



ВАРИАНТНОСТЬ И РОД

КОВАРИАНТНОСТЬ

```
trait Coll[+A]{  
  def apply(i: Int): A  
}
```

КОВАРИАНТНОСТЬ

```
trait Coll[+A]{  
  def apply(i: Int): A  
}
```

$A \leqslant B$



$\text{Coll}[A] \leqslant \text{Coll}[B]$

КОВАРИАНТНОСТЬ

```
trait Coll[+A]{  
  def apply(i: Int): A  
  def headOption: Option[A]  
  def tail: Coll[A]  
}
```

КОНТРВАРИАНТНОСТЬ

```
trait Printer[-A]{  
  def print(a: A): String  
}
```

КОНТРВАРИАНТНОСТЬ

```
trait Printer[-A]{  
  def print(a: A): String  
}
```

$A \leqslant B$



$\text{Printer}[B] \leqslant \text{Printer}[A]$

КОНТРВАРИАНТНОСТЬ

```
trait Printer[-A]{  
  def print(a: A): String  
  def printList(as: List[A]): String  
  def prefixed(s: String): Printer[A]  
}
```

ВАРИАНТНОСТЬ

$A \Rightarrow B$

```
trait Function1[-A, +R]
```

- Функции контрвариантны по первому параметру и ковариантны по второму

ВАРИАНТНОСТЬ

```
trait Coll[+A]{  
  def apply(i: Int): A  
  def map[B](f: A => B): Coll[B]  
}
```

- Функции контрвариантны по первому параметру и ковариантны по второму

ВАРИАНТНОСТЬ

```
trait Printer[-A]{  
  def print(a: A): String  
  def contramap[B](f: B => A): Printer[B]  
}
```

- Функции контрвариантны по первому параметру и ковариантны по второму

РОД

```
Int, String, List[Int]: T  
Map[Int, String], Int => String : T
```

- Род — это тип типов или форма типов

РОД

```
Int, String, List[Int]: T  
Map[Int, String], Int => String : T
```

```
List, Vector : T[_]
```

- Род — это тип типов или форма типов

РОД

```
Int, String, List[Int]: T  
Map[Int, String], Int => String : T
```

```
List, Vector : T[_]
```

```
Map, Function : T[_ , _]
```

- **Род** — это тип типов или форма типов

РОД

```
Int, String, List[Int]: T  
Map[Int, String], Int => String : T
```

```
List, Vector : T[_]
```

```
Map, Function : T[_ , _]
```

```
StateT: T[_[_], _, _]  
Hoist: T[_[_[_], _]]
```

- **Род** — это тип типов или форма типов

РОД

```
case class IntContainer[F[_]](value: F[Int])
```

- Параметры типа могут иметь высший род

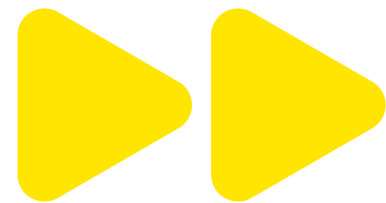
РОД

```
case class IntContainer[F[_]](value: F[Int])
```

```
case class Dict[K, V, T[X] <: Seq[X]](items: T[(K, V)])
```

- Параметры типа могут иметь высший род

**В этом разделе мы изучили
вариантность и род**



**В следующем
практика**