Problem Solving With Java #1

By Kevin 2022/2023

Outline

- Throwing a few programmes and see how they can be implemented using only simple Java **primitives**
- These **primitives** can be founded from the flipped classroom videos.

Making appointment

- Four students (Alice, Bob, Carol, Dave) are making appointment with a professor. Professor has x time slots available.
- Each student enters his/her choice into a program
- The program will check if their inputs are valid (enter an integer from 1 to x); and all inputs are distinct.

```
Professor has 5 free time slots (x=5):
Alice, enter your choice:1
Bob, enter your choice:3
Carol, enter your choice:1
Dave, enter your choice:4
There are invalid inputs! Please enter again:
Alice, enter your choice:4
Bob, enter your choice:3
Carol, enter your choice:5
Dave, enter your choice:2
Appointments made!
```

Making appointment

Essential Ingredients

- Scanner
- if
- loop

> 👨‍🏫💬 This program would be simpler if we can use array. Yet, it is also not too difficult to do it without an array.

- Let's try to work on no repeat version

Making appointment - no repeat version

```
int x = 5;
Scanner scanner = new Scanner(System.in);
System.out.printf(
    "Professor has %d free time slots (x=5):\n", x);
int alice, bob, carol, dave;
boolean problem = false;

System.out.print("Alice, enter your choice:");
alice = scanner.nextInt();
System.out.print("Bob, enter your choice:");
bob = scanner.nextInt();
System.out.print("Carol, enter your choice:");
carol = scanner.nextInt();
System.out.print("Dave, enter your choice:");
dave = scanner.nextInt();
//check problem
 ...
if (problem)
    System.out.println("There are invalid inputs! Please enter again:");
else
    System.out.println("Appointments made!");
```

Making appointments - check problem

```
//check problem
//check if in-range
if (alice < 1 || alice > x)
    problem = true;
if (bob < 1 || bob > x)
    problem = true;
if (carol < 1 || carol > x)
    problem = true;
if (dave < 1 || dave > x)
    problem = true;
//check repeat
if (alice == bob || alice == carol || alice == dave)
    problem = true;
if (bob == carol || bob == dave)
    problem = true;
if (carol == dave)
    problem = true;
```

Making appointments - make it shorter

```
//check if in-range
if (alice < 1 || alice > x ||
    bob < 1 || bob > x ||
    carol < 1 || carol > x ||
    dave < 1 || dave > x)
    problem = true;
//check repeat
if (alice == bob || alice == carol ||
    alice == dave || bob == carol ||
    bob == dave || carol == dave)
    problem = true;
```

Making appointments - loop it

```
int x = 5;
Scanner scanner = new Scanner(System.in);
System.out.printf(
    "Professor has %d free time slots (%d=5):\n", x, x);
int alice, bob, carol, dave;
boolean problem = false;
do {
    System.out.print("Alice, enter your choice:");
    alice = scanner.nextInt();
    System.out.print("Alice, enter your choice:");
    bob = scanner.nextInt();
    System.out.print("Alice, enter your choice:");
    carol = scanner.nextInt();
    System.out.print("Alice, enter your choice:");
    dave = scanner.nextInt();
    //check problem
     ...
    if (problem)
        System.out.println("There are invalid inputs! Please enter again:");
} while (problem);

System.out.println("Appointments made!");
```

Generalize the problem

```
for (int i = 0; i < 100; i++) {
    System.out.printf("Student %d, enter your choice:", i + 1);
    int choice = scanner.nextInt();
    //... check problem
    if (problem) {
        System.out.printf("Student %d, your choice is invalid", i + 1);
        //... try to recover it
    }
}
```

Using `XOR` operator `^`

- Bitwise operators operates on ***bit level***.
- Assume `a = 0b0010; b = 0b0110`

Operator	Meaning	Result
`~a`	Bitwise complement operation	0b1101
`a | b`	Bitwise or operation	0b0110
`a & b`	Bitwise and operation	0b0010
`a ^ b`	Bitwise exclusive-or operation (1 if the bits are different)	0b0100
`a << n`	Bitwise left shift operation on a for n position. ($\times 2^n$)	a << 2 = 0b1000
`a >> n`	Bitwise right shift operation on a for n positions (fills the top bits with the left most bit, that is, the sign bit. $\div 2^n$)	b >> 2 = 0b0001

Bitwise Operators

```
int a,b,c,d;
//assign values to a,b,c,d; assume they are between 0 to 8
int result = 1 << a;        //line 1
result = result ^ (1 << b);//line 2
result = result ^ (1 << c);//line 3
result = result ^ (1 << d);//line 4
boolean is1234 = (result == 16 + 8 + 4 + 2);
```

- e.g. `a` = 4, `b` = 3, `c` = 1, `d` = 2. The value of result in each line:

Lines	Expression	Expression's value	`result` in binary	`result` in decimal
1	`1 << a`	0b10000	0b10000	16
2	`1 << b`	0b01000	0b11000	16 + 8
3	`1 << c`	0b00010	0b11010	16 + 8 + 2
4	`1 << d`	0b00100	0b11110	16 + 8 + 2 + 4

Number guessing game

A number guessing game behave as follows:

- A player guesses a integer between 0 to 100.
- The program says "too big", "too small", or "hooray!" if the number is bigger, smaller, or same as the secret number respectively.
- The program repeats until the number is guess correctly

```
Guess a number between 0-100: 50
Too big, try again: 24
Too small, try again: 44
Hooray!
```

Number Guessing Game

Essential Ingredients

- `Scanner scanner = new Scanner(System.in);`
- `scanner.nextInt();`
- if-else
- loops

Number Guessing Game

Starting a blank project

```
import java.util.Scanner; //added for scanner

public class NumberGuessingGame {
    public static void main(String[] argv) {
        new NumberGuessingGame().runOnce();
    }
    public void runOnce() {
        //add your code here
    }
}
```

Number Guessing Game

Let's do a version without repeat

```
public void runOnce() {
    Scanner scanner = new Scanner(System.in);
    ...
}
```

Place them in order

1. `int guess = scanner.nextInt();`
2. `if (guess ...) { }`
3. `System.out.print("Guess a number between 0-100:");`

Number Guessing Game

We haven't through about the secret value yet. Let it be 60.

```
public void runOnce() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Guess a number between 0-100:");
    int guess = scanner.nextInt();
    if (guess > 60)
        System.out.print("Too big, try again:");
    if (guess < 60)
        System.out.print("Too small, try again:");
    if (guess == 60)
        System.out.print("Hooray!");
}
```

👨‍🏫💬 Can we change the line `if (guess == 60)` to `else`?

Number Guessing Game

We don't like ***hard-code*** the value 60. It makes the many problems when we want to modify the program.

```
public void runOnce() {
    Scanner scanner = new Scanner(System.in);
    int secret = 60;
    System.out.print("Guess a number between 0-100:");
    int guess = scanner.nextInt();
    //alternative we can do it as a if-else
    if (guess > secret)
        System.out.print("Too big, try again:");
    else if (guess < secret)
        System.out.print("Too small, try again:");
    else
        System.out.print("Hooray!");
}
```

Number Guessing Game - Adding a loop

👩‍🏫💬 Two important questions: 1) What to loop? 2) When does it stop?

```
Scanner scanner = new Scanner(System.in);
int secret = 60;
System.out.print("Guess a number between 0-100:");

int guess = scanner.nextInt();
if (guess > secret)
    System.out.print("Too big, try again:");
else if (guess < secret)
    System.out.print("Too small, try again:");
else
    System.out.print("Hooray!");
```

Number Guessing Game - Adding a loop

```
Scanner scanner = new Scanner(System.in);
int secret = 60;
System.out.print("Guess a number between 0-100:");

{    //add here
int guess = scanner.nextInt();
if (guess > secret)
    System.out.print("Too big, try again:");
else if (guess < secret)
    System.out.print("Too small, try again:");
else
    System.out.print("Hooray!");
}  //or add here
```

👩‍🏫💬 Pick one: while / do-while / for-loop

Number Guessing Game - Adding a loop

```
Scanner scanner = new Scanner(System.in);
int secret = 60;
System.out.print("Guess a number between 0-100:");

do {
    int guess = scanner.nextInt();
    if (guess > secret)
        System.out.print("Too big, try again:");
    else if (guess < secret)
        System.out.print("Too small, try again:");
    else
        System.out.print("Hooray!");
}  while (guess != secret);
```

👩‍🏫💬 Almost there except `guess` is not visible.

Number Guessing Game - Adding a loop

```
public void runOnce() {
    Scanner scanner = new Scanner(System.in);
    int secret = 60;
    System.out.print("Guess a number between 0-100:");
    int guess; //define here
    do {
        guess = scanner.nextInt();
        if (guess > secret)
            System.out.print("Too big, try again:");
        else if (guess < secret)
            System.out.print("Too small, try again:");
        else
            System.out.print("Hooray!");
    } while (guess != secret);
}
```

👩‍🏫💬 Randomize by `ThreadLocalRandom.current().nextInt(0, 101);`

Random

- There are many ways to generate random number in Java

```
ThreadLocalRandom.current().nextInt(min, max);
```

- This returns a random integer that is >= `min` and < `max`.
- To generate a random double, similarly

```
ThreadLocalRandom.current().nextDouble(min, max);
```

- A random boolean

```
ThreadLocalRandom.current().nextBoolean();
```

To use this API, add the following on the top of your file.

```
import java.util.concurrent.ThreadLocalRandom;
```

```java
import java.util.Scanner; //added for scanner
import java.util.concurrent.ThreadLocalRandom; //add for random

public class NumberGuessingGame {
    public static void main(String[] argv) {
        new NumberGuessingGame().runOnce();
    }
    public void runOnce() {
        Scanner scanner = new Scanner(System.in);
        int secret = ThreadLocalRandom.current().nextInt(0, 101);
        System.out.print("Guess a number between 0-100:");
        int guess;
        do {
            guess = scanner.nextInt();
            if (guess > secret)
                System.out.print("Too big, try again:");
            else if (guess < secret)
                System.out.print("Too small, try again:");
            else
                System.out.print("Hooray!");
        } while (guess != secret);
    }
}
```

Number Guessing Game v2

- Slightly modify the program so that it also prints the range of the numbers
- If the guess value exceed the range, give a warning

```
Guess a number between 0-100: 50
Too big, try again (0-49): 24
Too small, try again (25-49): 56
Out-of-range, try again (24-49): 44
Hooray!
```

👩‍🏫💬 Extra ingredients?

Number Guessing Game v2

```
Scanner scanner = new Scanner(System.in);
int secret = ThreadLocalRandom.current().nextInt(0, 101);
System.out.print("Guess a number between 0-100:");
int guess; //define here
int min = 0, max = 100;
do {
    guess = scanner.nextInt();
    if (guess > secret)
        System.out.print("Too big, try again (" + min + "-" + max + "):");
    else if (guess < secret)
        System.out.print("Too small, try again (" + min + "-" + max + "):");
    else
        System.out.print("Hooray!");
}  while (guess != secret);
```

👩‍🏫💬 Next: update `min` and `max`. When? How?

Number Guessing Game v2

```
Scanner scanner = new Scanner(System.in);
int secret = ThreadLocalRandom.current().nextInt(0, 101);
System.out.print("Guess a number between 0-100:");
int guess; //define here
int min = 0, max = 100;
do {
    guess = scanner.nextInt();
    if (guess > secret) {  //these { } are important!
        max = guess - 1;
        System.out.print("Too big, try again (" + min + "-" + max + "):");
    } else if (guess < secret) {
        min = guess + 1;
        System.out.print("Too small, try again (" + min + "-" + max + "):");
    } else
        System.out.print("Hooray!");
}  while (guess != secret);
```

Adding condition checking

Final version

```
Scanner scanner = new Scanner(System.in);
int secret = ThreadLocalRandom.current().nextInt(0, 101);
System.out.print("Guess a number between 0-100:");
int guess; //define here
int min = 0, max = 100;
do {
    guess = scanner.nextInt();
    if (guess < min || guess > max) {
        System.out.print("Out-of-range, try again (" + min + "-" + max + "):");
        continue;
    }
    if (guess > secret) {
        max = guess - 1;
        System.out.print("Too big, try again (" + min + "-" + max + "):");
    } else if (guess < secret) {
        min = guess + 1;
        System.out.print("Too small, try again (" + min + "-" + max + "):");
    } else
        System.out.print("Hooray!");
}  while (guess != secret);
```

A shorter version

```
Scanner scanner = new Scanner(System.in);
int secret = ThreadLocalRandom.current().nextInt(0, 101);
System.out.print("Guess a number between 0-100:");
for (int guess = -1, min = 0, max = 100; guess != secret; ) {
    guess = scanner.nextInt();

    if (guess < min || guess > max)
        System.out.printf("Out-of-range, try again (%d-%d)" , min, max);
    else if (guess > secret) {
        max = guess - 1;
        System.out.printf("Too big, try again (%d-%d)" , min, max);
    } else if (guess < secret) {
        min = guess + 1;
        System.out.printf("Too small, try again (%d-%d)" , min, max);
    } else
        System.out.print("Hooray!");
}
```

Finding a prime number

- Find the next prime number that is bigger or equal to the input.

```
50
The next prime number is 53.
```

Essential Ingredients

- Scanner
- Nested loop
- if

Find prime number

```java
import java.util.Scanner; //added for scanner
public class FindPrime {
    public static void main(String[] argv) {
        new FindPrime().runOnce();
    }
    public void runOnce() {
        //add your code here
    }
}
```

Strategy: how about print the input number if it is a prime?

```
50
.
53
The next prime number is 53.
```

Find prime number

```java
public void runOnce() {
    Scanner scanner = new Scanner(System.in);
    int input = scanner.nextInt();
    //test if input is a prime
    if (...)
        System.out.println("The next prime number is " + input);
    else
        System.out.println(".");
}
```

👩‍🏫💬 But how to test if an input is a prime? **Trial-and-Error**!

Find prime number

```
Scanner scanner = new Scanner(System.in);
int input = scanner.nextInt();
//test if input is a prime
boolean isPrime = true;
for (int i = 2; i < input; i++)
    if (input % i == 0)
        isPrime = false;

if (isPrime)
    System.out.println("The next prime number is " + input);
else
    System.out.println(".");
```

👩‍🏫💬 Now, create another loop that loops forward until it gets a prime

Find prime number

```
Scanner scanner = new Scanner(System.in);
int input = scanner.nextInt();
boolean isPrime = true;

{ //loop this until there is a prime
for (int i = 2; i < input; i++)
    if (input % i == 0)
        isPrime = false;

if (isPrime)
    System.out.println("The next prime number is " + input);
else
    System.out.println(".");
}
```

Find prime number

```
Scanner scanner = new Scanner(System.in);
int input = scanner.nextInt();
boolean isPrime = true;

do {
    for (int i = 2; i < input; i++)
        if (input % i == 0)
            isPrime = false;

    if (isPrime)
        System.out.println("The next prime number is " + input);
    //we don't need the else part
    //increase the value of input by 1
    input++;
} while (!isPrime);
```

The loop is faulty, why?

Final Version

```
Scanner scanner = new Scanner(System.in);
int input = scanner.nextInt();
boolean isPrime = true;

do {
    isPrime = true; //important
    for (int i = 2; i < input; i++)
        if (input % i == 0)
            isPrime = false;

    if (isPrime)
        System.out.println("The next prime number is " + input);
    input++;
} while (!isPrime);
```