



АБСТРАКТНЫЕ ТИПЫ

TRAIT

```
trait Animal{  
  def name: String  
  val greeting: String = s"Hi, I'm $name"  
}
```

- **Трейты** — это абстрактные типы, некоторые члены могут быть не определены
- В отличие от класса, у Trait не может быть параметров на момент scala 2.12
- В следующих версиях scala это может измениться

АБСТРАКТНЫЕ КЛАССЫ

```
abstract class Animal(name: String){  
    val greeting: String = s"Hi, I'm $name"  
}
```

- **Абстрактные классы** — это классы, у которых некоторые члены могут быть не определены
- У них также могут быть определены параметры

АБСТРАКТНЫЕ КЛАССЫ

```
abstract class Animal(name: String)

trait Mammal
```

- Тело класса или трейта также можно оставить пустым

СОЗДАНИЕ ЭКЗЕМПЛЯРОВ

```
trait Animal{  
  def name: String  
  val greeting: String = s"Hi, I'm $name"  
}  
  
val animal = new Animal{  
  def name = "Bobik"  
}
```

- Создать экземпляр абстрактного типа напрямую нельзя
- Но можно создать экземпляр "анонимного класса", доопределяющего абстрактный

ИНИЦИАЛИЗАЦИЯ

```
trait Animal{  
  def name: String  
  val greeting: String = s"Hi, I'm $name"  
  
  println("$name is created")  
}  
  
val animal = new Animal{  
  def name = "Bobik"  
}
```

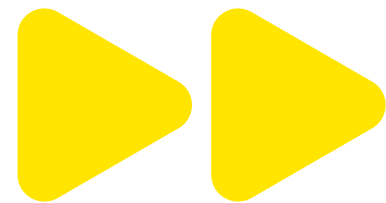
- Абстрактные типы также могут содержать блоки инициализации

СОЗДАНИЕ ЭКЗЕМПЛЯРОВ

```
abstract class Animal(name: String){  
    val greeting: String = s"Hi, I'm $name"  
}  
  
val animal = new Animal("Bobik"){ }
```

- Если мы создаём экземпляр абстрактного класса, мы должны передать все параметры
- Обратите внимание, что даже если все абстрактные члены определены, нам всё равно нужно поставить фигурные скобки

**В этом разделе
мы изучили абстрактные типы**



**В следующем
практика**