



برنامه نویسی پیشرفته

زمستان و بهار ۹۹-۱۳۹۸ - 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 دانشجو با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می‌کند.
- جبران فاصله ایجاد شده میان دانشجو ها 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 کونرا و در بخش پرسشنامه بررسی پیشرفت بارگذاری کنید.

سوال‌ها

حسین رحمانی

98110157

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();
    }
}

```

```

} in constructor a B:5
In constructor a A:0
We call super constructor
B:5 when we call method although we are in constructor A()overriding of
this method in B() will be called
B:5
B:0 until this line our answer is the same.
Now we are in constructor b
B:15
B:15
B:10
Now our commands start from b.print();
B:10 set intValue10
B:10 b and c are the same because they refer same object
B:10 although it cast to A but naturally it is a B object (its address)
A:10 super method of this method in class A
*
A:10 Printcaller call method print in A

1 B:15 all 3 of them in b.printcaller call printvalue
2B:15 super.printvalue and again it calls
3B:15

B:15 exact work like b
B:15
B:15
A:11 because increment method now int value is 11.
B:16 i explain before these three lines print caller on b one time print
value
B:16 two time its super method
B:16

B:17
B:17 c is exactly like b.
B:17
With one increment method now integervalue is 17.
B:17
B:17 explain before.
B:17

B:17
B:17 exactly same as b
B:17

```

سوال ۲

توضیح دهید که هدف از ارث بری در شی گرای چيست. چه زمان از composition و چه زمان از inheritance استفاده می‌کنیم؟ چگونه می‌توانیم از سازنده پدر را فراخوانی کنیم؟ چگونه می‌توانیم سازنده دیگری از خود کلاس را فراخوانی کنیم؟
جواب: مثلا با این کار با حفظ ویژگی های قبل ویژگی جدید میپذیرد. مثلا برای نوشتن سگ می‌توانیم آن را از کلاس animal extend کرد. (هدف: کوتاه نویسی و عدم تکرار کد) وقتی بین اشیا رابطه a - s برقرار باشد بهتر است از inheritance استفاده کنیم. و در غیر این صورت از composition استفاده کنیم. با کلیدواژه super در اولین خط constructor پس (البته با کلید سمت راست alt به صورت پیشفرض constructor میسازد.) با دادن پارامتر های ورودی مناسب هر کانستراکتوری که وجود داشته باشد فراخوانی میشود.

سوال ۳

توضیح دهید که چرا از رابطها (interface) استفاده می‌کنیم. چه محدودیت‌هایی نسبت به یک کلاس دارند و چرا امکان پیاده‌سازی متد در آنها داده شده است؟ **جواب:** این آپشن (interface) برای رفع محدودیت اینک یک کلاس فقط میتواند یک کلاس دیگر را extends کرد. بوجود آمده و همه کلاس هایی که یک ویژگی مشترک دارند آن را implements کنند. توانایی ساخت شی ندارند. از نسخه 8 به بعد اجازه این کار داده شد و استفاده آن هارا بیشتر کرد.

سوال ۴

کلاس انتزاعی (abstract) چیست و چه زمانی در مدل‌سازی از یک کلاس انتزاعی استفاده می‌کنیم؟ این نوع کلاس چه تفاوتی با رابط (interface) دارد؟ **جواب:** کلاسی که نتوان از آن شی ساخت (حالا میخواهد تابع abstract داشته باشد یا نه صرفا تعریف شده باشد). زمانی که همه اشیا یک ویژگی را دارند ولی نحوه پیاده سازی فرق میکند. همه متد یک متد را دارند ولی نحوه پیاده سازی ها متفاوت است. در کل می‌شود گفت interface یک کلاس abstract است که تمام توابع غیر استاتیک آن abstract هستند. و مثلا خودش نمیتواند ارث بری کند. ولی abstract کلاس میتواند.

سوال ۵

override کردن تابع و متغیر چه تأثیری در عملکرد متد در یک کلاس فرزند می‌گذارد؟ چگونه می‌توانیم پس از override شدن یک متد در کلاس فرزند در هر کدام از مکان‌های زیر به نسخه هم نام آن متد در کلاس پدر دسترسی پیدا کنیم؟

- متدی داخل کلاس پدر this. اسم متد
 - متدی داخل کلاس فرزند با کلید واژه super
 - خارج از دو کلاس با نوشتن اسم کلاس + "." + اسم متد
- جواب:** دیگر کار در کلاس پدر را انجام نمیدهد. و مفهومش عوض میشود.

سوال ۶

توضیح دهید که منظور از چندریختی در شی گرای چيست و چه مزیتی ایجاد می‌کند. دو نوع است یکی override کردن تابع است تعریف تابع abstract در کلاس پدر و نوشتن آن در کلاس فرزند نمونه polymorphism است و باعث فهمیدن بهتر کد میشود. یکی از ابزار های مهم شی گرای است. و بعدی مثلا می‌توانیم دو تابع با اسم های برابر ولی مثلا پارامتر های متفاوت به اصطلاح overload

سوال ۷

چرا از توابع و متدها در زبان برنامه نویسی استفاده می‌کنیم؟ در طراحی برنامه و شکستن آن به توابع و متدهای مختلف چه نکته‌هایی را باید رعایت کرد که خوانایی آن بیشتر شود و پیچیدگی اضافی نداشته باشیم؟ **جواب:** برنامه نویسی خوانا تر و Debug کد راحت تر میشود و در صورت غلط بودن عملکرد میتوان بقیه کد را زد و بعدا ان قسمت را درست کرد. اسم مناسب داشته باشند و مثلا در این کار زیاده روی نداشته باشیم.و الکی کار را سخت نکنیم.

سوال ۸

کلاس درونی (inner class) چه انواعی دارد و هر کدام چه کاربردی در مدل‌سازی و توصیف موجودات دارد؟ چگونه می‌توانیم یک شی از هر نوع ایجاد کنیم؟ در صورت override شدن یک متد یا متغیر توسط یک کلاس درونی چگونه می‌توان به نسخه override شده از کلاس بیرونی دسترسی پیدا کرد؟ دو نوع استاتیک و غیر استاتیک تا اونجایی که من پرسیدم مثلا ما به یک کلاس داخل یک کلاس دیگر نیاز مندیم فقط پس انرا همانجا تعریف میکنیم. قسمت override رو نمیدونم.

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

جواب: تابع یا متد final قابل override کردن در زیر کلاس ها نیست.و کلاس final هم کلاسی هست که هیچ زیر کلاسی نداره. یعنی هیچ کلاسی نمیتواند آن را extend کند.شی final هم یعنی pointer ان ثابت باشد.یعنی ادرسی که نگه میدارد از اوجکت فیکس باشد این اینستنس فقط به این اوجکت اشاره کند.متغیر پایه هم یعنی مقدار ان تغییر نمیکند.

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

جواب: متغیری پایه که با static تعریف شده باشد، می تواند بدون ساخته شدن نمونه از کلاس میزبان فراخوانی شود و بین تمامی نمونه های کلاس مشترک می باشد.بین تمام object های ساخته شده. یک متد static به کل کلاس میزبان تعلق دارد و مستقل از نمونه های ساخته شده از روی کلاس می باشد.یک متد static می تواند بدون نیاز به ایجاد نمونه از روی کلاس فراخوانی شود.یک متد static می تواند به عضوهای داده ای static دسترسی داشته و مقدار آن ها را دستکاری کند.ولی امکان دسترسی به this و super وجود ندارد. چون به اوجکت وابسته نیست.کلاس استاتیک وجود ندارد مگر اینکه inner class باشد.تنها به متد ها و فیلد های استاتیک کلاس پدر خود دسترسی دارد.و بدون شی از کلاس پدر فراخوانی میشود.