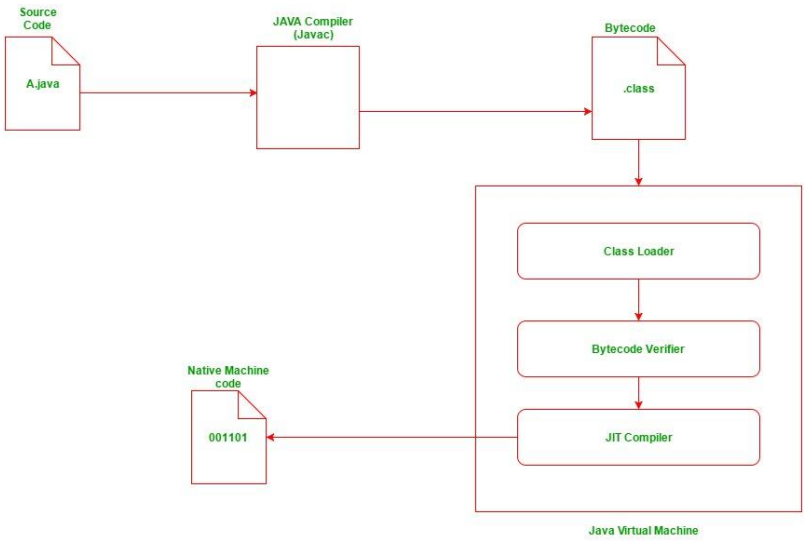
امیرحسین ایروانی‌منش 9931066 airavanimanesh@aut.ac.ir

تکلیف شماره ۱

تمرین شماره ۱

الف) اولین تفاوت بین این ۳ اصطلاح، کلمات توصیفی آنها به صورت کامل(مخفف نشده) است. JRE یا Java Runtime Environment و JDK یا Java Development Kit و JVM یا Java Virtual Machine. همانطور که از نام کاملشان پیدا است، بین این ۳، تفاوت های بسیاری وجود دارد، JDK کیت و ابزاری برای توسعه نرم افزار بر پایه جاوا است در حالی که JRE مجموعه ای از نرم افزار ها است که اجازه میدهند یک برنامه جاوا اجرا شود. JVM هم یک محیطیست ساختگی برای اجرای برنامه های جاوا. JRE و JDK وابسته به یک پلتفرم خاص هستند (یعنی نسخه های متفاوتی برای پلتفرم های متفاوت دارند) در حالی که این مسئله برای JVM صادق نیست. JDK شامل ابزار هایی برای توسعه، دیباگ و کارهایی از این قبیل است و JRE شامل کتابخانه و کلاس های مختلف است در حالی که JVM هیچ ابزاری برای توسعه نرم افزار و برنامه بر پایه جاوا ندارد.

ب) در ابتدا، فایل java. برنامه به کامپایلر پاس داده میشود، کامپایلر فایل برنامه را به زبان مستقل از ماشین (بایت کد) تبدیل میکند و محتویات هر کلاس در فایل سورس به صورت یک فایل class. تولید میشوند. بعد از آن فایل class. اصلی (که در آن متد main وجود دارد) به JVM پاس داده میشود و سپس فایل های class. ای که در فایل اصلی صدا زده شده اند، توسط class loader لود می شوند (منظور از لود شدن تبدیل شدن به بایت کد است). بعد از این مرحله، بایت کد ها به Bytecode Verifier پاس داده میشوند که کار آن چک کردن کد برای جلوگیری از ایجاد آسیب های احتمالی است. چند نمونه از فاکتور هایی که توسط Bytecode Verifier چک میشوند: ۱. درست صدا زده شدن متد ها بر اساس publicیا private بودن آنها ۲. اورفلو نکردن استک در حین اجرای برنامه و ... . مرحله بعدی و آخرین مرحله پاس داده شدن Bytecode به JIT Compiler است که مخفف Just-In-Time است که بایت کد های وریفای شده و بدون مشکل را به زبان طبیعی ماشین تبدیل میکند و سپس برنامه اجرا میشود.



ج) داده ها در زبان جاوا به دو نوع داده های ساده (Primitive data) و داده های غیر ساده (Non-Primitive data) تقسیم می شوند، داده های ساده فقط مقادیری واحدی دارند و قابلیت خاصی ندارند. در کل داده های ساده را میتوان داده هایی تعریف کرد که نمیتوان آن ها را به داده های ریز تری تقسیم و تبدیل کرد. مثال هایی از این نوع داده در جاوا : byte, short, int, long, float, double, char, Boolean.

د) در برنامه نویسی ساخت یافته، از تابع‌ها و ترکیب آنها برای نوشتن برنامه استفاده می‌شود. برای مثال یک برنامه به چند تابع تقسیم میشود که هر تابع مسئول انجام قسمتی از پردازش اطلاعات است و در قسمت‌های مختلف برنامه استفاده می‌شوند. اما در برنامه نویسی شیء گرا برنامه به چند بخش مخلف تقسیم شده که متشکل از کلاس‌ها و متدهاست. در این روش اجزای اصلی اجرای برنامه نمونه‌هایی از کلاس‌های تعریف شده می‌باشند که در ادامه آن بخش‌های فرعی اعم از پردازش اطلاعات با استفاده از متدهای هر شیء ساخته شده انجام می‌شود. برای مثال برای انجام یک پردازش روی یک استرینگ میتوان از متد تعریف شده در کلاس اصلی استریگ استفاده کرد و نیازی به نوشتن تابع اضافی یا صدا زدن تابع خارجی برای آن نیست. و از برنامه نویسی تابعی برای محاسبه عملیات‌های ریاضی و دوری از داده‌های قابل تغییر استفاده می‌شود.

ه) کلاس: به طور کلی، به مجموعه کدی گویند که برای یک هدف نوشته شده اند و در کنار یکدیگر قرار گرفته اند. آن را میتوان به عنوان الگو یا Pattern خاصی گرفت. برای مثال میتوان ماشین را یک کلاس در نظر گرفت.

آبجکت: یک شی یک نمونه از کلاس است که بر اساس اون کلاس ساخته شده و ویژگی های آن کلاس را داراست.

کانستراکتور: سازنده (constructor) شبیه به متد است (اما در واقع متد نیست) که هنگام ایجاد شی فورا به صورت خودکار فراخوانی می شود.

متد: یکی از فیچرهایی که در زبان‌های برنامه‌نویسی به منظور انجام تَسک‌های تکراری و کدنویسی اصولی‌تر گنجانده شده است مفهومی تحت عنوان Method می‌باشد.

پارامتر: پارامتر داده ای است که می توان آن را به متد پاس داد و پردازش های مورد نظر روی آن انجام شود.

نمونه:

و) متد مانند تابع است که کار های مشخصی را انجام میدهد، تنها تفاوت آن با تابع این است که متعلق و مرتبط به یک شی است.

امیرحسین ایروانی‌منش 9931066 airavanimanesh@aut.ac.ir

تکلیف شماره ۱

تمرین شماره ۲

الف) غلط.

ب) صحیح.

ج) غلط.

د) غلط.

ه) غلط.

امیرحسین ایروانی‌منش 9931066 airavanimanesh@aut.ac.ir

تکلیف شماره ۱

تمرین شماره ۳

۱) Method

۲) field

۳) class

۴) object

۵) class

۶) java

۷) javac

۸) .java

۹) .class

۱۰) Bytecodes

امیرحسین ایروانی‌منش 9931066 airavanimanesh@aut.ac.ir

تکلیف شماره ۱

تمرین شماره ۴

اولین ایراد کد در خط دوم است که به متد printArray به اشتباه int ورودی تنظیم شده در صورتی که باید int[] arr باشد.

ایراد بعدی در خط چهارم است که در for باید i<n باشد، اگر علامت کوچکتر مساوی باشد از سایز آرایه بیشتر پیمایش می شود و در نتیجه خطای رانتایم ایجاد می شود.

در خط ۸، در قسمت ورودی متد، آرایه به شکل دستور زبان C به عنوان ورودی ذکر شده که اشتباه است. شکل درست آن: int[] arr.

در خط ۱۲، در قسمت گزاره شرطی for باید gap > 0 باشد چون بعد از تقسیم های پیاپی gap صفر شده و شرط همیشه برقرار است، در همان خط قسمت سوم for نیز باید gap /= 2 باشد چون تقسیم بر یک بی معنی است و الگوریتم سورت به هم میریزد.

در خط ۱۸ ام باید به جای arr[i + gap] ، arr[i – gap] نوشته شود این قسمت علاوه بر ایراد الگوریتمی ارور out of bound به دلیل پیمایش بیشتر از سایز آرایه را ایجاد میکند.

در خط ۲۵ نیز دوباره ورودی متد main به شکل دستور زبان C فراخوانی شده و باید String[] args باشد.

این اشکال در خط ۲۶ نیز تکرار شده و باید به صورت int[] arr باشد.

شکل کامل و درست کد:

```
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);
    }
}
```