

Знакомство со Scala

Пятое занятие

ООП на Scala

- Любое значение – объект
- Любая операция – метод
- $1 + 2$ – то же самое, что $(1).+(2)$
- Классы не содержат статики
- Объекты-синглтоны, компаньоны
- Traits – интерфейсы с реализацией

ООП на Scala

```
class BaseDAO[T] { // #1
  def get(id: Long): T
  def save(t: T): T
}

trait Logging { // #2
  protected val log = LoggerFactory.getLogger(getClass)
}

object UserDAO extends BaseDAO[User] with Logging { // #3
  override def get(id: Long): User = { // #4
    log.info("...")
    // ...
  }
  override def save(u: User): User = { /* ... */ }
}
```

Конструкторы классов

```
class BasePerson(val name: String) // #1
class Person(val name: String, val age: Int) extends
    BasePerson(name) { // #2
    def this(name: String) = {
        this(name, 18) // #3, обязательно начинается с этого
    }

    {
        var localConstructorVar = 1 // #3
    }

    val ageTitle = { // #4
        // ...
    }
}
```

Инициализация trait-ов

```
trait A {  
    println("A")  
    val a = 1 // #1  
    lazy val b = 2 // #2  
}  
  
trait B {  
    println("B")  
}  
  
trait C extends B with A { // #3  
    println("C")  
}  
  
trait D extends A with B with C { // #4  
    println("D")  
}
```

“Self type” trait-ов

```
abstract class Button {  
    def draw: Unit  
    def push: Unit  
}
```

```
trait ButtonDecorator {  
    this: Button => // #1  
    // ...  
}
```

```
class MyButton extends Button with ButtonDecorator // #2
```

Задание №5-1

1. Объявить иерархию из 3-х классов: Shape, Circle и Square. Объявить trait Color с полем color:String. Объявить классы ColoredSquare и ColoredCircle. Переопределить toString() для вывода состояния объектов. Написать тестовый код.
2. Выводить состояние ColoredSquare и ColoredCircle без явного переопределения toString() в них (использовать вызов super())