



DEV-J110. Java SE. Введение в язык Java

Наследование классов



Рассматриваемые вопросы

- Наследование.
- Наследование членов класса.
- Переопределение методов при наследовании.
- Порядок вызова конструкторов при наследовании.
- Спецификатор `final`.



Наследование

- **Наследование** - это механизм, позволяющий определять новый пользовательский тип на основе другого пользовательского типа. Наследование – один из основных принципов объектно-ориентированной парадигмы программирования.



Основные понятия

- Суперкласс (superclass).
- Подкласс (subclass).
- Одночное наследование.
- Влияние спецификаторов **public**, **protected** и **private** на наследование членов класса.



Синтаксис наследования

- Определение суперкласса:

```
1. class A { ... }
```

- Определение подкласса:

```
1. class B extends A {  
2.     ...  
3. }
```



Наследование членов класса

▣ Наследуются:

1. поля класса;
2. методы класса.

▣ Не наследуются:

1. статические поля и методы класса;
2. конструкторы класса.



Переопределение методов

- Переопределение (**overriding**) является формой динамического полиморфизма.
- При переопределении метода суперкласса метод подкласса должен:
 - воспроизводить **точную сигнатуру** переопределяемого метода суперкласса;
 - иметь область видимости такую же или более широкую, чем область видимости переопределяемого метода;
 - выбрасывать исключение более «узкого типа», чем исключение, которое выбрасывает переопределяемый метод, или вообще не выбрасывать такое исключение .



Порядок вызова конструкторов

□ При создании объекта (экземпляра) подкласса первым вызывается конструктор суперкласса.

□ Пример:

```
1. class A {
```

```
2.     public A () { System.out.println ("A:A()"); } ...  
   }
```

```
3. class B extends A {
```

```
4.     public B () { System.out.println ("B:B()"); } ...  
   }
```




Запрет на наследование

- Спецификатор `final`.
- Определение константы:
 1. `final A a = new A();`
- Запрет на наследование класса:
 2. `public final class A { ... }`
- Запрет на переопределение метода.
 3. `public class B {`
 4. `public final void a () { ... }`
 5. `}`



Заключение

- Обзор рассмотренных вопросов.
- Вопросы для самостоятельного изучения.