



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 میاد دانشجوها با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می کند.
- جبران 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 مفاهیم از طریق منابع درس
- جهت دهنده 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 آنلайн
- دقت کنید که این پرسشنامه تنها مرجع برای بازخورد پیشرفت مناسب شما در مطالعه منابع نیست. تمرین ها و پژوهه 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 آنلайн 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 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 کوئری و در بخش پرسشنامه بررسی پیشرفت بارگذاری کنید.

سوال ها

سوال ۱

خروجی این برنامه را بحسبت 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
B:15
A:11
B:16
B:16
B:16
B:17
B:17

سوال ۲

توضیح دهید 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؟

پاسخ :

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

سوال ۳

توضیح دهید که چرا از رابطه‌ها (interface) استفاده می‌کنیم. چه محدودیت‌های نسبت به یک کلاس دارند و چرا امکان پیاده‌سازی متدهای آنها داده شده است؟

پاسخ :

اینترفیس کاری می‌کند تا یک سری رفتار مشخص را بین کلاس‌هایی که آن را پیاده‌سازی می‌کنند به اشتراک بگذارند و این کار باعث کوتاه تر شدن کد می‌شود. محدودیت آنها نسبت به کلاس این است که ما اجازه تعریف عملکرد متدهای آن را نداریم و همچنین نمی‌توانیم شی از آن بسازیم و در واقع به این دلیل امکان پیاده‌سازی متدهای آنها داده شده که ویژگی و رفتارهای کلاس‌های پسر را در آن تعریف کنیم.

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؟

پاسخ:

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

فرق abstract و interface در این است که کل متدها در abstract بصورت خودکار تعریف می‌شوند ولی در کلاس abstract می‌توان از نوع‌های دیگری نیز استفاده کرد

سوال ۵

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

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

پاسخ:

وقتی متدهای override را می‌کنیم متدهای فرزند تغییر می‌کنند. پس از override شدن متدهای دسترسی به متدهای پدر از کلمه super استفاده می‌کنیم و در داخل کلاس از this استفاده می‌کنیم و اگر خارج دو کلاس باشد باید با ساخت شیی از آن کلاس آن متدهای فراخوانی کرد.

سوال ۶

توضیح دهید که منظور از چند ریختی در شی گرایی چیست و چه مزیت ایجاد می‌کند.

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 همه‌ی شکل‌ها دارند ولی تعریف مساحت هر شکل متفاوت است. مزیت آن نیز آن است که به ما این امکان را میدهد که از یک متدهای مختلف استفاده کنیم.

سوال ۷

چرا از توابع و متدهای 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؟

پاسخ:

زیرا استفاده از توابع و متدهای باعث سادگی برنامه می‌شود به این صورت که هر تابع قسمتی از برنامه را در دست می‌گیرد و مسئله را ساده‌تر می‌کند. همچنین برای خوانایی و سادگی فهمیدن کد بهتر است از نام‌های مرتبط با تابع و متدهای 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 متدها یا متغیر توسط یک کلاس درونی چگونه می توان به نسخه override شده از کلاس بیرونی دسترسی پیدا کرد؟

پاسخ :

کلاس داخلی ۳ 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 متدها آن شی استفاده شود. برای دسترسی به متدها override شده ای آن باید نام کلاس بیرونی را بنویسیم و سپس با کلمه `this` به آن ارجاع دهیم.

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

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 متدها
- 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 متدها استاتیک است برای استفاده از آن نیازی به ساخت شی نداریم و متدهای 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 متدها و متغیر های آن باید استاتیک باشند.