

Assignment 1

1. What is JDK? JRE? JVM?

JDK (Java Development Kit) provides software for working with Java applications, for example, the virtual machine and a compiler.

JRE: Java runtime environment. It contains the JVM to run compiled java bytecode together with standard library.

JVM: Java virtual machine. The native software that interprets Java byte code and executes it. It is part of JRE/JDK.

2. What is java compiler?

A Java compiler is a program that takes java source code and compiles it into a platform-independent Java file.

3. Why is java platform independent?

Java platform is independent because the java compiled code (byte code) can run on all operating systems. It does not depend on any type of platform. JVM is platform dependent such that the compiled Java byte code is platform independent.

4. What is IDE? Why is it important for developers?

IDE (integrated development environment) is a software for building applications. It is a software suite that combines basic tools required to write and test software.

It is important for developers because it allows the developers to start programming new applications quickly as they don't need to manually configured and integrated multiple utilities. Also, there are many features, such as intelligent code completion and automated code generation, can help the developers save time.

5. Is java case sensitive?

Java is a case-sensitive language. The upper or lower case of letters does matter.

6. What do the following key words do? static, final, public, private, void, null, package, Class, new

static: a non-access modifier used for methods and attributes. Static methods/attributes can be accessed without creating an object of a class.

final: a non-access modifier used by classes, methods, and attributes. Final makes them non-changeable (impossible to inherit or override).

public: an access modifier used for classes, attributes, methods, and constructors, making them accessible by any other class.

private: an access modifier used for attributes, method, and constructors, making them only accessible within declared class.

void: specifies that a method should have any return value.

null: it is for literal values. It represents a null reference, one that does not refer to any object.

package: creates a package. A package encapsulates a group of classes, sub packages, and interfaces. It is used for preventing naming conflicts.

Class: creates a class.

new: creates new objects. It calls the constructor to instantiate an object.

7. What is primitive type and reference type?

8. Is parameter passed by value or reference?

9. What is the output: `System.out.println(1 > 0 : "A":"B");`

This is a compile time error and there will not be any program to execute.

If you mean `System.out.println(1 > 0 ? "A": "B");`

Then the output will be "A".

10. How to define constants in java?

We use the key word final in front of declaration.

ex: `final int num = 5;` // then we cannot modify the value of num

11. What is String? Is it primitive type?

String is a sequence of characters, and it is immutable. String is not a primitive type.

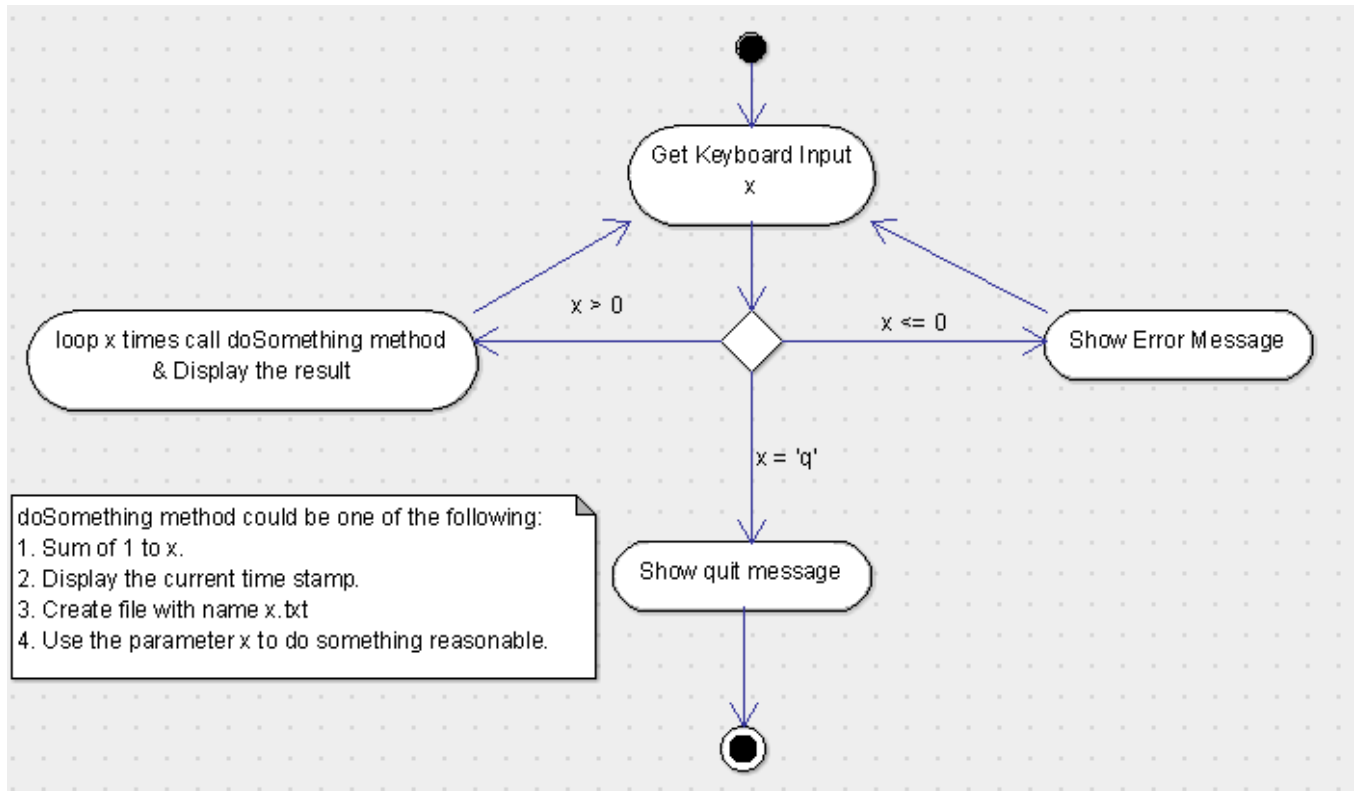
12. How to check if a String is representing a number?

We can use regular expression.

Ex:

```
public static boolean isNumber(String s) {  
    if (s == null) {  
        return false;  
    }  
    Pattern pattern = Pattern.compile("-?\\d+(\\.\\d+)?");  
    return pattern.matcher(s).matches();  
}
```

13. Write a program to implement the following activity diagram:



```

package Day2.HomeWork;
import java.util.Scanner;
import java.util.regex.Pattern;

public class ActivityDiagram {
    public static void main(String[] args) {
        func();
    }

    public static void func() {
        while (true) {
            String input = getInput();
            if (input.equals("q")) {
                System.out.println("You quit.");
                return;
            } else if (isNegNumber(input)) {
                System.out.println("Your input is a negative number. Please
input a positive number.");
            } else if (isPostiveNumber(input)) {
                soSomething(Double.parseDouble(input));
            } else {
                System.out.println("Please input a positive number.");
            }
        }
    }

    public static void soSomething(double d) {
        System.out.println("This is your input: " + d + ".");
    }
}
  
```

```

public static String getInput() {
    Scanner myObj = new Scanner(System.in);
    System.out.println("Please enter something: ");
    String input = myObj.nextLine();
    return input;
}

public static boolean isNegNumber(String s) {
    if (s == null) {
        return false;
    }
    Pattern pattern = Pattern.compile("-\\d+(\\.\\d+)?");
    return pattern.matcher(s).matches();
}

public static boolean isPostiveNumber(String s) {
    if (s == null) {
        return false;
    }
    Pattern pattern = Pattern.compile("\\d+(\\.\\d+)?");
    return pattern.matcher(s).matches();
}
}

```

14. Write a program to merge two array of int.

```

15. package Day2.HomeWork;

import java.util.Arrays;

public class MergeTwoArrays {
    public static void main(String[] args) {
        int[] a = new int[]{1,2,3};
        int[] b = new int[]{6,7,9,0};
        int[] res = merge(a,b);
        if (res != null) {
            for (int i = 0; i < res.length; i++) {
                System.out.println(res[i]);
            }
        } else {
            System.out.println("result is null");
        }
    }

    public static int[] merge(int[] a, int[] b) {
        if (a == null && b == null) { // both a and b are null
            return null;
        } else if (a == null) { // only a is null, b is not null
            return Arrays.copyOf(b, b.length);
        } else if (b == null) { // only b is null, a is not null
            return Arrays.copyOf(a, a.length);
        }

        // both a and b are not null
        if (a.length == 0 && b.length == 0) { // both a and b are empty
            return new int[0];
        }
    }
}

```

```

    } else if (a.length == 0) { // only a is empty
        return Arrays.copyOf(b, b.length);
    } else if (b.length == 0) { // only b is empty
        return Arrays.copyOf(a, a.length);
    }

    //both a and b are not empty
    int[] result = new int[a.length + b.length];
    for (int i = 0; i < a.length; i++) {
        result[i] = a[i];
    }
    for (int j = 0; j < b.length; j++) {
        result[a.length + j] = b[j];
    }
    return result;
}
}

```

16. Write a program to find the second largest number inside an array of int.

```

package Day2.HomeWork;

public class SecondLargest {
    public static void main(String[] args) {
        int[] a = new int[]{2, 4, 3, 56};
        System.out.println(SecondLargestInt(a));
    }

    public static int SecondLargestInt(int[] array) {
        if (array == null || array.length < 2) {
            return Integer.MIN_VALUE;
        }

        // array is not null and there are at least two elements in array
        int max = Integer.MIN_VALUE;
        int secondMax = Integer.MIN_VALUE;
        int index = 0;
        while (index < array.length) {
            if (array[index] > max) { // cur number is the largest so far
                secondMax = max;
                max = array[index];
            } else if (array[index] > secondMax) { // cur number is <= max
                secondMax = array[index];
            }
            index++;
        }
        return secondMax;
    }
}

```