ITEC1620
Object-Based Programming

Lecture 8

Programming with Objects

Review

- Sequence, Branching, Looping
- Primitive datatypes
- Mathematical operations
  - Four-function calculator
- Scientific calculator
  - Don’t program the functions yourself

Math Class

- Java set of pre-defined mathematical functions and constants
  - Constants (E, PI)
  - Basic functions
    - round, floor, ceil, min, max, abs
  - Trigonometric functions
    - sin, cos, tan, asin, acos, atan
  - Exponential funcions
    - pow, log

Functions

- Blocks of JAVA code that provide a specific, reusable functionality

static double sqrt (double a)

static double pow (double a, double b)

Functions and APIs

- Application Programmer Interfaces (APIs) list the available functionality of a class
- An API includes
  - Constants, constructors, and methods
  - Datatypes
  - Argument lists with datatypes
  - Descriptions of functionality

Constants

static double PI

The double value that is closer than any other to *pi*, the ratio of the circumference of a circle to its diameter.

- static ➔ “global variable”
- double ➔ datatype
- PI ➔ variable name (identifier)

Functions

static double sqrt (double a)

Returns the correctly rounded positive square root of a double value.

- static ➔ “stateless function”
- double ➔ datatype of result
- sqrt ➔ name (identifier) of method
- double a ➔ datatype of parameter

Sample Program I

- Calculate the radius or a circle with area = 1

$$area = \pi r^2$$

$$r = sqrt(area/\pi)$$

Program

```
public class Radius
{
   public static void main (String[] args)
   {
        double temp = 1/Math.PI;
        double radius = Math.sqrt(temp);
   }
}
```

Keyboard Input

- Scanner class
  - Provides functionality in Java to get user (keyboard) input
  - Constructors
    - Build new Scanner objects
  - Methods
    - Read data

Methods to Read Data

int nextInt()

Scans the next token of the input as an int.

- int ➔ datatype of result
- nextInt ➔ name (identifier) of method

- Not static!

static vs. non-static methods I

- static
  - Same result for same parameters, every time
  - “stateless”
- non-static
  - Result depends on state

static vs. non-static methods II

nextInt()

- Should this method always return the same result?
  - Which input source?
  - Which item in input series?
- Proper operation of method requires “state” information
- “State” information stored in objects

Creating Objects

Scanner (InputStream source)

Constructs a new Scanner that produces values scanned from the specified input stream.

- Scanner ➔ type of object being created
- source ➔ “state” information for new object

Example

```
import java.util.Scanner;

public class Input
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner (System.in);
        int input = scan.nextInt();
    }
}
```

Random Numbers

- Random class
  - Provides functionality in Java to generate (pseudo) random numbers
  - Constructors
    - Build new Random objects
  - Methods
    - Generate (pseudo) random numbers

Creating Objects

Random (long seed)

Creates a new random number generator using a single long seed

- Random ➔ type of object being created
- seed ➔ “random” number sequence
  - Two Random objects with the same seed will produce the same sequence of “random” numbers

Example

```
import java.util.Random;

public class RandomNumbers
{
    public static void main (String[] args)
    {
        Random generator = new Random (1);
        int randomInt = generator.nextInt();
    }
}
```

Function Names

int input = scan.nextInt();

- Gets next input from user

int randomInt = generator.nextInt();

- Generates next random number

- The meaning of nextInt() depends on context

Sample Program I

- Simulate the rolling of two six-sided dice

Program

```
import java.util.*;

public class TwoDice
{
    public static void main (String[] args)
    {
        Random generator = new Random ();
    }
}
```

Sample Program II

- Generate a random Pick3 or Pick4 number

Data

Processes

Program

```
import java.util.*;

public class Lotto
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner (System.in);
        Random generator = new Random ();
    }
}
```

Sample Program III

- Generate a random number between 1 and 10, and take user guesses until they guess the number. Output the number of guesses.

Data

Processes

Program

```
import java.util.*;

public class GuessingGame
{
    public static void main (String[] args)
    {
        Scanner scan = new Scanner (System.in);
        Random generator = new Random ();
    }
}
```

Readings and Assignments

- Text sections (5th, 6th, or 7th edition)
  - 2.6, 3.4, 3.5
- Tutorials – Branching Programs in JAVA
- Tutorials – Looping Programs in JAVA
- Mid-term Sample 1
- Lab Assignment 3