



# 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 دانشجویها با همدیگر که امکان ارائه بازخورد پیشرفت مطالعه به یکدیگر را ایجاد می‌کند.
- 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
- 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 پروژه‌ها 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 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();
        با ساخت این کانتراکتور مربوط به ای فراخوانده می شود و بدین ترتیب دستور های
        پرینت و پرینت ولیو در ای صدا زده می شوند پس خواهیم داشت:
B:5
A:0

        Classes.B b = new Classes.B(10);
        کانستراکتور بی صدا زده می شود و با توجه به آن خواهیم داشت:
B:15
B:15
B:10

        Classes.A c = b;
        در اینجا تنها سی هم برابر با بی می شود و کانستراکتور صدا زده نمی شود.

        b.print();
B:10

        c.print();
A:10

        ((Classes.A) b).print();
A:10

        b.printSuper();
A:10

        a.printCaller();
A:0

```

```

        b.printCaller();
B:15
B:15
B:15
        c.printCaller();
A:10
        incrementValue(a);
در اینجا اینت ولیو در ای به 1 میرسد و اینتیجر ولیو در ای به 6 میرسد و چیزی چاپ
نمی شود.
        a.printCaller();
A:1
        incrementValue(b);
در اینجا اینت ولیو به 11 و اینتیجر ولیو در بی به 16 می رسد و چیزی چاپ نمی شود.
        b.printCaller();
B:16
B:16
B:16
        incrementValue(c);
در اینجا اینت ولیو به 11 و اینتیجر ولیو در سی به 16 می رسد و چیزی چاپ نمی شود.
        c.printCaller();
A:11
        incrementValue(b.intValue(), b.integerValue());
در اینجا اینت ولیو به 12 و اینتیجر ولیو در بی به 17 می رسد و چیزی چاپ نمی شود.
        b.printCaller();
B:17
B:17
B:17
        c.printCaller();
A:11
    }
}

```

پاسخ:  
خط به خط در main توضیح داده شده است.

## 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؟

پاسخ:  
هدف از ارث بری در شی گرایی این است که وقتی یک مفهومی قرار است داشته باشیم که مفاهیم دیگری زیرمجموعه آن قرار می گیرند، از آن استفاده کنیم.  
در این حالت در واقع توابع کلاس اصلی در کلاس فرزند پیاده سازی می شوند اما می توانند کامل تر شوند و حتی توابع دیگری نیز داشته باشند.  
زمانی که بخواهیم از کلاس دیگری مصداقی در یک کلاس ایجاد کنیم، از composition استفاده می کنیم و زمانی که قرار باشد ارث بری به نحوی که گفته شد انجام بشه، از inheritance استفاده می کنیم.  
با دستور this

### 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؟

پاسخ:

برای اینکه یک سری چیزها ویژگی‌های مشترک دارند و به نوعی interface مجموعه‌ای از ویژگی‌هاست. متد‌ها در آن abstract هستند. چون قرار است رفتارها رو در خودش جا بده.

### 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:

کلاسی است که نمی‌توان از آن مصداق درست کرد. زمانی که مجموعه‌ای از کلاس‌های شبیه به هم داریم که می‌توانند کلاس پدر داشته باشند. آنگاه کلاس پدر abstract خواهد بود. امکان پیاده‌سازی متد‌ها در آن وجود ندارد.

### سوال ۵

override کردن تابع و متغیر چه تاثیری در عملکرد متد در یک کلاس فرزند می‌گذارد؟ 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
- متدی داخل کلاس فرزند
- خارج از دو کلاس

پاسخ:

با این کار می‌توان متد رو بازنویسی کرد. تنها با صدا کردن تابع با اضافه کردن super قبل از صدا زدن تابع با اسم کلاس پدر یا فرزند

### سوال ۶

توضیح دهید که منظور از چندریختی در شی گرای چيست و چه مزیتی ایجاد می‌کند.

پاسخ:

اجازه می‌دهد که یک تابع ورودی‌های مختلفی بگیرد. بنابراین نیازی نیست یک متد یکسان چندین جا تعریف شود.

## 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؟

پاسخ:

باید تمام مفاهیم و متغیرهایی که تنها مربوط به تابع هستند را داخل همان تعریف کنیم. برای این از متدها استفاده می‌کنیم که کارهای تکراری که قرار است انجام شود سریعتر رخ بدهند.

## 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؟

پاسخ:

## 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

پاسخ:

هیچگاه بدنه آن قابل تغییر نخواهد بود.  
هیچ کلاسی نمی‌تواند از آن ارث بری کند.  
نمی‌دونم تفاوتش با پایینی چیه!!  
ثابت و غیر قابل تغییر خواهد شد اندازه آن

## 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

پاسخ:

می تواند به عضو های **static** دسترسی داشته باشد و مقدار آنها را دستکاری کند و بدون ایجاد مصداق از کلاس می تواند فراخوانده شود و به کل کلاس متعلق است.

مطمئن نیستم اما فکر میکنم که تمام متد ها و متغیر های آن **static** می شوند و فراخوانی خود کلاس.

نمیدونم فرقش رو با پایینی!!!

قابل صدا زدن در همه جا که اجازه دسترسی داره بدون نیاز به ساخت مصداق از کلاس است.