



«آیا انسان نمی‌داند که خداوند او را می‌بیند»

- ۱) صحیح یا غلط بودن هر گزاره را (بدون ذکر دلیل) مشخص کنید. (۲۰) (هر جواب نادرست، نیمی از نمره‌ی یک جواب صحیح را از بین می‌برد)
- الف. اگر متغیری با نام `person` از کلاس `Person` به شکل `final` تعریف شده باشد، آنگاه دستور `person.setName();` الزاماً منجر به تولید خطا می‌شود.
- ب. برنامه‌ای که به زبان جاوا در لینوکس نوشته و اجرا شده است، روی ویندوز هم قابل اجراست، به شرطی که یک بار برای ویندوز هم کامپایل شود.
- ج. به تعریف چند متد هم نام درون یک کلاس، چندریختی (`Polymorphism`) گفته می‌شود.
- د. کلاس `Cat` و پدر آن، `Animal` را در نظر بگیرید. دستور `Cat c = new Animal();` یک `DownCasting` (تغییر نوع به پایین) را به درستی اجرا می‌کند.
- ه. از آنجایی که جاوا از وراثت چندگانه پشتیبانی نمی‌کند، هر واسط (`interface`) جاوا حداکثر می‌تواند از یک واسط جاوا ارث‌بری کند.
- و. در مدیریت استثنائات، همواره نوشتن یک منطق خاص در بلاک `finally` معادل نوشتن آن منطق پس از ساختار `try-catch` و بدون `finally` است.
- ز. در برنامه‌های جاوا، اشیاء در مکانی از حافظه به نام `Heap` ذخیره می‌شوند.
- ح. پیاده‌سازی کلاس `HashSet` با استفاده از کلاس `HashMap` انجام شده است.
- ط. کلاس `LinkedList` به منظور مقایسه‌ی اشیاء از متد `compareTo` شیء داده شده استفاده می‌کند.
- ی. نمی‌توان انواع داده‌ی اولیه مانند `int` را به عنوان جنس عناصر `ArrayList` در نظر گرفت.
- ۲) ساختار درونی `HashSet` و الگوریتم افزوده شدن یک عضو جدید به آن، `add(E e)`، را به واضح‌ترین شکل ممکن توضیح دهید. (۲۰) (پاسخ در همین صفحه)

۳) یکی از اصول مهندسی نرم افزار OCP (Open Closed Principle) است. براساس این اصل، بایستی بتوان ویژگی‌های جدیدی به نرم افزار اضافه کرد، به نحوی که کمترین تغییر بر روی بخش های کنونی کد اعمال شود. توضیح دهید که چرا در نمونه کد زیر این اصل نادیده گرفته شده است؛ سپس حالت بهبود یافته‌ی آن را بنویسید. (۲۰) (پاسخ در همین صفحه)

```
Class Parking {
    int capacity;
    List<Pride> prides;
    List<Peykan> peykans;
    List<Bmw> bmws;

    // methods

    public int getFreeSpaces() {
        return capacity – prides.size() – peykans.size() – bmws.size();
    }
}
```

۴) یک نرم افزار مدیریت پارکینگ را در نظر بگیرید. کلاس پارکینگ یک کلاس Singleton است. پارکینگ دارای ظرفیتی است که در ابتدای کار مشخص می شود و قابل تغییر نیست. هر جایگاه درون پارکینگ با یک کد رشته ای مشخص می شود. مسئول پارکینگ با ورود هر اتومبیل به درون پارکینگ، متد ورود پارکینگ را صدا کرده و شئی ای از کلاس اتومبیل و کد جایگاه را به آن می دهد. شماره ی انتظامی خودرو (پلاک) و نام خانوادگی راننده در سیستم ثبت می شود. بدیهی است که با خروج هر اتومبیل، متد خروج پارکینگ صدا زده می شود تا جایگاه مربوطه خالی شود. در هر لحظه بایستی مشخص باشد که آیا یک جایگاه خالی است یا اتومبیلی درون آن پارک شده است و بتوان مشخصات اتومبیل مربوطه را درخواست کرد. ساعت ورود و خروج هر اتومبیل بایستی در سیستم ثبت شود و هیچ گاه پاک نمی شود (برای گزارشات بعدی). بایستی متدی برای محاسبه هزینه در سیستم در نظر گرفته شود که اختلاف ساعت ورود و خروج را در هزینه یک ساعت ضرب کرده و نتیجه را بر می گرداند. هزینه یک ساعت بایستی در جای مناسبی از سیستم ثبت می شود و قابل ویرایش است. توجه داشته باشید که در آینده بایستی بتوان هزینه ی یک ساعت و هزینه ی کل اتومبیلی را که قبلاً در پارکینگ بوده است از سیستم درخواست کرد. با فرض پر بودن ظرفیت پارکینگ، متد ورود بایستی استثنائی با عنوان FullException را برگرداند که در متد اصلی برنامه مدیریت می شود. کامپایلر نباید مدیریت FullException را بررسی نماید. براساس توضیحات داده شده، کلاس ها، ویژگی ها و متدهای لازم را پیاده سازی نمایید (نیازی به پیاده سازی متد main برنامه نیست). (۴۰)

پیروز باشید

مصطفی زاده