به نام خدا

پاسخنامه تمرین اول درس برنامه نویسی پیشرفته

نیمسال دوم ۱۳۹۹-۱۴۰۰

فهرست سوالات

سوال اول

الف) Java Development Kit بسته‌ای است شامل کامپایلر جاوا و ابزارهای اشکال زدایی و توسعه برنامه‌های تحت پلتفرم جاوا و همچنین شامل نسخه ای از JRE است.

JRE یا Java Runtime Environment به معنای محیط یا بستر اجرای برنامه‌ی جاوا می‌باشد.

هر برنامه جاوا برای کامپایل شدن به JDK نیاز دارد ولی برای اجرا باید JRE روی سیستم نصب باشد در واقع JRE تنها از سوی کسانی استفاده می‌شود که صرفاً قصد دارند برنامه‌های جاوا را اجرا کنند مثلاً کاربران سیستم.

JVM یا ماشین مجازی جاوا یک بخش بسیار مهم از JDK و همچنین JRE است زیرا در هر دو آنها گنجانده شده است. هر زمان که برنامه‌ی جاوا با استفاده از JRE یا JDK اجرا می‌شود به JVM می‌رود و JVM مسئول اجرای خط به خط برنامه جاوا است ازاین‌رو به نام مفسر نیز شناخته می‌شود.

(ماشین مجازی جاوا Java bytecode را به زبان ماشین تبدیل می‌کند)

ب) در اولین مرحله ، کامپایلر جاوا کد نوشته شده رو به بایت کد تبدیل می‌کند. سپس از با استفاده از JVM موجود در JRE کد میانی (بایت کد) را به کد قابل فهم برای پردازنده تبدیل می‌کنیم. (می‌توان بایت کد را به هر سیستم عاملی منتقل نمود و برای اجرای کدمیانی سیستم عامل مقصد تنها باید نرم افزار JRE را بر روی خودش نصب داشته باشد)

ج) هشت نوع داده اولیه توسط جاوا پشتیبانی می‌شود. نوع داده‌های اولیه با زبان از پیش تعریف شده شناخته می‌شوند و با کلمات کلیدی نام گذاری می‌شوند. مانند: int, long, short, char, byte

د)

Object Oriented:

در رویکرد شی گرایی، ما با یک سری اشیا در برنامه سروکار داریم که با یکدیگر در تعامل هستند. در این پارادایم، هر چیزی را در قالب یک شیء می‌بینیم.

Functional Programming:

در این پارادایم، برنامه از اجرای یک سری توابع ریاضیاتی خالص (pure) تشکیل می‌شود. ویژگی‌ها:

- توابع موجود کاملا مستقل هستند و هیچ اثر جانبی بر روی سایر اجزای برنامه ندارند.
- در برنامه نویسی تابعی از حلقه‌ها استفاده نمی‌شود و در صورت نیاز باید از برنامه نویسی بازگشتی استفاده کنیم.
- از assignment استفاده نمی‌کنیم و متغیرها در طول برنامه تغییر نمی‌کنند.
- و ...

Structured Programming:

در این روش از برنامه‌نویسی، انجام یک روال به روال‌های کوچک‌تر تقسیم می‌شود و به این ترتیب یک برنامه با شکسته شدن به ریز برنامه‌های کوچک‌تر سعی می‌کند تا عملکرد مد نظر را پیاده‌سازی کند.
مزیت‌های برنامه نویسی‌شی‌گرا:

- قابلیت سازمان دهی بهینه‌تر کدها.
- قابلیت تقسیم برنامه به برنامه‌های کوچک‌تر اما مستقل. برنامه‌ی اصلی به صورت یک exe در می‌آید که دیگر قسمت‌های مستقل برنامه را فراخوانی می‌کند.
- عدم نیاز به نوشتن کدهای تکراری و قابلیت‌هایی که قبلاً پیاده‌سازی شده‌اند و صرفه جویی در استفاده از منابع.

اما برای پروژه‌های کوچک برنامه نویسی تابعی مناسب‌تر است.

ه)

Class	کلاس نقطه شروع همه اشیا محسوب می‌شود. کلاس را می‌توان به عنوان یک قالب برای ایجاد شیئ تصور کرد. هر کلاس به طور معمول شامل فیلدهای عضو، متدهای عضو، و یک متد خاص به نام سازنده (constructor) است.
Object	شیءها از کلاس‌ها ساخته می‌شوند و به صورت نمونه‌هایی (Instance) از کلاس نامیده می‌شوند.
Constructor	یا تابع سازنده یک متد خاص یا ویژه از کلاس است یا ساختاری در برنامه نویسی شیء گرا است به منظور مقدار دهی اولیه قرار می‌گیرد و هنگام ایجاد کلاس به صورت اتوماتیک فراخوانی می‌شود.
Method	متد در برنامه‌نویسی شیء گرا کاربرد دارد و از نظر تعریف شباهت زیادی به تابع دارد. تفاوت متد و تابع در محل تعریف آن‌هاست. متد تابعی است که داخل یک کلاس تعریف می‌شود و یک فعالیت را روی نمونه‌ی ساخته شده از کلاس به انجام می‌رساند.
Parameter	داده‌های خامی هستند که method به عنوان ورودی میگیرد و آن‌ها را پردازش می‌کند.
Instance	مفهومی مشابه object دارد. نمونه ای از یک کلاس است که شامل اطلاعات می‌شود و می‌تواند رفتارهای به خصوصی انجام دهد.

و)

تفاوت‌های اصلی:

- محل تعریف آن‌ها؛ method داخل یک کلاس تعریف می‌شود و یک فعالیت را روی نمونه‌ی ساخته شده از کلاس به انجام می‌رساند.
- method حتما بر روی یک object صدا زده میشود در حالی که function این گونه نیست.
- method برای برنامه نویسی شی گرایی و function برای برنامه نویسی ساختار یافته استفاده می‌شود.

سوال دوم

1. False, Java is statically-typed and type checking is performed during compile-time as opposed to run-time.

2. True, A Java class file is a file (with the. class filename extension) containing Java byte code that can be executed on the Java Virtual Machine (JVM).

3. False, we can’t store different types in one single array.

4. False.

5. False, Though Java is probably the most successful Object-oriented programming language, which also got some functional programming touch in Java 8, it has never been considered 100% or pure object-oriented programming language. If it were, all its primitives would be objects.

سوال سوم

A. Java classes contain **methods** (which implement class behaviors) and **fields** (which implement class/object data).

B. In Java, the unit of programming is the **classes** from which **objects** are eventually instantiated.

C. Java programmers concentrate on creating their own user-defined types, called **class**.

D. The **java** command of JRE executes an application.

E. The **javac** command of JDK compiles a Java program.

F. A Java program file must end with the **.java** file extension.

G. When a Java program is compiled, the file produced by the compiler ends with the **.class** file extension.

H. The file produced by the Java compiler contains **bytecodes** that are interpreted to execute a Java applet or application.

سوال چهارم

```
class FindMistake {
   private static void printArray(int[] arr)
   {
       int n = arr.length;
       for (int i = 0; i < n; ++i)
           System.out.println(arr[i] + " ");
       System.out.println();
   }

   private static void sort(int[] arr) {
       int n = arr.length;
       for (int gap = n / 2; gap > 0; gap /= 2)
       {
           for (int i = gap; i < n; i += 1) {
               int temp = arr[i]; //
               int j;
               for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)
                   arr[j] = arr[j - gap];
               arr[j] = temp;
           }
       }
   }

   public static void main(String[] args) {
       int[] arr = {12, 34, 54, 2, 3};
       System.out.println("Array before sorting");
       printArray(arr);
       sort(arr);
       System.out.println("Array after sorting");
       printArray(arr);
   }
}
```

الگوریتم مرتب سازی استفاده شده معروف به Shell Sort می باشد.

سوال پنجم

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a1, b1, a2, b2;
        a1 = sc.nextInt();
        b1 = sc.nextInt();
        a2 = sc.nextInt();
        b2 = sc.nextInt();
        char op = '1';
        while (op != '#') {
            op = sc.next().charAt(0);
            String res = doMath(op, a1, b1, a2, b2);
            System.out.println(res);
        }
        return;
    }

    private static String doMath(char op, int a1, int b1, int a2, int b2) {
        switch (op) {
            case '+':
                return sum(a1, b1, a2, b2);
            case '-':
                return subtract(a1, b1, a2, b2);
            case '*':
                return multiplicate(a1, b1, a2, b2);
            case '/':
                return div(a1, b1, a2, b2);
        }
         return "";
     }

     private static String sum(int a1, int b1, int a2, int b2) {
         float real = a1 + a2;
         float imaginary = b1 + b2;
```

```
        return String.format("%.2f%+.2fi", real, imaginary);
    }

    private static String subtract(int a1, int b1, int a2, int b2) {
        float real = a1 - a2;
        float imaginary = b1 - b2;
        return String.format("%.2f%+.2fi", real, imaginary);
    }

    private static String multiplicate(int a1, int b1, int a2, int b2) {
        float real = a1 * a2 - b1 * b2;
        float imaginary = a1 * b2 + b1 * a2;
        return String.format("%.2f%+.2fi", real, imaginary);
    }

    private static String div(int a1, int b1, int a2, int b2) {
        float fac = 1f / (a2 * a2 + b2 * b2);
        float real = fac * (a1 * a2 + b1 * b2);
        float imaginary = fac * (b1 * a2 - a1 * b2);
        return String.format("%.2f%+.2fi", real, imaginary);
    }
}
```

سوال ششم

کد زیر نمونه ای از پیاده سازی این سوال می‌باشد. همچنین مثال گفته شده در صورت سوال را به عنوان ورودی به تابع decrypt داده ایم:

```
public class Main {

    private static String decrypt(String input, int key){
        String output = "";
        for (char c : input.toCharArray()) {
            if (c != ' ')
                output = output.concat(Integer.toBinaryString(c ^
key));
            output = output.concat(" ");
        }
        return output;
    }

    public static void main(String[] args) {
        String binaryKey = "11110011", data = "Java J";
        int key = Integer.parseInt(binaryKey, 2);
        System.out.println(decrypt(data, key));
    }
}
```

سوال هفتم

```
import java.util.Arrays;

class Main {
    // Determine if the first string can be transformed into the
    // second string with a single edit operation
    public static boolean checkEditDistance(String first, String
second) {
        // store length of both strings
        int firstLength = first.length();
        int secondLength = second.length();

        // difference between the length of both strings is more than
one
        if (Math.abs(firstLength - secondLength) > 1) {
            return false;
        }

        // to keep track of the total number of edits
        int edits;
        int replaces = 0, additions = 0, deletions = 0;

        // `i` and `j` keep track of the index of current characters
in the first and
        // second strings, respectively
        int i = 0, j = 0;

        // `c1` and `c2` keep track of the character of last replacing
operation in the first and
        // second strings, respectively
        char c1 = 0, c2 = 0;

        // loop till either string runs out
        while (i < firstLength && j < secondLength) {

            // if the current character of both strings doesn't match
            if (first.charAt(i) != second.charAt(j)) {

                // when the length of the first string is more than
the length
                // of the second string, remove the current character
at
```

```
                // index `i` in the first string
                if (firstLength > secondLength) {
                    // increment the number of deletions
                    deletions++;
                    i++;
                }

                // when the length of the first string is less than
the length
                // of the second string, add the current character at
index `j`
                // in the second string to the first string
                else if (firstLength < secondLength) {
                    // increment the number of additions
                    additions++;
                    j++;
                }

                // when the length of both strings is the same,
replace the character
                // present at index `i` in the first string with the
character present
                // at index `j` in the second string.
                // if it is the first replacing operation, initialize
c1,c2
                // else check if we are replacing same characters in
each replacing operation()
                else {
                    if (replaces == 0) {
                        c1 = first.charAt(i);
                        c2 = second.charAt(j);
                    } else if (c1 != first.charAt(i) || c2 !=
second.charAt(j))
                        return false;

                    // increment the number of replaces
                    replaces++;
                    i++;
                    j++;
                }

            }

            // if the current character of both strings matches
            else {
```

```
                i++;
                j++;
            }
        }

        // remove any extra characters left in the first string
        if (i < firstLength) {
            deletions++;
        }

        // add any extra characters left in the second string at the
end of
        // the first string
        if (j < secondLength) {
            additions++;
        }

        edits = replaces + deletions + additions;

        if (edits == 1)
            return true;

        // no edit
        if (edits == 0)
            return false;

        // if we have more than 1 edits, it is true when we only do
replacing operation
        return replaces == edits;
    }

    public static void main(String[] args) {
        String str1 = "";
        String str2 = "";

        System.out.println(checkEditDistance(str1, str2));
    }
}
```