Enumerations Types :-

Defs- A java Enum is a Special Java Type used to define Collections of Constants and we called it a Special Class That represents a group of Constants (unChangeable Variables Like final Variables.

you may be didn't understand!

let me show you what I mean:

ex Day { SAT, SUN, MON, TUE, WED, THU, FRI }

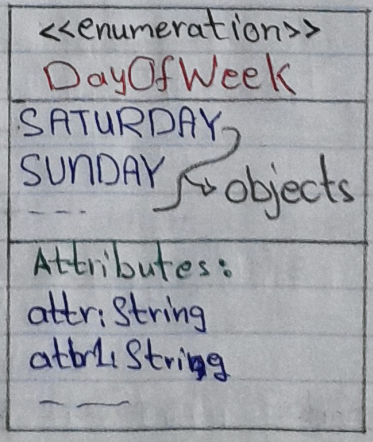
Here a Collection of Constants represents Days of the week.

we can write Enum UML Diagram as followings:

and we define Enum as following Syntax:

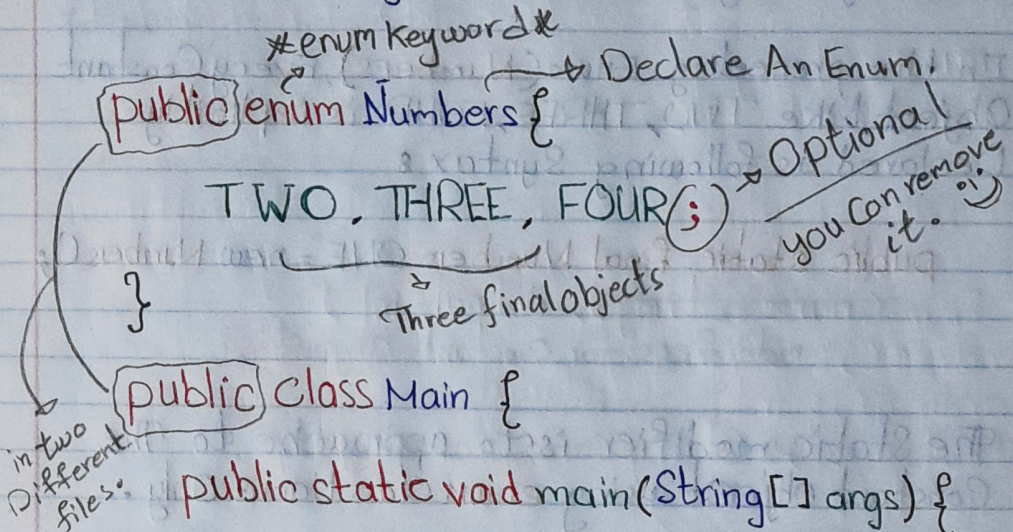
enum enumName {Constant objects}

you can write an enum in or out of The Classes and all ~~its objects~~ objects Their Objects are UpperCase letters To Consider Them as Constants.



I will show you The benefits of the Enum
See the following example:

ex:



Output: 2

as The previous example you can use The Enum with any thing needs a constants.

In the preceeding Enum(Numbers), Every Constant Object like TWO, THREE, FOUR is implicitly Declared as following Syntax :

```
public static final Numbers ONE = new Numbers();
```

modifiers for Constant Objects.

The Static modifier is to approache to The Constant object using enum class Name only.

The most powerful features of JAVA Enum is to use it as a regular class show the following example :

ex:

```
public enum WeekDays {
    MONDAY("workingday"), TUESDAY, WEDNESDAY,
    THURSDAY, FRIDAY, SATURDAY, SUNDAY;
    private String status;
    WeekDays() {}
    WeekDays(String status) { this.status = Status;}
    public void setStatus(String status){
        this.Status = Status;
    }
    Public String getStatus() { return status; }
}
```

note: Any Enum you created, extends ~~from~~ Enum abstract class.

if we used an abstract method in any enum we should Override it in every Constant object as following example:

ex:

```
        ↱ Already Abstract*
enum WeekDays {

    Day {
                    ↱ Override the TestMethod.
      @Override
      public void testMethod() {
          System.out.println("Override me please!");
      }
    }
                         ↙ Abstract Method
    public abstract void testMethod();
}
```

Remember that java dosen't provide Multiple ~~Inheritance~~ Inheritance so any enum Cannot extend any Class Cause it's already extends Enum Class.

note: Any Constructor in the enum is private access modifier are Changeable to private access modifier

now we want to Know about Enum Class:

java.lang.Enum<E>;

Parameters:

- name : String
- ordinal : int

} <<final>>

behaviours:

Enum(name : String, ordinal : int)
+ name() : String
+ ordinal() : int
+ toString() : String
+ ~~equals() : boolean~~
+ ~~compareTo()~~
+ equals(~~Object~~ other : Object) : boolean
+ CompareTo(o : E) : int
+ valueOf(enumType<T> Class, name : String) : T
+ values() : ~~enum~~ class<T>[]

} <<final>>

ordinal(); Returns The index of The constant object.

compareTo(); Return The index of high order of Two Constant objects.
→ index1 (this) - index2 return it

toString(); Return The name of constant object as a String.

name(); The Same as toString

valueOf("String"); Return The representative constant object of String.

values(); Returns an Array Consisting all Constant objects of the enum.

notes you can Compare with two Constant objects using == operator as following Syntax:

```
WeekDays.MONDAY == WeekDays.SUNDAY;
```

now we have Two Important data Structures for enumrations EnumSet and EnumMap from java.util Package.

Let me show you how to declare one of them as following Syntax:

```
EnumSet<WeekDays> days;
```

we can use of() method of EnumSet to add some Constant objects as following Syntax:

```
EnumSet.of(ConstantObject);
```
→ you can add objects separated by Comma

and if you want to add more als a group or all Constant Objects of some enumration use allof() method as following Syntax:

```
EnumSet.allOf(enumclass);
```
→ Type of the enum

we have range(from, To) method To enter enum Constants of some range by the begining and the end, and noneOf() which fill all with [null].

and we can use add(Constant), addAll(Constant list) for adding one constant or a list of some constants and we can use remove(Constant), removeAll(List) To remove one Constant or remove a whole list of Some Constants.

we have another datastructure called EnumMap The Same as EnumSet but its storing with every object a key like pairs as following Syntax:

```
EnumMap<WeekDays, String> daysMap;
        constant  key
```

Note: EnumSet is Abstractclass So you can not create it using new Operator but EnumMap is aregularclass which you can create an instanceofit using new operator.