

Enums

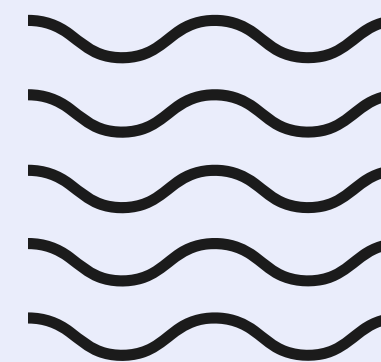


ПОДГОТОВИЛ: АЛИШЕР ХАМИДОВ

План лекции

Основные моменты:

1. Что такое enum?
2. Объявление
3. Метод `.values`
4. Конструкторы, поля, методы
перечислений
5. Общие замечания





Перечисления(enum) представляют набор логически связанных **констант**.





Объявление перечисления происходит с помощью оператора **enum**, после которого идет название перечисления. Затем идет список элементов перечисления через запятую:

```
enum Day{  
    MONDAY,  
    TUESDAY,  
    WEDNESDAY,  
    THURSDAY,  
    FRIDAY,  
    SATURDAY,  
    SUNDAY  
}
```



Метод .values()

```
public static void main(String[] args) {  
    enum Size{  
        SMALL,  
        MIDDLE,  
        BIG  
    }  
    System.out.println(Arrays.toString(Size.values()));  
}
```

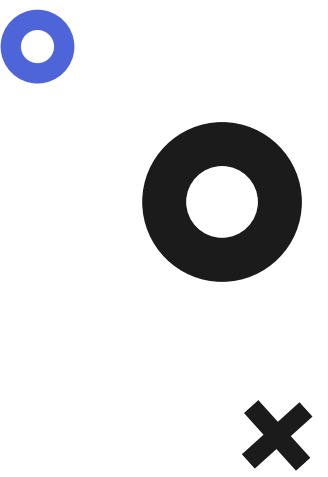
Метод .values()

```
public static void main(String[] args) {  
    enum Size{  
        SMALL,  
        MIDDLE,  
        BIG  
    }  
    System.out.println(Arrays.toString(Size.values()));  
    // [SMALL, MIDDLE, BIG]  
}
```

Метод `.values()` и цикл `for-each`

```
public static void main(String[] args) {  
    enum Size{  
        SMALL,  
        MIDDLE,  
        BIG  
    }  
    System.out.println(Size.MIDDLE.ordinal()); // 1  
}
```

Конструкторы, поля и методы перечисления



```
System.out.println(Color.RED.getCode());    // #FF0000
```

```
System.out.println(Color.GREEN.getCode());  // #00FF00
```

```
enum Color {  
    RED("#FF0000"), BLUE("#0000FF"), GREEN("#00FF00");  
    private String code;  
  
    Color(String code) {  
        this.code = code;  
    }  
    public String getCode() {  
        return code;  
    }  
}
```


Конструкторы, поля и методы перечисления

Конструктор в enum по умолчанию приватный, то есть имеет модификатор **private**. (нельзя сделать его **public**)

```
enum Color {  
    RED("#FF0000"), BLUE("#0000FF"), GREEN("#00FF00");  
    private String code;  
  
    Color(String code) {  
        this.code = code;  
    }  
    public String getCode() {  
        return code;  
    }  
}
```

Методы в качестве констант

```
Operation op = Operation.SUM;  
System.out.println(op.action(10, 4)); // 14
```

```
enum Operation{  
    SUM{  
        public int action(int x, int y){ return x + y;}  
    },  
    SUBTRACT{  
        public int action(int x, int y){ return x - y;}  
    }  
};  
public abstract int action(int x, int y);  
}
```

Общие замечания

- все константы определенные в перечислении являются `public static final`
- доступ к этим переменным можно получить через `EnumName.instanceName;` (ИмяПеречисления.имяПоля)
- нельзя создать экземпляр `Enum`, можно только полагаться на его константы
- перечисления можно использовать в `switch-case`
- для сравнения `enum` используется `"=="`. Например `size == Size.small`