



برنامه نویسی پیشرفته

زمستان و بهار ۱۳۹۸-۹۹ - دانشکده علوم ریاضی

دانشگاه صنعتی شریف

با توجه به شرایط خاص پیش آمده تیم درس برنامه نویسی پیشرفته تصمیم گرفتند که یک پرسش‌نامه بدون تاثیر در ارزیابی برای بررسی میزان پیشرفت مطالعه برگزار کند. هدف از این پرسش‌نامه بررسی پیشرفت عملکرد و مطالعه شما است. هدف از این پرسش‌نامه این موارد است:

- جبران فاصله ایجاد شده میان دانشجویان با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می‌کند.
- جبران فاصله ایجاد شده میان دانشجویان و تیم درس برای دریافت بازخورد پیشرفت تحصیلی
- بازخورد هر دانشجو به خود در رابطه با پیشرفت مناسب در مطالعه و یادگیری مفاهیم از طریق منابع درس
- جهت‌دهی به اشکالاتی که شما ممکن هست هنوز در جریان وجود نقطه ضعف خود در این رابطه نباشید و رفع آنها در جلسه‌های رفع اشکال آنلاین
- دقت کنید که این پرسش‌نامه تنها مرجع برای بازخورد پیشرفت مناسب شما در مطالعه منابع نیست. تمرین‌ها و پروژه نیز سهم بزرگی در این مساله دارند. پس حتما پس از پاسخ به سوال‌ها و ارسال پاسخ‌ها، اشکال‌ها و ابهام‌هایی که داشتید در جلسه‌های آنلاین رفع اشکال در میان بگذارید و رفع کنید و از این فرصت استفاده کنید.

توضیحات

- نتیجه این پرسش‌نامه تاثیری در ارزیابی نهایی این درس ندارد.
- این پرسش‌نامه برای اطمینان بیشتر از اینکه مسیر درس را درست طی می‌کنید طراحی شده.
- اگر نیاز به بررسی صحت پیشرفتتان دارید حتما در این پرسش‌نامه شرکت کنید.
- در صورتی که با مطالب درس به درستی پیش آمده باشید می‌توانید به تمام سوال‌ها پاسخ دهید.
- در صورتی که به بخشی از هر سوال تسلط ندارید یا احتیاج به بررسی صحت پاسخ‌ها دارید حتما در جلسه‌های رفع اشکال شرکت کنید و اشکال یا ابهام‌های خود را رفع کنید.
- سعی کنید جواب‌ها کوتاه و دقیق باشند که مرور جواب در جلسه رفع اشکال سریع‌تر انجام شود.
- از آنجایی که این پرسش‌نامه برای یادگیری طراحی شده می‌توانید در پر کردن سوال‌ها با هر فردی مشورت و همفکری کنید.

نحوه انجام پرسش‌نامه

- برای پاسخ به این پرسش‌نامه یک نسخه از این فایل را از منو فایل و گزینه گرفتن یک کپی برای خود ایجاد کنید و جواب‌های آن را در همین فایل بنویسید.
- پس از جواب دادن به سوال‌ها آن را در قالب PDF دانلود کنید.
- فایل PDF در یک ریپازیتوری github بارگذاری کنید.
- آدرس این ریپازیتوری را در یک فایل یک خطی با پسوند جاوا داخل کوئرا و در بخش پرسش‌نامه بررسی پیشرفت بارگذاری کنید.

سوال‌ها

سوال ۱

خروجی این برنامه را بدست بیاورید و به ازای هر خط توضیح دهید که چرا به این خروجی رسید؟

```
class Classes {
    static class A {
        static int intValue = 0;
        int integerValue = 20;

        A() {
            integerValue = 5;
            printValue();
            print();
        }

        void printCaller() {
            print();
        }

        void printValue() {
            System.out.println("B:" + integerValue);
        }

        void print() {
            System.out.println("A:" + intValue);
        }
    }

    static class B extends A {
        B(int v) {
            intValue = v;
            integerValue = 15;
            printValue();
            print();
        }

        void print() {
            System.out.println("B:" + intValue);
        }

        void printSuper() {
            super.print();
        }

        void printCaller() {
            printValue();
            super.printValue();
        }

        void printValue() {
            System.out.println("B:" + integerValue);
            super.printValue();
        }
    }

    static public class C extends A {
        void printCaller() {
            System.out.println("B:" + integerValue);
        }
    }
}
```

```

    }

    void print() {
        System.out.println("A:" + intValue);
        super.printCaller();
    }
}

class Problem1 {
    public static void incrementValue(Classes.A object) {
        object.intValue++;
        object.integerValue++;
    }

    public static void incrementValue(int firstValue, int secondValue) {
        firstValue++;
        secondValue++;
    }

    public static void main(String[] args) {
        Classes.A a = new Classes.A();
        Classes.B b = new Classes.B(10);
        Classes.A c = b;

        b.print();
        c.print();
        ((Classes.A) b).print();
        b.printSuper();
        a.printCaller();
        b.printCaller();
        c.printCaller();
        incrementValue(a);
        a.printCaller();
        incrementValue(b);
        b.printCaller();
        incrementValue(c);
        c.printCaller();
        incrementValue(b.intValue, b.integerValue);
        b.printCaller();
        c.printCaller();
    }
}

```

پاسخ :

B:5
 A:0
 B:5
 B:5
 B:0
 B:15
 B:15
 B:10
 B:10
 B:10

B:10
A:10
A:10
B:15
B:15
B:15
B:15
B:15
B:15
A:11
B:16
B:16
B:16
B:17
B:17
B:17
B:17
B:17
B:17
B:17
B:17
B:17

سوال ۲

توضیح دهید که هدف از ارث بری در شی گراپی چیست. چه زمان از composition و چه زمان از inheritance استفاده می کنیم؟ چگونه می توانیم از سازنده پدر را فراخوانی کنیم؟ چگونه می توانیم سازنده دیگری از خود کلاس را فراخوانی کنیم؟

پاسخ :

هدف از استفاده ارث بری جلوگیری از کپی کردن کد است و مانع از این امر میشود. زمانی از composition استفاده میکنیم که رابطه has a بین دو کلاس وجود داشته باشد و هر وقت رابطه is a بود از inheritance استفاده میکنیم. برای مثال بین حیوان و سگ رابطه is a است ولی رابطه دست و آدم رابطه has a است. اگر سازنده کلاس پدر بدون پارامتر باشد کانستراکتر به صورت خودکتر اجرا میشود در غیر این صورت باید از super استفاده کنیم. اگر بخواهیم که در کلاس مان کانستراکتری استفاده کنیم در کانستراکتر کلاس پدر در خط اول ابتدا با کلمه super کانستراکتر کلاس پدر را فرا میخوانیم سپس باقی کد خود را می زنیم.

سوال ۳

توضیح دهید که چرا از رابطها (interface) استفاده می کنیم. چه محدودیتهایی نسبت به یک کلاس دارند و چرا امکان پیاده سازی متد در آنها داده شده است؟

پاسخ :

اینترفیس کاری میکند تا یک سری رفتار مشخص را بین کلاس هایی که آن را پیاده سازی می کنند به اشتراک بگذارند و این کار باعث کوتاه تر شدن کد میشود. محدودیت آنها نسبت به کلاس این است که ما اجازه تعریف عملکرد متد در آن را نداریم و همچنین نمی توانیم شی از آن بسازیم و در واقع به این دلیل امکان پیاده سازی متد در آنها داده شده که ویژگی و رفتارهای کلاس های پدر را در آن تعریف کنیم.

سوال ۴

کلاس انتزاعی (abstract) چیست و چه زمانی در مدل سازی از یک کلاس انتزاعی استفاده می کنیم؟ این نوع کلاس چه تفاوتی با رابط (interface) دارد؟

پاسخ :

وقتی کد را از کاربر مخفی نگه میداریم و فقط ویژگی ها را برای کاربر قرار میدهیم در واقع از کلاس abstract استفاده میکنم. در واقع باعث میشود بجای نگاه کردن به کد و اینکه کد چگونه عملیات را انجام می دهد، بر روی خود عملیات تمرکز کنیم.

فرق interface و abstract در این است که کل متد ها در interface بصورت خودکار abstract تعریف میشوند ولی در کلاس abstract میتوان از نوع های دیگری نیز استفاده کرد

سوال ۵

override کردن تابع و متغیر چه تاثیری در عملکرد متد در یک کلاس فرزند می گذارد؟ چگونه می توانیم پس از override شدن یک متد در کلاس فرزند در هر کدام از مکان های زیر به نسخه هم نام آن متد در کلاس پدر دسترسی پیدا کنیم؟

- متدی داخل کلاس پدر
- متدی داخل کلاس فرزند
- خارج از دو کلاس

پاسخ :

وقتی متدی را override میکنیم متد در کلاس فرزند تغییر میکند. پس از override شدن متد یا متغیر برای دسترسی به متد پدر از کلمه super استفاده می کنیم و در داخل کلاس از this استفاده میکنیم و اگر خارج دو کلاس باشد باید با ساخت شی از آن کلاس آن متد را فراخوانی کرد.

سوال ۶

توضیح دهید که منظور از چندریختی در شی گرایی چیست و چه مزیتی ایجاد می کند.

پاسخ :

منظور قابلیت و توانایی آن تابع برای انجام کارهای مختلف بر اساس آبجکتی است که به آن متعلق است. برای مثال ویژگی مساحت را همه ی شکل ها دارند ولی تعریف مساحت هر شکل متفاوت است. مزیت آن نیز آن است که به ما این امکان را میدهد که از یک متد، با ورودی های مختلف استفاده کنیم.

سوال ۷

چرا از توابع و متدها در زبان برنامه نویسی استفاده می کنیم؟ در طراحی برنامه و شکستن آن به توابع و متدهای مختلف چه نکته هایی را باید رعایت کرد که خوانایی آن بیشتر شود و پیچیدگی اضافی نداشته باشیم؟

پاسخ :

زیرا استفاده از توابع و متد ها باعث سادگی برنامه میشود به این صورت که هر تابع قسمتی از برنامه را در دست میگیرد و مسئله را ساده تر میکند. همچنین برای خوانایی و سادگی فهمیدن کد بهتر است از نام های مرتبط با تابع و متد استفاده کنیم و همچنین برای هر بخش از کد کامنت بزاریم.

سوال ۸

کلاس درونی (inner class) چه انواعی دارد و هر کدام چه کاربردی در مدل سازی و توصیف موجودات دارد؟ چگونه می توانیم یک شی از هر نوع ایجاد کنیم؟ در صورت override شدن یک متد یا متغیر توسط یک کلاس درونی چگونه می توان به نسخه override شده از کلاس بیرونی دسترسی پیدا کرد؟

پاسخ :

کلاس داخلی 3 نوع دارد کلاس داخلی غیراستاتیک ، کلاس داخلی استاتیک و کلاس داخلی بی نام. برای ایجاد شی از کلاس داخلی باید یک شی در داخل همان کلاس ایجاد شود و در یک متد آن شی استفاده شود. برای دسترسی به متد override شده ی آن باید نام کلاس بیرونی را بنویسیم و سپس با کلمه ی this. به آن ارجاع دهیم.

سوال ۹

کلمه کلیدی final روی هر کدام از موارد زیر چه تاثیری دارد؟

- تابع و متد
- تعریف کلاس
- یک متغیر از نوع شی
- یک متغیر از نوع پایه

پاسخ :

اگر در تعریف یک تابع از کلمه final استفاده کنیم در واقع کاری کردیم که نتوان آن را override کرد و همچنین در کلاس ها به این معنی است که نمیتوان از آن ارث بری کرد و در متغیر های شی به این معناست که نمی توان آن متغیر را به شی دیگری نسبت داد و همچنین در متغیر ها به این معنی است که اندازه آن ثابت است و غیر قابل تغییر است.

سوال ۱۰

کلمه کلیدی static روی هر کدام از موارد زیر چه تاثیری دارد؟

- تابع و متد
- تعریف کلاس
- یک متغیر از نوع شی
- یک متغیر از نوع پایه

پاسخ :

وقتی یک متغیر را استاتیک تعریف میکنیم در واقع داریم به آن متغیر یک حافظه اختصاص میدهیم که مخصوص شی نیست و به کلاس اختصاص دارد. زمانی که یک متد استاتیک است برای استفاده از آن نیازی به ساخت شی نداریم و متد های استاتیک متعلق به کلاس هستند و برای همه اشیا یکسان است. اگر یک کلاس استاتیک هست نیازی برای تعریف شی از آن کلاس نیست و میتوان از آن کلاس استفاده کرد همچنین همه توابع و متد ها و متغیر های آن باید استاتیک باشند.