THE UNIVERSITY OF MELBOURNE

week10

COMP90041 Programming and software development

Zhe(Zoe) Wang

github: https://github.com/Zoeewang/COMP90041-2020-sem1-tutorial

Enumerated type

- is a type whose values are all specific constants
- Form: enum typename {value1, value2,…}
- place this at top level inside a class, or in a file by itself, named *typeName.java*
- Convention: values all uppercase, words separated by underscores
- Example:
- enum DayOfWeek {SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY};

Using enumerated types

- with enum declaration, typeName is a valid type
- Each value is a constant reffered to as typeName.value
- use == or != to compare enum values for equality

compareTo

int a.compareTo(b)
Returns a negative value if the calling object "less than" the argument in the list of values, returns zero if the calling object equals the argument, and returns a positive value if the calling object "greater than" the argument.

For enum type: "less than" means appearing earlier in the declaration

What this will return?? positive/negative?

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

DayOfWeek.SATURDAY.compareTo(DayOfWeek.FRIDAY)
```

Methods

- Java implements enums as classes

- they automatically have useful public methods:
  - String toString()
  - static type valueOf(String) returns enum value of string
  - int ordinal() returns 0 for the first value, 1 for the second....
  - static type[] values() returns an array of all the enum values for the type, in order

  - values is very useful, as it allows you to loop over all values of an enum type

Copy constructor

a constructor that takes one argument of the same type as the object being constructed

- just make the new object an exact copy of the input argument

```
public Dog(Dog orig){
    this.age = orig.age;
    this.name = orig.name;
}
```

Q1

Write a Java program that prompts the user for one integer. Use a try/catch block to handle the InputMismatchException.

Q2

Define an Exception class called NegativeNumberException.
The class should have a constructor with no parameters.
If an exception is thrown with this zero-argument constructor, the getMessage() method should return “Negative Number Not Allowed!” This class should also have a construction with a single parameter of type String. If an exception is thrown with this construction, then the getMessage() method returns the value that was used as an argument to the constructor.

write a program prompts the user for one non-negative integer

Q3

Revise the program in Exercise 1 above to throw a NegativeNumberException if the user enters a negative number.

Thank you