UC Leuven Limburg

MOVING MINDS

Enum

Task

Imagine you have to write a class Date...

- How to save the month?

Instance variable

```
public class Date {
        ...
        private int month = 1;
        ...
}
```

Task

Imagine you have to write a class Date…

- How to save the month?
- toString() has to show: “21 JULI 2010”

toString()

```
public class Date {
    ...
    private int month = 1;
    ...
    public String toString(){
        String monthString = " ”;
        switch(month){
            case 1: monthString = "JANUARY"; break;
            case 2: monthString = "FEBRUARY"; break;
            case 3: monthString = "MARCH"; break;
            ...
        }
        return getDay() + " " + monthString + " " + getYear();
    }
}
```

Task

Imagine you have to write a class Date…

- How to save the month?
- toString() has to show: “21 JULI 2010”
- Date has to be valid

Validation

```
public class Date {
    ...
    private int month = 1;
    ...

    public boolean isValidDate()
    {
        if((month == 1 && 31 < day) || (month == 2 && 29 < day) ||
         (month == 3 && 31 < day) || (month == 4 && 30 < day) ||
         (month == 5 && 31 < day) || (month == 6 && 30 < day) ||
         (month == 7 && 31 < day) || (month == 8 && 31 < day) ||
         (month == 9 && 30 < day) || (month == 10 && 31 < day) ||
         (month == 11 && 30 < day) || (month == 12 && 31 < day) && ...)
        {
            return false;
        }
    }
}
```

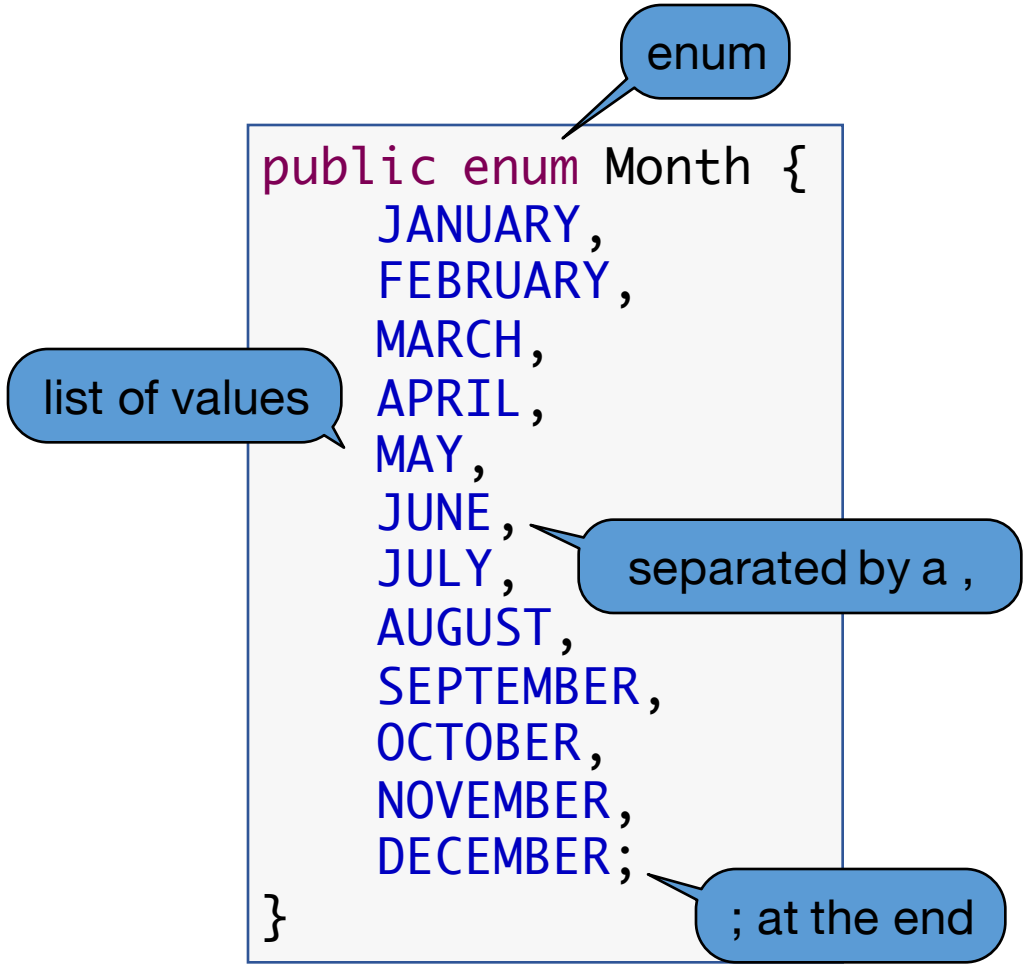
Alternatives?

Enum

- Special kind of class
- `enum` instead of `class`
- Fixed set of objects
  - months
  - wind directions
  - days of the week
  - …

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

```
public enum WindDirection {
    NORTH,
    EAST,
    SOUTH,
    WEST;
}
```



Instance variable revisited

```
public class Date {
        ...
        private Month month = Month.JANUARY;
        ...
}
```

toString() revisited

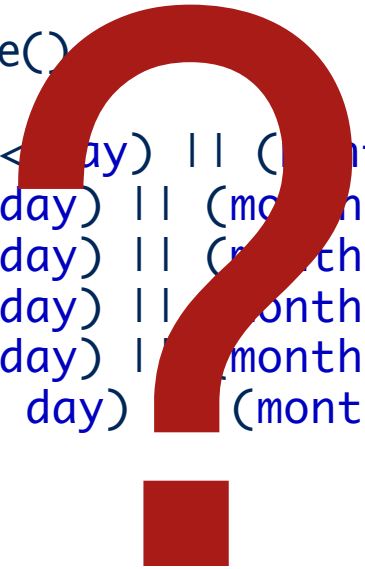
```
public class Date {
        ...
        private Month month = Month.JANUARY;
        ...

        public String toString(){
            return day + " " + month.toString() + " " + year;
        }
}
```

Validation revisited

```
public class Date {
    ...
    private Month month = Month.JANUARY;
    ...

    public boolean isValidDate()
     {
         if((month == 1 && 31 < day) || (month == 2 && 29 < day) ||
          (month == 3 && 31 < day) || (month == 4 && 30 < day) ||
          (month == 5 && 31 < day) || (month == 6 && 30 < day) ||
          (month == 7 && 31 < day) || (month == 8 && 31 < day) ||
          (month == 9 && 30 < day) || (month == 10 && 31 < day) ||
          (month == 11 && 30 < day) || (month == 12 && 31 < day) && ...)
         {
             return false;
         }
     }
}
```



Month revisited

```
public enum Month {
    JANUARY (31),
    FEBRUARY (29),
    MARCH (31),
    APRIL (30),
    MAY (31),
    JUNE (30),
    JULY (31),
    AUGUST (31),
    SEPTEMBER (30),
    OCTOBER (31),
    NOVEMBER (30),
    DECEMBER (31);

    private int nbrOfDays;

    private Month(int nbrOfDays){
            this.nbrOfDays = nbrOfDays;
    }
    public int getNbrOfDays(){
            return nbrOfDays;
    }
}
```

extra instance variable

never called

constructor

getter

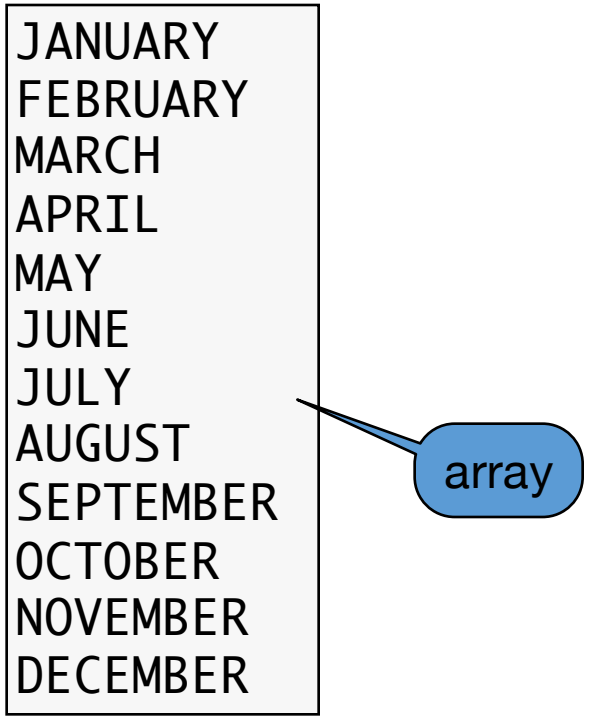
Validation revisited

```
public class Date {
    ...
    private Month month = Month.JANUARY;
       . . .

    public boolean isValidDate()
    {
       if(day > month.getNbrOfDays() || ...)
       {
           return false;
       }
    }
}
```

values()

```
...
public void printMonths(){
        for(Month m : Month.values()){
                System.out.println(m);
        }
}
```



valueOf()

- String → enum

```
String monthAsString = "September";
String monthAsUppercaseString = monthAsString.toUpperCase();
Month monthAsEnum = Month.valueOf(monthAsUppercaseString);
```

?

- See also: *Example use of enum in Java*