PDP
IT-ACADEMY

# Javada Enum (Enumeration)

PDP
IT-ACADEMY

# Reja:

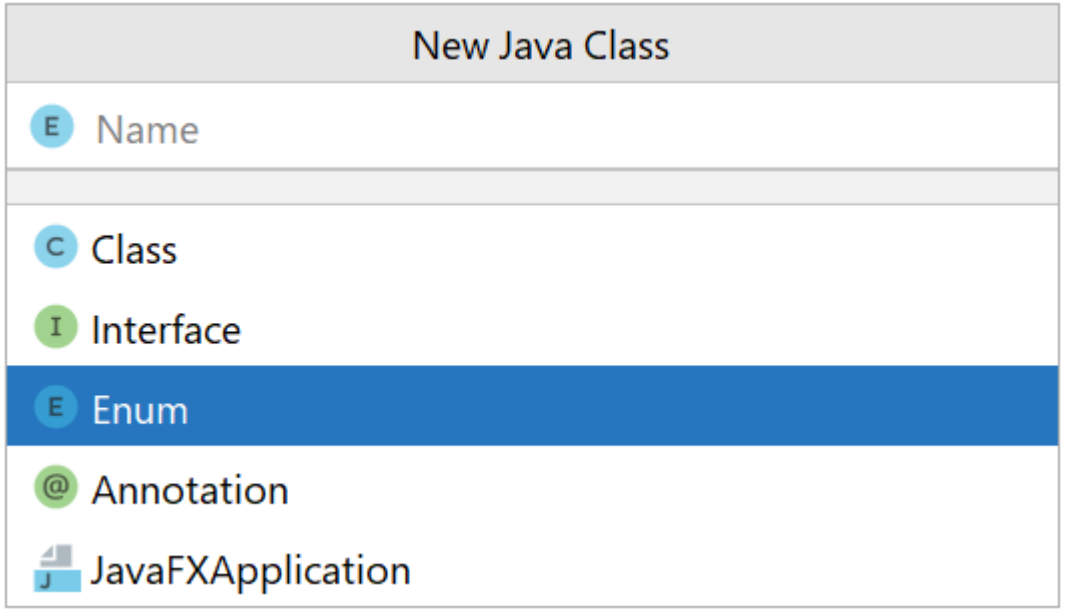
1. Hayotda uchraydigan cheklangan holatlar
2. Enum
3. Methodlari
4. EnumSet
5. Implementing an Interface

# Hayotda uchraydigan cheklangan holatlar

- Hayotda biz ayrim  hollarda cheklangan variantlardan iborat tanlovni oshirishimizga to’g’ri keladi. Masalan bunga aloqa operatorlari va internet provayderlarning ta’riflari yoki restoran yoki kafening ovqatlar menyusini misol qilib keltirishimiz mumkin.
- Javada shunday cheklangan tanlovlarni aynan  **Enum** orqali amalga oshirish mumkin.  Ya’ni Enum bizga keraklisini  tanlash mumkin bo’lgan cheklangan qiymatlardan iborat menyuni yaratishga imkon beradi.

# Enum

**Enum** - bu struktura bo’lib alohida faylda saqlanishi mumkin yoki class ning tarkibiy qismi ham bo’lishi mumkin. Uni xuddi yangi klass yaratganday yaratiladi. Intellij Idea da u alohida band qilib kiritilgan bo’lib **Class** o’rniga **Enum** tanlanadi:



PDP
IT-ACADEMY

- Agarda **enum** biror klassning tarkibiy qismi bo’lmasa uning access modifikatori **public** bo’lishi shart; Agarda uni **private** qilmoqchi bo’lsangiz xatolik yuz beradi;
- Agarda **enum** class ichida e’lon qilingan bo’lsa uning access modifikatori **private** bo’lishi mumkin;
- Java class bo’lib `java.lang.Enum` classining barcha methodlaridan voris oladi;
- Boshqa hech qanday classdan voris ololmaydi, lekin interface larni realizasiya qilishi mumkin;
- O`zgaruvchilari avtomatik tarzda public static final bo`ladi va ularni o`zgartirib bo`lmaydi;
- Constructori bo`lishi mumkin lekin u **public** va **protected** bo`maydi (private);
- Switch bilan ishlatish mumkin;
- Class ichida elon qilinsa default holatda static bo`ladi;
- new() operatorini enum bilan ishlatish taqiqlangan, hattoki uning ichida ham;
- **Serializable va Comparable** interfacelarini implement qilgan;
- Ikki xil usulda taqqoslash mumkin equals va ==; == bilan null ni ham taqqoslasa bo`ladi.

PDP
IT-ACADEMY

**Enum** ning barcha objectlarini katta harflar bilan nomlash qabul qilingan bo’lib ular “,” bilan ajratiladi va oxirgi objectdn keyin “;” qo’yiladi.

```
public enum Season  {
    WINTER,
    SPRING,
    SUMMER,
    AUTUMN;
}
```

PDP
IT-ACADEMY

# Methodlari

equals()

hashCode()

toString()

clone()

name() – nomini qaytaradi

ordinal() – tartib raqamini qaytaradi

values()-qiymatlarini massiv shaklda qaytaradi

valueOf()-stringdan enum yaratadi

PDP
IT-ACADEMY

- **`name()`** – enum nomini string shaklda qaytaradi.

```
Seasons s = Seasons.AUTUMN;
System.out.println(s.name // AUTUMN
```

PDP IT-ACADEMY

- **ordinal()** – enum qiymatning tartib raqami.

```
Seasons s = Seasons.AUTUMN;
System.out.println(s.ordinal()); // 3
```

Misol: Muzikantlar sonini toping

```
public enum Ensemble {
    SOLO, DUET, TRIO, QUARTET, QUINTET,
    SEXTET, SEPTET, OCTET, NONET, DECTET;

    public int numberOfMusicians() {
        return ordinal() + 1;    }
}
```

PDP
IT-ACADEMY

- ***values()*** – enum ning barcha qiymatlari

```
for (Seasons s : Seasons.values())
  {
      System.out.println(s);
}
```

PDP
IT-ACADEMY

- **valueOf(String)** – parameter qilib berilgan satrga mos enum ni qaytaradi

```
Seasons.valueOf("WINTER"); // Seasons.WINTER
Seasons.valueOf("Winter"); // java.lang.IllegalArgumentException!!!
```

PDP
IT-ACADEMY

# EnumSet

```
Set<Season> seasons = EnumSet.allOf(Season.class);
System.out.println(seasons); //output: [WINTER, SPRING, SUMMER,  AUTUMN]
```

```
EnumSet<Season> range = EnumSet.range(WINTER, SUMMER);
System.out.println(range); //output: [WINTER, SPRING, SUMMER]
```

# Implementing an Interface

```
public interface Operator {
   int calculate(int firstOperand, int secondOperand);
}
```

```
public enum EOperator implements Operator {
  SUM {
     @Override public int calculate(int firstOperand, int
secondOperand) {
              return firstOperand + secondOperand
     }
  },
  SUBTRACT {
     @Override public int calculate(int firstOperand, int
secondOperand) {
              return firstOperand - secondOperand;
     }
   };
}
```

PDP IT-ACADEMY

```
public class Operation {
  private int firstOperand;
  private int secondOperand;
  private EOperator operator;
  public Operation(int firstOperand, int secondOperand, EOperator operator) {
    this.firstOperand = firstOperand;
    this.secondOperand = secondOperand;
    this.operator = operator;
  }
  public int calculate(){
    return operator.calculate(firstOperand, secondOperand);
  }
}

public class Main {
  public static void main (String [] args){
    Operation sum = new Operation(10, 5, EOperator.SUM);
    Operation subtraction = new Operation(10, 5, EOperator.SUBTRACT);
    System.out.println("Sum: " + sum.calculate());
    System.out.println("Subtraction: " + subtraction.calculate());
  }
}
```

PDP
IT-ACADEMY

# E’TIBORINGIZ UCHUN RAXMAT