



برنامه نویسی پیشرفت

زمستان و بهار ۱۳۹۸-۹۹ - دانشکده علوم ریاضی
دانشگاه صنعتی شریف

با توجه به شرایط خاص پیش آمده تیم درس برنامه نویسی پیشرفت تضمین 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:

- جراین فاصله ایجاد شده میان دانشجوها با هدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می کند.
- جراین فاصله ایجاد شده میان دانشجوها و تیم درس برای دریافت بازخورد پیشرفت تحصیلی
- بازخورد هر دانشجو به خود در رابطه با پیشرفت مناسب در مطالعه و یادگیری مقایمه از طریق منابع درس
- چهتدهی به اشکالاتی که شما ممکن هست هنوز در جریان وجود نقطه ضعف خود در این رابطه نباشد و رفع آنها در جلسه های رفع اشکال آنلاین
- دقت کنید که این پرسشنامه تنها مرجع برای بازخورد پیشرفت مناسب شما در مطالعه منابع نیست. تمرین ها و پروژه نیز سهم بزرگی در این مساله دارند. پس حتماً پس از پاسخ به سوال ها و ارسال پاسخ ها، اشکال ها و ابهام هایی که داشتید در جلسه های 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 می کنید طراحی شده.
- اگر نیاز به بررسی صحت پیشرفتان دارید حتماً در این پرسشنامه شرکت کنید.
- در صورتی که با مطالب درس به درستی پیش آمده باشید می توانید به تمام سوال ها پاسخ دهید.
- در صورتی که به بخشی از هر سوال تسلط ندارید یا احتیاج به بررسی صحت پاسخ ها دارید حتماً در جلسه های رفع اشکال شرکت کنید و اشکال یا ابهام های خود را رفع کنید.
- سعی کنید جواب ها کوتاه و دقیق باشند که مرور جواب در جلسه رفع اشکال سریعتر انجام شود.
- از انجایی که این پرسشنامه برای یادگیری طراحی شده می توانید در پر کردن سوال ها 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 جواب های آن را در همین فایل بنویسید.
- پس از جواب دادن به سوال ها 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 کوئری و در بخش پرسشنامه بررسی پیشرفت بارگذاری کنید.

سوال‌ها

98110157

حسین رحمانی

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 دهد که هدف از ارث بری در شی گرایی چیست. 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: مثلاً با این کار با حفظ ویژگی های قبل ویژگی جدید می‌بینیم. مثلاً برای نوشتن سگ میتوانیم ان را از کلاس animal extend کرد. (هدف: کوتاه نویسی و عدم تکرار کد) وقتی بین اشیارابطه a- ابرقرار باشد بهتر است از inheritance استفاده کنیم و در غیر این صورت از composition استفاده کنیم. با کلیدوازه super در اولین خط constructor پسر (البته با کلید سمت راست alt به صورت پیشفرض constructor می‌سازد). با ادان پارامتر های ورودی مناسب هر کانسٹراکتوری که وجود داشته باشد فراخوانی می‌شود.

سوال ۳

توضیح دهد که چرا از رابطه (interface) استفاده می‌کنیم. چه محدودیت‌هایی نسبت به یک کلاس دارند و چرا امکان پیاده‌سازی متدهای آنها داده شده است؟ **جواب:** این آپشن (interface) برای رفع محدودیت اینکه یک کلاس فقط میتواند یک کلاس دیگر را extends کرد. بوجود امده و همه کلاس هایی که یک ویژگی مشترک دارند آن را implements کنند. توافقی ساخت شی ندارند. از نسخه 8 به بعد اجازه این کار داده شد و استفاده ان ها را 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 صرفاً تعریف شده باشد). زمانی که همه اشیا یک ویژگی را دارند ولی نحوه پیاده سازی فرق می‌کند. همه متدهای دارند ولی نحوه پیاده سازی ها متفاوت است. در کل می‌شود گفت یک کلاس abstract است که تمام توابع غیر استاتیک آن abstract هستند و مثلاً خودش نمیتواند ارث بری کند. ولی abstract کلاس میتواند.

سوال ۵

کردن تابع و متغیر چه تاثیری در عملکرد متدهای فرزند است؟ چطور می‌توانیم پس از override شدن یک متدهای فرزند در هر کدام از مکان‌های زیر به نسخه هم نام آن متدهای فرزند می‌گذاریم؟

• متدهای داخل کلاس پدر this. اسم متدهای داخل کلاس فرزند با کلید واژه super

• خارج از دو کلاس با نوشتن اسم کلاس + ".+" + اسم متدهای کار در کلاس پدر را انجام نمیدهد و مفهومش عوض می‌شود.

سوال ۶

توضیح دهد که منظور از چندربختی در شی گرایی چیست و چه مزیتی ایجاد می‌کند. دو نوع است یکی **override** کردن تابع است تعريف تابع abstract در کلاس پدر و نوشتن آن در کلاس فرزند نمونه polymorphism است و باعث فهمیدن بهتر کد می‌شود. و یکی از ابزارهای مهم شی گرایی است. و بعدی مثلاً میتوانیم دو تابع با اسم های برابر ولی مثلاً 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 نکتهایی را باید رعایت کرد که خوانایی آن بیشتر شود و پیچیدگی اضافی نداشته باشیم؟ **جواب:** برنامه نویسی خواناًتر و Debug راحت‌تر می‌شود و در صورت غلط بودن عملکرد می‌توان بقیه کرزاً و بعد از قسمت را درست کرد. اسم مناسب داشته باشند و مثلاً 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 متدها می‌توان به نسخه override شده از کلاس بپردازی دسترسی پیدا کرد؟ دو نوع استاتیک و غیر استاتیک تا اونجایی که من پرسیدم مثلاً ما به یک کلاس داخل یک کلاس دیگر نیازمندی فقط پس از همانجا تعریف می‌کنیم. قسمت override رو نمیدومن.

سوال ۹

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

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

جواب: تابع یا متدها final قابل override کردن در زیر کلاس‌ها 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 این ثابت باشد. یعنی ادرسی که نگه میدارد از ابجکت فیکس باشد این اینستنس فقط به این ابجکت اشاره کند. متغیر پایه هم یعنی مقادیر این تغییر نمی‌کند.

سوال ۱۰

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

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

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