



Лекция 5

Особенности объектной модели Java





Содержание





Статические элементы

- Статические элементы
 - Относятся к классу в целом, а не к экземплярам класса (объектам)
 - Существуют в единственном экземпляре независимо от числа созданных объектов
 - Обращение – через имя класса, а не через ссылку на объект





Инициализация и вызов

```
class StudentGroup {  
    public static int totalCount;  
  
    public static int getCount() {  
        return totalCount;  
    }  
  
    public static void sessionFailed() {  
        totalCount--;  
    }  
}
```





Инициализация и вызов

```
StudentGroup.totalCount = 24;
```

```
StudentGroup group4350 = new StudentGroup();  
int i = group4350.getCount();  
System.out.println(i); // ????
```

```
group4350 = null;  
group4350.sessionFailed(); // ????  
System.out.println(group4350.totalCount);  
    // ????  
System.out.println(StudentGroup.totalCount);  
    // ????
```





Статический и динамический контекст

- Статический контекст
 - Статические методы и инициализаторы
 - Инициализаторы статических полей
- Динамический контекст
 - Все прочие части кода

В динамическом контексте работа всегда идет с экземпляром класса; в статическом – с самим классом





Особенности

- Обращение к полям и методам
 - По простому имени – в порядке следования; по составному имени – в любом порядке
 - Обращение из статического контекста в динамический – запрещено

```
class StudentGroup {  
    static int a = b; // ????  
    static int c = StudentGroup.b;  
    static int b = 7;  
  
    public void process() {}  
    public static void main (String[] s) {  
        process(); // ????  
    }  
}
```





Ключевые слова `this` и `super`

- Ключевое слово `this` возвращает ссылку на **объект**, в методе которого оно использовано
- Использования `this`:
 - Передача данного объекта по ссылке
 - Разрешение «затемняющих» объявлений
 - Обращение к полям, объявляемых ниже
 - Вызов других конструкторов класса





Ключевые слова `this` и `super`

- Ключевое слово `super` позволяет обратиться к методам родительского объекта:
 - Вызов конструктора родительского класса
 - Вызов переопределенного метода родительского класса





Абстрактные классы и методы

- Абстрактный метод
 - не имеет реализации
 - обязательно должен быть реализован в наследниках

При наличии абстрактных методов класс обязательно объявляется как абстрактный

- Абстрактный класс
 - может иметь абстрактные методы
 - не может быть инстанцирован





Абстрактный класс

- Заголовок класса

[модификатор доступа]

```
abstract final class <имя класса>  
    { [тело класса] }
```

- Объявление метода

```
abstract
```

```
private | native | static
```

```
<тип> <имя>([параметры]);
```





Интерфейс

- Класс, содержащий только абстрактные методы = интерфейс
- Не может быть инстанцирован

[модификатор доступа]

interface <имя интерфейса>

{[тело интерфейса]}

Тело интерфейса состоит только из абстрактных методов





Интерфейс

- Способ реализации множественного наследования
- Мощный способ организации полиморфных программ

Тип переменной:

- «Абстрактный класс»
 - Неабстрактный наследник
- «Интерфейс» – классы, реализующие его:
 - напрямую (implements)
 - наследуемые от реализующих классов
 - реализующие наследников интерфейса





Структура объявления класса

- Заголовок класса

[модификатор доступа]

[**final**]

class <имя класса>

[**extends** <имя класса>]

[**implements** <имя интерфейса>[, ...]]

{ [тело класса] }





Полиморфизм

- Ссылочный тип обладает элементами:
 - Непосредственно объявленными в его теле
 - Объявленными в родительском классе или интерфейсах, кроме:
 - private-элементов
 - «скрытых» одноименными элементами
 - переопределенных методов
- Модификаторы доступа можно только «расширять» (сужать запрещено)





Массив

- Массив – набор однотипных значений
 - Отдельный элемент массива не имеет имени
- Объявление массива

```
int a[];
```

```
int[] a;
```

```
int[] a[];
```

```
int a[][];
```





Размер массива

- Задается при создании

```
int a[] = new int[5];
```
- Доступна в специальном свойстве

```
int length = a.length;
```
- Индексация начинается с «0»
- Ошибка индексации – ошибка времени исполнения программы





Содержание

