

5. OOP. JAVA

Overloading, overriding, abstract classes, interfaces

Перегрузка методов

Overloading

```
class A {  
    void m() {...}  
    void m(int x) {...}  
}
```

Конструкторы класса всегда перегружены.

Перекрытие методов

Overriding

Позволяет реализовать полиморфизм

```
class A {  
    void m() {...}  
}  
class B extends A {  
    void m() {...}  
}
```

Значения полей по умолчанию

- примитивные типы числовые ==> 0
- boolean ==> false
- ссылочные ==> null

Инициализация полей

- При объявлении
- В конструкторе
- В блоках инициализации
- В методах

Ключевое слово `final`

Четыре контекста:

- `класс` - нельзя наследовать
- `метод` - нельзя перекрыть
- `поле` - константа
- `локальная переменная` - константа

Абстрактные классы

<http://docs.oracle.com/javase/tutorial/java/landl/abstract.html>

Класс объявленный со спецификатором
abstract

Может содержать абстрактные методы (методы без реализации).

```
abstract class A {  
    abstract void m();  
}
```

Интерфейсы

<http://docs.oracle.com/javase/tutorial/java/landl/createinterface.html>

- Определяют границы взаимодействия между объектами
- Определяют абстракцию, реализацию которой предоставляет имплементирующая интерфейс сторона

Множественное наследование интерфейсов

```
interface Interf extends Interf1, Interf2 {  
    ...  
}
```

Реализация интерфейса

```
class A implements Interf1, Interf2 {  
    ...  
}
```

Реализация интерфейсов и расширение класса

```
class A extends B implements Interf1, Interf2 {  
    ...  
}
```