CMSC131

Enumerated Types

Enumerated Types

There are times when a programmer might want to have a natural way to represent and use a small set of words as if they were primitive values.

In Project 6, we used numbers to represent suits for the cards. Would you have preferred it if you could have used their names?

Enumerated types can essentially allow you to define a list of new human-friendly constants.

Enumerated Type: Suits

Imagine if in Project 6, we had the following enumerated type defined:

```
public static enum Suits {
 SPADES,HEARTS,CLUBS,DIAMONDS,STARS
}
```

Your code could then have used `Suits` as a type the same way we use other data types for variable names, fields, parameters, etc.

Syntax Examples

Assignments to local variables might look like:

```
    Suits localSuit = Suits.CLUBS;
```

Comparisons to fields might look like:

```
    if (this.mySuit==Suits.HEARTS)
```

Loops to do something for each enumerated value might look like:

```
for (Suits val:Suits.values()){
```

More advanced uses of enum

In later courses you might see ways that Java’s approach for enumerated types can support additional fields and methods and become a more complex for of user-defined constants.

Copyright © 2011-2019 : Evan Golub