5. Function (2) and Exercises
August 9, 2015

Minhaeng Lee
Random Number

• Generate random number between 0 to 1.0
  – `Math.random();`

• Generate random number between 0 to N
  – `(int)(Math.random() * N);`

• **Exercise:** Try to print 10 times of randomly generated numbers using loop statement
Exercise : coin gambling

• Make function that return true/false
• Inside of the function, do random between 0 to 1, if the value is less than 0.5 then return true, otherwise return false

• Run 1000 times, count true and false then compute mean of them.
Factorial

• Factorial
  
  \[ N! = N\times(N-1)\times(N-2)\times\ldots \]

• Define function
  
  \[ \text{public static int factorial(int N)}\{ /* fill me! */ } \}

• Exercise : Using loop statement
Recursion

• Call function itself
• Need termination phrase
Recursion Example: Factorial

```java
public static int factorialIter(int N){
    int var=1;
    for(int i = N ; i > 0 ; i--)
        var *= i;
    return var;
}

public static int factorialRecur(int N){
    if(N == 1) return 1;
    else return N*factorialRecur(N-1);
}
```
피보나치 수는 수학에서 아래의 점화식으로 정의되는 수열이다.

\[
F_n := \begin{cases} 
0 & \text{if } n = 0; \\
1 & \text{if } n = 1; \\
F_{n-1} + F_{n-2} & \text{if } n > 1.
\end{cases}
\]

피보나치 수는 0과 1로 시작하며, 다음 피보나치 수는 바로 앞의 두 피보나치 수의 합이 된다. 
\( n = 0, 1, \ldots \)에 해당하는 피보나치 수는 (OEIS의 수열 A000045)

\[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946\ldots\]

이다.
Write Fibonacci

• Iterative way

• Recursive Way
Function Overloading

• To make functions *Same name* but *different body*

• Why?
  – Sometimes, we need make multiple functions have similar name for program simplicity
  – Example: add functions using integer input and String inputs. They should have similar name but different body to process different inputs

• For overloading, *at least one of parameters must be different* to the others
Overloading. When do we need?

```java
public int add(int a, int b){
    return a+b;
}

public int add(String a, String b){
    return Integer.parseInt(a)+Integer.parseInt(b);
}

public int integerAdd(int a, int b){
    return a+b;
}

public int stringAdd(String a, String b){
    return Integer.parseInt(a)+Integer.parseInt(b);
}
```

They have similar function name but parameter types are different.
Overloading Example

```java
public int add(int a, int b){
    return a+b;
}

public int add(int a2, int c2){
    return a2+b2;
}

public double add(int a, int b){
    return a+b;
}

public double add(int a2, double b2){
    return a2+b2;
}
```
Static?

- Compile-time
- Run-time
- Memory loading timing
- Usually variable/functions will be loaded to memory when it reached
- *static* keywords infers
  - The variable/function will be loaded at the beginning of running

Why does main have *static* keyword?
Useful library Vector

• Array has size limit (defined at the first time)
• But sometimes, we cannot expect its optimized size
• By using Vector, we don’t have to think about it
  • `Vector<Type> <variable Name> = new Vector<Type>();`
Vector Example

Type it and see how it works

```java
import java.util.Vector;

public static void main(String[] args) {
    Vector myList = new Vector();
    myList.add(10);
    myList.add(15);
    System.out.println(myList);
    System.out.println(myList.get(0));
    myList.set(0, 20);
    System.out.println(myList.get(0));
}
```
Vector Example
Type specified

Type it and see how it works

```java
public static void main(String[] args) {
    Vector<Integer> myList = new Vector<Integer>();
    myList.add(10);
    myList.add(15);
    System.out.println(myList);
    System.out.println(myList.get(0));
    myList.set(0, 20);
    System.out.println(myList.get(0));
}
```

Type cannot be primitive types. Use Integer instead of int, Double instead of double, Float instead of float
Exercise Time!

• You have Two options

• 1. Write “Big Number Calculator”

• 2. Complete
Homework 2 Exercise (type 1): Big number calculator

• How can we store a number over the memory size of variable types?
  – We know the upper-bound of variables

• Store each digit of input number as individual letter

• Then compute add, minus, multi, divide as we’ve learned in elementary school
Exercise Time!

• You have Two options

• 1. Write “Big Number Calculator”

• 2. Complete small exercises
  – Into, data types, loop, and arrays
  – You can waive two exercises in “loop” and “array” section
  – You should log-in to save your source code.
Exercises (type 2)

- Do those exercises!

Intro and Cmd Line Args
Data Types
Loops and Conditionals
Arrays

HelloWorld TwoPlusTwo NameAge NextYear SquareOf SquareSwap Eggsactly PercentScore PizzaCalculator AboveAverage Distance ThreeSort Flag PowersOfTwo FiniteSum AgeChecker SquareCensus Quadratic ModularSqrt Distinct Reverse NOrdered NSwap Commonest Students