

انجمن جاواکاپ تقدیم می کند

دوره برنامه نویسی جاوا

پروژه سوم

صادق علی اکبری

# یادآوری موضوع پروژه

- موضوع پروژه: نرم‌افزاری برای مدیریت یک کتابخانه عمومی
- مدیریت کتاب‌ها و اعضای کتابخانه
- امکان ثبت، حذف و ویرایش اعضا و کتاب‌ها
- امکان جستجو در کتاب‌ها و اعضا
- امکان امانت و عودت کتاب



# یادآوری نتایج فاز قبل (پروژه دوم)

- مفاهیم برنامه‌نویسی شیء‌گرا را به کار بستیم
- از مهارت‌های برنامه‌نویسی جدیدی که آموختیم، بهره گرفتیم



- بسته‌ها

- سطوح دسترسی

- ...

- امکانات جدیدی به پروژه اضافه کردیم

- ثبت، ویرایش، حذف و جستجو را برای اعضا و کتاب‌ها پیاده‌سازی کردیم

- امانت و عودت کتاب پیاده‌سازی شد



# نیازمندی‌های فاز سوم پروژه

- بخشهایی از پروژه را بازنویسی کنیم
- هدف این فاز: تمرین مفاهیم زیر
  - وراثت
  - چندریختی
  - واسط
  - مدیریت خطا و استثنا
  - نمودارهای UML
- در این فاز، امکانات جدیدی به برنامه اضافه نمی‌شود
- فقط زیرساخت و پیاده‌سازی آن بهبود می‌یابد و تصحیح می‌شود



# موضوع پروژه سوم

- در این بخش تعدادی کلاس و واسط معرفی می‌شوند
- کلاس‌های لازم را پیاده‌سازی کنید
- از این کلاس‌ها برای پیاده‌سازی پروژه استفاده کنید



```
interface Entity {  
    Integer getID();  
    void readFromConsole() throws BadEntityException;  
    void showOnConsole() ;  
}
```

```
abstract class Person implements Entity {...}
```

```
class Member extends Person {...}
```

```
class Book implements Entity {...}
```

```
enum Gender {  
    Male, Female  
}
```

```
interface Library {  
    void save(Entity entity);  
    void update(Entity e) throws EntityNotFoundException;  
    void delete(Entity e) throws EntityNotFoundException;  
    Entity[] find(Entity example);  
    void borrow(Member member, Book book);  
    void giveBack(Member member, Book book);  
}
```

```
class LibraryImpl implements Library {...}
```

- واسط‌های طراحی شده را در نظر بگیرید و سپس:

- ۱- کلاس‌های مطرح شده را پیاده‌سازی کنید

- ۲- برنامه را بازنویسی کنید

- از کلاس‌ها و واسط‌های جدید استفاده کنید

- استثناهای (exception) معرفی شده را پیاده‌سازی و استفاده کنید

- ۳- برای این طراحی، UML Class Diagram رسم کنید

- واسط‌ها و کلاس‌ها را در نمودار بگنجانید و ارتباط بین آن‌ها را مشخص کنید



- یکی از این دو روش را برای نگهداری اشیاء به کار گیرید:
  - یک آرایه از کتاب‌ها و یک آرایه از اعضا ایجاد کنید
  - و یا یک آرایه از همه موجودیت‌ها تعریف کنید
- در موارد لازم از عملگر `instanceof` برای تشخیص نوع شیء استفاده کنید

• مثلاً در متد `update` در `Library`

- برای این که بفهمیم موجودیت موردنظر کتاب است یا عضو
- روند اجرای برنامه، مشابه مراحل قبلی پروژه است
- یک برنامه مبتنی بر کنسول که یک منو در اختیار کاربر قرار می‌دهد





# حرفه‌ای باشید

- برنامه را به صورت کامل پیاده‌سازی کنید
- با برخی از نکات در اثر تکمیل برنامه مواجه می‌شوید
- برنامه را چند بار اجرا کنید و سناریوهای مختلف را آزمایش کنید
- برنامه‌ای که مفصلاً تست نشده، هنوز کامل نیست
- تولید مستندات مناسب (javadoc)
- در تهیه نمودار UML به جزئیات دقت کنید
- چه نوع ارتباطی بین کلاس‌ها و واسط‌ها هست؟
- هر جا ابهام داشتید، در اینترنت جستجو کنید



پایان

# تاریخچه تغییرات

نسخه	تاریخ	توضیح
۱.۰.۰	۱۳۹۴/۶/۲۳	نسخه اولیه ارائه آماده شد

