

به نام خدا
مستند پروژه برنامه نویسی پیشرفته
ترم 99-00

فاز 2

شهردار بکام

پس از پایان موفقیت آمیز فاز 1، شهردار ناکام به شهردار بکام تبدیل شده و حالا دیگه به داشتن یک شهر راضی نیست. (میشه گفت ابی هوس ریاست کشور کرده)



هشدار: هرگونه تشابه اسمی و تصویری کاملاً تصادفی است.

برای همین شهر ساخته شده در فاز یک را به عنوان پایتخت انتخاب کرده و چند شهر دیگر را ساخنه مجهز کرده و به شهروندانش معرفی میکند. ولی خب شهرهای ساخته شده شهروند ندارند و نمی توان برای هر شغلی نیروی مورد نیازش را استخدام کند. همه مردم در پایتخت متمرکز شده اند حالا فرصت آن است که ابی از آن همه ترمینالی که در فاز یک ساخته استفاده کند و تعدادی از شهروندانش را خودش انتخاب کرده و به شهرهای دیگر کشورش بفرستد و با این کار هم نیروی کار در شهرهای دیگر داشته باشد و بعد از این همه هزینه ای که فاز یک به بار آورد در آمدی از این طریق (حمل نقل مسافر) بدست بیاورد از طرفی چون فاز یک را به بهترین وجه ممکن پیاده سازی کردید ابی مشتری شما شده و باز فاز دوم را به شما می سپارد.

برای کاهش حجم پروژه، بخش بندرگاه و ایستگاه قطار و به تبع آن کلاس های قطار، وسیله نقلیه دریایی، کشتی و قایق نیاز به هیچگونه تغییری ندارند و همانگونه که در فاز یک بود باقی می ماند.

افرادی که تمایل به پیاده سازی این کلاس ها در فاز 2 دارند، نمره اضافه دریافت خواهند کرد.

به عنوان نکته اول تمامی نقص هایی که پروژه شما در فاز یک داشت باید برطرف شود و سپس فاز دو انجام شود.

در ابتدا کلاس شهر را در این فاز طوری تغییر دهید که هر شهر ویژگی ها و توابعش مخصوص خود باشد در واقع باید هر شهر شامل لیستی از شهروندان، ترمینال ها و هتل های مخصوص به خود باشد و توابع مورد نیاز این موجودیت ها نیز در همان شهر باشد.

(میتوانید برای راحتی کار شهروندان را بصورت یکجا و توسط یک تابع به یک شهر اضافه کنید.)

کلاس کشور

حال کلاس کشور را با ویژگی ها و قابلیت های زیر بسازید:

- مجموعه ای از شهر ها
- جمعیت کل
- بودجه کل که معادل مجموع بودجه شهر هاست
- قابلیت افزودن شهر جدید
- قابلیت دسترسی به شهرها و گرفتن اطلاعات آن ها

کلاس سفر

در پروژه خود پکیج سفر را ساخته و دو مورد زیر را در آن اضافه نمایید.

کلاس سفر را با ویژگی های ترمینال مبدا، ترمینال مقصد، لیست مسافران، راننده، وسیله نقلیه، ای دی، تاریخ و هزینه سفر، به همراه متد هایی که فکر میکنید نیاز است، پیاده سازی کنید. (تاریخ را میتوانید از جنس استرینگ گرفته و فرمت آن را هرطور که راحتید در نظر بگیرید.)

کلاس سفر باید قابل مقایسه باشد، از این رو باید از واسط کامپریل ارث بری کرده و مقایسه آن طوری باشد که ابتدا بر اساس تاریخ و سپس بر اساس هزینه مقایسه شود.

واسط سَفَرِبِل را که دارای چهار تابع زیر است، بسازید:

- تابع سفر جدید که آرگومان های ورودی آن تمامی ویژگی های کلاس سفر است.
- تابع مرتب سازی سفر
- تابع محاسبه هزینه سفر (آرگومان های ورودی این تابع لیست مسافران و وسیله نقلیه است).
- تابع تاریخچه سفر ها

به کلاس ترمینال لیستی از سفر ها را اضافه کنید.

همچنین کلاس ترمینال باید واسط سفرِبِل را ایمپلمنت کرده و متدهای آن را بازنویسی کند.

تابع محاسبه هزینه با توجه به تعداد مسافران و نوع وسیله نقلیه باید هزینه این سفر را محاسبه نماید.

تابع سفر جدید پس از گرفتن ویژگی های سفر

- سفر جدیدی ایجاد کرده آن را به لیست سفرهای ترمینال مبدا و مقصد اضافه میکند.
- ماشین و راننده از ترمینال مبدا حذف کرده و به ترمینال مقصد اضافه میکند.
- مسافران به همراه راننده را نیز از لیست شهروندان شهر مبدا حذف کرده و به لیست شهروندان شهر مقصد اضافه میکند. (دقت کنید که تنها افرادی که در شهر خود سرکار نیستند قابلیت رفتن به سفر را دارند)
- هزینه سفر به مقدار موجودی شهر مبدا اضافه خواهد شد.

برای راحتی کار میتوانید برای هر ترمینال بجای یک مجموعه از سفرها، دو مجموعه از سفرهای ورودی و سفرهای خروجی را ساخته و سفرهای ارسالی و دریافتی را جداگانه نگه دارید. همچنین میتوانید عملیات بالا را تابع تابع کرده و هرکدام را فراخوانی نمایید.

تابع مرتب سازی سفر نیز باید سفرها را با استفاده از یکی از روش های مرتب سازی ، سورت کند. (با توجه به اینکه کلاس سفر قابل مقایسه شد این کار بسیار راحت است.)
(استفاده از سورت های آماده جاوا منعی ندارد.)

تابع تاریخچه سفر باید تمامی سفر های ورودی یا خروجی یا هردو را به ترتیب چاپ کند.
(با استفاده از آرگومان هایی که برای این متد تعریف میکنید باید متد تشخیص دهد که باید سفر های ورودی را چاپ کند یا خروجی و یا هردو)

دقت کنید هیچ نیازی به بازنویسی دوباره این متدها در کلاس های فرزند ترمینال (فرودگاه و پایانه اتوبوس رانی) نیست.
با توجه به اینکه هر شهر آرایه ای از ترمینال ها را بصورت یکجا دارد، سعی کنید بخش سفرها را بصورت کلی پیاده سازی کرده تا از نوشتن کد تکراری برای هر نوع ترمینال (فرودگاه و پایانه اتوبوس رانی) جلوگیری شود.

کلاس های استثنا

یک پکیج به اسم اکسپشنز ساخته و تمامی اکسپشن ها با پیغام مناسب در آنجا نوشته میشوند.
اکسپشن سفر نامعتبر ، و لغو سفر و ورودی نامعتبر که هر کدام پیغام مناسب را چاپ میکنند بسازید.
و سپس اکسپشن های دیگری که هر یک به طور محسوس تر یک اکسپشن را توصیف میکنند بسازید واز کلاس های اکسپشن بالا ارث بری میکنند.

برای مثال در صورت مغایرت اطلاعات مثلا ترمینال مبدا زمینی باشد و ترمینال مقصد هوایی و یا ترمینال یا وجود خارجی نداشته یا ترمینال مبدا در شهر کنونی وجود ندارد ... باید نوعی اکسپشن پرتاب بشود ک از اکسپشن سفر نامعتبر ارث بری میکند .

و در صورتی که راننده ذکر شده در ترمینال مبدا حضور نداشته باشد و یا تعداد مسافران از نصف ظرفیت وسیله نقلیه کمتر باشد باید نوعی اکسپشن پرتاب بشود ک از اکسپشن لغو سفر ارث بری میکند .
توجه داشته باشید که خلاقیت در تفکیک این کلاس های اکسپشن جزِ نمره اصلی به حساب می آید.

مدیریت استثنا

در فهرست اصلی فاز یک باید یک گزینه اضافه کردن سفر افزوده شود و در صورت انتخاب آن گزینه توسط شهردار سپس شهردار باید بتواند مسافران این سفر را انتخاب کند(طریقه پیاده سازی انتخاب مسافر با خودتان) و بعد از آن هیچ فهرست دیگری باز نمیشود بلکه به ترتیب آی دی یا نام (ترمینال مبدا / ترمینال مقصد / ماشین) گرفته میشود و در صورت درست بودن اطلاعات سفر اعمال میشود و در صورت درست نبودن آن اکسپشن مربوطه پرتاب و پیغام مناسب چاپ میشود.
توجه داشته باشید که خطا ها در متد مین باید مدیریت شود.

همچنین برنامه شما باید تمامی خطاهایی که برای ورودی های نامعتبر در کل برنامه از سوی کاربر داده میشود را مدیریت کرده و بجای متوقف شدن برنامه، پیغام مناسبی چاپ کند و به منوی قبلی بازگردد.

منو های برنامه :

منوی ابتدایی باید شامل موارد ساختن شهر جدید، ورود به شهر مورد نظر ، نمایش اطلاعات شهرها (بصورت یکجا و بصورت تکی) و سایر آپشن هایی که فکر میکنید نیاز است باشد.

پس از ورود به هر شهر باید منوی دیگری نمایش داده شود که همان ویژگی های منوی قبلی در فاز یک را داشته و علاوه بر آن ویژگی افزودن نمایش سفر ها را نیز داشته باشد.

- دقت کنید که رابط کاربری بصورت قبل بوده و همه کارها را کاربر اصلی یا همان شهردار انجام میدهد.
- این نکته را توجه داشته باشید که نمایش اطلاعات باید بصورت کامل بوده و تمامی جزییات را نمایش دهد تا درستی پروژه را در هنگام اجرا بتوان بررسی نمود.