



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 دانشجویان با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می کند.
- 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
- 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();
        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();
    }
}

```

```

B:5  constructor A - taghire integervalue -printvalue
A:0  constructor A- print -
B:5  constructor B- constructor A -yaghir integervalue-printvalue
B:0  constructor B-constructor A-print
B:15 constructor B-taghir integervalue- printvalue
B:15 constructor B-taghir integervalue-super.printvalue
B:10 constructor B-taghir intvalue- print
B:10 method print  dar B
B:10 methos print dar B
B:10 method print dar B
A:10 printsuper dar B- print dar A
A:10 printceller dar A-print dar A
B:15 print celler dar B-printvalue dar B-printvalue dar A
B:15  print celler dar B- printvalue dar A
B:15  printceller dar B-print value dar A
B:15  print celler dar B- printvalue dar B
B:15  printceller dar B-printvalue dar A
B:15 print celler dar B-print value dar A
A:11 printceller dar A-print dar A
B:16 printceller dar B- print value dar B
B:16 printceller dar B- print value dar A
B:16 print celler dar B-print value dar A
B:17 pritceller dar B-printvalue dar B
B:17 printceller dar B- printvalue dar B-print value dar A
B:17 printceller dar B- print value dar A
B:17 printceller dar B- print value dar B
B:17 printceller dar B- print value dar B -printvalue dar A
B:17 printceller dar B- print value dar A
B:17 printceller dar B-printvalue dar B
B:17 printceller dar B-printvalue dar A
B:17 printceller dar B-printvalue dar A

```

سوال ۲

توضیح دهید که هدف از ارث بری در شی گراپی چیست؟ صرفه‌جویی در نوشتن کد و صحت کد
 چه زمان از composition و چه زمان از inheritance استفاده می‌کنیم؟
 rabete class ma ba class haye digar
 rabeteye has-a bood composition va agar is-a bood inheritence estefade mishavad
 چگونه می‌توانیم از سازنده پدر را فراخوانی کنیم؟ **super()**;
 چگونه می‌توانیم سازنده دیگری از خود کلاس را فراخوانی کنیم؟

سوال ۳

توضیح دهید که چرا از رابط‌ها (interface) استفاده می‌کنیم؟
اگر ما توابع را، بدون تعریف کردن آنها در اینترفیس، در کلاس‌ها پیاده‌سازی کنیم، دسترسی به این توابع از طریق اینترفیس امکان‌پذیر نخواهد بود و عملاً ما از چند ریختی محروم خواهیم ماند.

چه محدودیت‌هایی نسبت به یک کلاس دارند و چرا امکان پیاده‌سازی متد در آنها داده شده است؟
نمی‌توانیم تابعی را پیاده‌سازی کنیم
یک کلاس بتواند از اینترفیس‌های مختلف ارث‌بری کند و رفتار مربوط به هر کدام از آن‌ها را داشته باشد.

سوال ۴

کلاس انتزاعی (abstract) چیست و چه زمانی در مدل‌سازی از یک کلاس انتزاعی استفاده می‌کنیم؟

کلاس Abstract، یکی از ابزارهای مهم OOP می‌باشد که نمی‌توان از آنها نمونه‌ای ساخت. به عبارتی دیگر نمی‌توانیم متغیری از کلاس Abstract تعریف کنیم. یک کلاس Abstract شبیه Interface می‌باشد ولی با دیدی وسیع‌تر. این کلاس‌ها می‌تواند دارای متدهای Abstract باشند که شبیه Interface فقط اعلام می‌شوند و باید در کلاسهای مشتق شده بازنویسی شوند. البته می‌توان در این کلاس‌ها متدهایی داشت که Abstract نیستند و احتیاجی به پیاده‌سازی آنها در کلاسهای مشتق شده ندارند.

وقتی در پروژه‌های بزرگ با تغییرات زیادی مواجه هستیم، استفاده از کلاس Abstract توصیه می‌شود؛ چون با تغییر آن به طور خودکار تغییرات در کلاسهای مشتق شده اعمال می‌شوند.
- با توجه به اینکه به غیر از اعلان متدها و خصوصیت‌ها امکان تعریف عناصر دیگری در Interface‌ها وجود ندارد، در صورتیکه ملزم به استفاده از این عناصر باشیم، استفاده از کلاسهای Abstract ضروری می‌باشد.
- در صورتی که نخواهیم کلیه متدها در کلاسهای مشتق شده پیاده‌سازی شوند و تعدادی از آنها را در کلاس پدر کدنویسی کنیم، باید از کلاس Abstract استفاده کنیم.

این نوع کلاس چه تفاوتی با رابط (interface) دارد؟

- 1- یک کلاس معمولی تنها می‌تواند از یک کلاس Abstract ارث‌بری کند ولی همان کلاس می‌تواند از چندین Interface ارث‌برد.
- 2- یک Interface فقط می‌تواند اعلان متدها و خصوصیت‌ها را داشته باشد؛ اما یک کلاس Abstract علاوه بر آنها می‌تواند متدها و خصوصیت‌هایی با کدهای کامل داشته باشد.
- 3- عناصر موجود در کلاس Abstract می‌توانند مانند یک کلاس معمولی دارای سطح دسترسی باشند؛ ولی Interface‌ها فاقد این امکان هستند.
- 4- وقتی شما متدی را به کلاس Abstract اضافه می‌کنید، به طور خودکار به همه زیر کلاسها اعمال می‌شود؛ اما در Interface اگر متدی اضافه کنید باید در تمام زیر کلاسها آن را اعمال کنید.
- 5- کلاس‌های Abstract مانند کلاسهای معمولی می‌توانند دارای فیلد و عناصر دیگری (مثل ثابت‌ها) باشند؛ در حالیکه یک Interface فاقد این امکان می‌باشد. همچنین کلاس abstract می‌تواند شامل سازنده باشد، اما اینترفیس نمیتواند.
- 6- Abstract یکی از انواع کلاس است؛ ولی Interface کلاس نیست.

7- اینترفیس تنها میتواند از اینترفیس ارث بری کند اما کلاس abstract میتواند از اینترفیس، کلاس Abstract و یا سایر کلاس‌ها ارث بری کند.

سوال ۵

override کردن تابع و متغیر چه تاثیری در عملکرد متد در یک کلاس فرزند می‌گذارد؟ متد کلاس پدر اجرا نمیشود و متغیر عوض میشود

چطور 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 `method();`
- متدی داخل کلاس فرزند `super.method();`
- خارج از دو کلاس `pedar.method()` و `farzand.method();`

سوال ۶

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

میتوانید یک رابط تعریف کرده و چندین پیاده سازی از آن داشته باشید

با استفاده از این قابلیت می‌توان توابع یا متغیرهای مرجع را طوری ایجاد نمود، که در قسمت های مختلف برنامه رفتار متفاوتی داشته باشند.

سوال ۷

چرا از توابع و متدها در زبان برنامه نویسی استفاده می‌کنیم؟ خلاصه کردن استفاده مجدد 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؟

سوال ۸

کلاس درونی (inner class) چه انواعی دارد و هر کدام چه کاربردی در مدل‌سازی و توصیف موجودات دارد؟

1. Nested Inner class به تمامی متغیرهای Private کلاس بالا دستی خودش دسترسی دارد.
2. Method Local inner classes در داخل یکی از متدهای کلاس بالا دستی خود تعریف می‌شود و داخل همان متد قابل استفاده است.
3. Anonymous inner classes این نوع از کلاس تو در تو، اسمی ندارد و در جا تعریف و استفاده میشود.
4. این روش مخصوصا در JavaFX و برنامه‌نویسی گرافیکی بسیار کاربردی هستش و دائما استفاده میشود.

چگونه می‌توانیم یک شی از هر نوع ایجاد کنیم؟ Object obj;
 در صورت override شدن یک متد یا متغیر توسط یک کلاس درونی چگونه می‌توان به نسخه override شده از کلاس بیرونی دسترسی پیدا کرد؟
 super.method();
 super.variable();

سوال ۹

- کلمه کلیدی final روی هر کدام از موارد زیر چه تاثیری دارد؟
- تابع و متد override nemishan
 - تعریف کلاس extend nemishan
 - یک متغیر از نوع شی mishavad faghat dar constructor meghdar dehi
 - یک متغیر از نوع پایه mishavad faghat yek bar meghdar dehi

سوال ۱۰

- کلمه کلیدی static روی هر کدام از موارد زیر چه تاثیری دارد؟
- تابع و متد mishavad bedoone object seda zade
 - تعریف کلاس mishavad bedoone object farakhani
 - یک متغیر از نوع شی har gah taghir konad baraye tamam jaha taghir mikonad
 - یک متغیر از نوع پایه digar rabti be class nadarad va mostaghel ast