваются соответствующим полям объекта.

- 2) Конструктор копирования был реализован следующим образом

```
constructor(company: Company) {  
    this.id = company.id  
    this.name = company.name  
    this.address = company.address  
    this.tel = company.tel  
    this.owner = company.owner  
    this.createDate = company.createDate  
}
```

Данный конструктор принимает единственный аргумент - объект класса Company. После устанавливает все поля текущего объекта (this) соответственным поля объекта, переданного в аргументах

- 3) Конструктор без параметров:

```
constructor()
```

## Заключение:

Данная лабораторная работа на примере объяснила концепцию объектов в принципе ООП. Мне пришло понимание необходимости конструкторов, полей и методов для каждого конкретного экземпляра класса - объекта.