

Лабораторная работа. Интерфейсы. Анонимные классы.

Анонимные классы

Механизм анонимных классов позволяет объявить класс и сразу создать его экземпляр. Это позволяет сделать код кратким и выразительным. Анонимные классы удобно использовать, если класс нужен единожды.

Основная особенность - анонимный класс не имеет имени. Анонимный класс может быть подклассом существующего класса или реализацией интерфейса.

Особенности анонимного класса:

- нет явного конструктора;
- к анонимному классу невозможно обратиться извне объявляющего его выражения;
- анонимные классы не могут быть статическими;
- анонимный класс всегда конечен (`final`);
- каждое объявление анонимного класса уникально.

Пример.

Создать интерфейс «Умение готовить» с методом `cook`.

Создать класс «Повар», наследующий интерфейс, реализующий метод `cook` (вывод сообщения о готовке).

Создать класс «Банкет», с методом `force`, параметром которого является объект интерфейса, вызывающий метод `cook`.

Реализовать следующие варианты:

- 1) В классе `Main` создать объект класса «Повар», объект класса «Банкет» и вызвать метод `force` с параметром (объект класса «Повар»).
- 2) Выполнить те же действия, но с использованием анонимного класса (без класса «Повар»)

1 Вариант

```
public interface Cookable { //Интерфейс: Умение готовить
    void cook();
}
public class Cooker implements Cookable { // класс Повар
    @Override
    public void cook() { // метод Готовить
        System.out.println("Готовлю...");
    }
}
public class Banket { //класс Банкет
    public void force (Cookable cookable) {
        cookable.cook();
    }
}
public class Main{
    public static void main(String[] args) {
        Cooker cooker = new Cooker(); // создание Повара
        Banket banket = new Banket(); // создание Банкета
        banket.force(cooker); // вывод Готовлю... Вызов метода с параметром (Повар)
    }
}
```

2 вариант

А если нам повар нужен один раз, то можно класс не создавать. Тогда удаляем повара и используем анонимный класс (без имени), реализующий метод `cook`.

```
public class Main{
    public static void main(String[] args) {
        Banket banket = new Banket(); // создание Банкета
        banket.force(new Cookable() { // создание анонимного класса в параметре метода force
            @Override
            public void cook() { // реализация метода Готовить
                System.out.println("Готовлю...");
            }
        });
    }
}
```

Задание.

1. Создать интерфейс Профком, в котором определить один метод для возврата отчета в виде строки с одним параметром (количество месяцев, за которые нужно сделать отчет - int).

2. Создать класс Директор с одним методом request, который в качестве параметра принимает объект типа Профком и переменную int. В этом методе мы получаем отчет Профкома с выводом его на экран в виде строки «Отчет профкома за i месяцев».

3. Реализовать выполнение метода request двумя способами:

а) Создание объекта реализующего метода и передача его в качестве параметра.

б) Использование анонимного класса, метод request принимает 2 аргумента: первый - Профком, второй - число месяцев.