



МОДИФИКАТОРЫ

МОДИФИКАТОРЫ

```
final  
sealed  
private  
protected
```

- С наследованием и доступом связано несколько особенностей, управляемых ключевыми словами — **модификаторами**

FINAL

```
final class Human
```

```
class SuperHuman extends Human // ошибка!!
```

- Классы могут быть помечены **final**. Такие классы нельзя наследовать

FINAL

```
class Human{  
    final def powerLevel: Int = 100  
}  
  
class SuperHuman extends Human{  
    override def powerLevel: Int = 9001 // ошибка!  
}
```

- Члены классов тоже могут быть помечены как **final**. Такие члены нельзя переопределять

SEALED

```
sealed trait Weekday
```

```
case object Monday extends Weekday
```

```
case object Tuesday extends Weekday
```

```
case object Wednesday extends Weekday
```

```
case object Thursday extends Weekday
```

```
case object Friday extends Weekday
```

```
case object Saturday extends Weekday
```

```
case object Sunday extends Weekday
```

- Классы и trait могут быть помечены как **sealed**. Это значит, что унаследовать их можно только в том же файле исходного кода
- Вы заранее знаете все альтернативы данного типа. Очень полезно для pattern matching

PRIVATE

```
class Dog {  
    private val kind: String = "dog"  
}  
  
object Dog{  
    def kindOf(dog: Dog): String = dog.kind  
}  
  
(new Dog).kind // ошибка!!!
```

- Члены классы могут быть помечены **private**. Это значит доступ к ним возможен только из экземпляров того же класса или компаньона

PRIVATE [THIS]

```
class Dog {  
  private [this] val kind: String = "dog"  
  def kindOf(dog: Dog): String = dog.kind // Ошибка!!!  
}
```

- Пометка **private[this]** означает, что член класса может быть доступен только из того же самого экземпляра

PRIVATE [...]

```
package tinkoff.course.classes

abstract class Dog {
    private [Dog] val age: Int
    private [classes] val kind: String
    private [course] val name: String
}
```

- **Private[...]** означает, что член класса виден только из какого-то пакета или класса (полезно для вложенных классов)

PROTECTED

```
trait Animal {  
  protected val kind: String  
}  
  
class Dog extends Animal {  
  protected val kind = "dog"  
  def kindOf(dog: Dog): String = dog.kind  
}  
  
(new Dog).kind // ошибка!!!
```

- Член класса, помеченный как **protected**, виден в том числе и всем потомкам этого класса

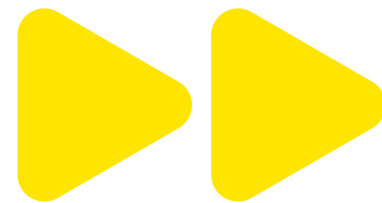
ПАРАМЕТРЫ КОНСТРУКТОРА

```
trait Animal {  
  protected val kind: String = "animal"  
}
```

```
class Mammal  
  (override protected val kind: String)  
  extends Animal {  
  def printKind = println(kind)  
}
```

```
(new Mammal("dog")).printKind  
(new Mammal("dog")).kind // ошибка !!!
```

**В этом разделе
мы изучили модификаторы**



**В следующем изучим
обобщённые типы**