### Enums in Java

\*\*Enums\*\* (short for enumerations) are a special Java type used to define collections of constants. They make code more readable and less error-prone by providing a way to use descriptive names for sets of related values.

### Creating an Enum

To define an enum in Java, use the `enum` keyword. Here’s a short example:

```java

public enum Day {

SUNDAY,

MONDAY,

TUESDAY,

WEDNESDAY,

THURSDAY,

FRIDAY,

SATURDAY

}

```

### Using an Enum

You can use enums in your code to refer to constants. Here’s how you can use the `Day` enum:

```java

public class EnumExample {

public static void main(String[] args) {

Day today = Day.WEDNESDAY;

switch (today) {

case MONDAY:

System.out.println("Start of the work week");

break;

case FRIDAY:

System.out.println("End of the work week");

break;

case SATURDAY: case SUNDAY:

System.out.println("Weekend!");

break;

default:

System.out.println("Midweek day");

break;

}

}

}

```

### Explanation

1. \*\*Defining an Enum\*\*: The `Day` enum lists the days of the week.

2. \*\*Using an Enum\*\*: In the `EnumExample` class, we assign `Day.WEDNESDAY` to the variable `today`.

3. \*\*Switch Statement\*\*: We use a `switch` statement to perform different actions based on the value of `today`.

Enums are powerful and can include fields, methods, and constructors for more complex use cases, but the example above covers the basics.

Example 2 :

public enum Coin {

    PENNY(1),

    NICKEL(5),

    DIME(10),

    QUARTER(25);

    public int value;

    Coin(int value) {

        this.value = value;

    }

}