



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 کنند. هدف از این پرسشنامه بررسی پیشرفت عملکرد و مطالعه شما است. هدف از این پرسشنامه این موارد است:

- جبران فاصله ایجاد شده میان دانشجویها با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می کند.
- جبران فاصله ایجاد شده میان دانشجویها 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
- جهت دهی به اشکالاتی که شما ممکن هست هنوز در جریان وجود نقطه ضعف خود در این رابطه نباشید 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 حتما 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 پیشرفتتان دارید حتما 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 حتما 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 سریع تر انجام شود.
- از آنجایی 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 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(); //B:5    printValue in A constructor
                                         //A:0    print in A constructor
        Classes.B b = new Classes.B(10); //B:5    printValue for class B in A
        constructor
                                         //B:5    super.printvalue in B
        printvalue
                                         //B:0    print for class B
                                         //B:15    printvalue in B constructor
                                         //B:15
                                         //B:10

        Classes.A c = b;

        b.print(); //B:10
        c.print(); //B:10
        ((Classes.A) b).print();
        b.printSuper(); //A:10
        a.printCaller(); //A:10
        b.printCaller(); //B:15
                           //B:15
                           //B:15
        c.printCaller(); //B:15
                           //B:15

```

```

//B:15
incrementValue(a);
a.printCaller();//A:11
incrementValue(b);
b.printCaller();
incrementValue(c);
c.printCaller();
incrementValue(b.intValue, b.integerValue);
b.printCaller();
c.printCaller();
}
}

```

سوال ۲

توضیح دهید که هدف از ارث بری در شی گرایی 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؟ هدف از ارث بری code reuse و استفاده مجدد است تا متد ها و فیلد هایی که تعریف شده اند برای یک کلاس و در کلاس دیگری تمام این رفتار ها و وضعیت ها وجود دارد دوباره نوشته نشوند البته به شرط آن که رابطه is a بین آن ها برقرار باشد. زمانی که رابطه is a بین کلاس ها برقرار باشد از inheritance و اگر رابطه has a برقرار باشد از composition استفاده می کنیم. با کلید وازه super سازنده پدر و با کلید وازه this سازنده از خود کلاس را فراخوانی می کنیم.

سوال ۳

توضیح دهید که چرا از رابط ها (interface) استفاده می کنیم. چه محدودیت هایی نسبت به یک کلاس دارند و چرا امکان پیاده سازی متد در آنها داده شده است؟ استفاده از آن ها ساده تر است و یک کلاس میتواند از چند interface ارث بری کند ولی معمولا در آن ها فیلدی تعریف نمیشود. متد ها در آن abstract هستند و باید در زیر کلاس ها پیاده سازی شوند دلیل وجود متد در آن ها استفاده از امکان چند ریختی ست.

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؟ زمانی که کلاسی حداقل یک فیلد abstract داشته باشد abstract می شود زیرا شیء سازی از آن غیر ممکن می شود. تفاوت آن با abstract این است که متود غیر انتزاعی می تواند داشته باشد و تعریف کردن فیلد نیز در آن معمول است.

سوال ۵

override کردن تابع و متغیر چه تاثیری در عملکرد متد در یک کلاس فرزند می گذارد؟ چطور 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: با کلید وازه `super.method`
 - متدی داخل کلاس فرزند: با کلید وازه `super.method`
 - خارج از دو کلاس: با ساختن شیء ای از جنس کلاس پدر و فراخوانی متد.
- زمانی که یک شی از جنس فرزند ساخته می شود متد override شده فراخوانی می شود.

سوال ۶

توضیح دهید که منظور از چندریختی در شی گرایی چیست و چه مزیتی ایجاد می کند. زمانی که چند کلاس از یک کلاس پدر ارث بری می کنند و نمی دانیم که دقیقا از کدام کلاس می خواهیم استفاده کنیم یک شیء از جنس کلاس پدر می سازیم و در وقت مناسب (مثلا با ایجاد یک if) شیء ساخته شده را new می کنیم. که در کوتاه شدن برنامه ها بسیار مؤثر است.

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؟ زمانی که چند بار یک قطعه کد با بدنه یکسان ولی پارامترهای متفاوت داریم با ایجاد متد میتوان برنامه را خوانا تر و خلاصه تر کرد. متودها باید در کلاس های مختلف که شبیه همن نگه داری شوند. نام گذاری مناسب باید داشته باشند.

سوال ۸

کلاس درونی (inner class) چه انوعی 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؟

سوال ۹

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

- تابع و متد: امکان override کردن آن در زیر کلاس وجود ندارد.
- تعریف کلاس: امکان ارث بری از آن وجود ندارد.
- یک متغیر از نوع شی: امکان تغییر رفرنس آن وجود ندارد.
- یک متغیر از نوع پایه: امکان تغییر مقدار آن وجود ندارد.

سوال ۱۰

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

- تابع و متد: برای کل کلاس است نه روی شیء خاصی و رفرنس به کلاس وجود ندارد. (به this دسترسی ندارد)
- تعریف کلاس: مربوط به کلاس داخلی است که رفرنس به کلاس خارجی وجود ندارد. (به outer.this دسترسی ندارد)
- یک متغیر از نوع شی: برای کل کلاس است نه روی شیء خاصی.
- یک متغیر از نوع پایه: برای کل کلاس است نه روی شیء خاصی.