



مدرس: رامتین خسروی

طراحان: امیررضا نادی، مجید صادقی نژاد، سهیل
حاجیان منش، عرفان میرشمس، امیرحسین عارف زاده،
مهدی نائینی، مهرداد لیویان، طاها مجلسی

مهلت تحویل: چهارشنبه ۱۲ دی ۱۴۰۳، ساعت ۲۳:۵۵



مقدمه

این پروژه به جمع‌بندی آموخته‌های شما در این درس می‌پردازد. انتظار می‌رود مهارت‌هایی را که در تمرین‌های پیشین و سایر بخش‌های درس آموخته‌اید، در پیاده‌سازی این پروژه به کار گیرید. هدف از این فاز پروژه آشنایی با مفاهیم ارث‌بری^۱ و چندریختی^۲ است و انتظار می‌رود با استفاده از این مفاهیم پیاده‌سازی این فاز را انجام دهید.

در این فاز پروژه، تعدادی دستور در راستای گسترش منطق و افزایش هوشمندی برنامه به آن افزوده و پیاده‌سازی می‌شوند. قالب این دستورات مانند فاز قبلی می‌باشد و می‌توانید برای بررسی دقیق‌تر به فاز قبل مراجعه کنید. همچنین باید تمام دستورات فاز قبلی در این فاز نیز قابل اجرا باشد (به جز دستوراتی که در این فاز دستخوش تغییر شده‌اند، این دستورات باید به صورت جدید اجرا شوند). از قابلیت‌های جدیدی که باید در این فاز پیاده‌سازی کنید می‌توان به اضافه شدن انواع تخفیفات به سفارشات و اضافه شدن کیف پول برای هر شخص اشاره کرد.

¹ Inheritance

² Polymorphism

نکته قابل توجه در این پروژه این است که باید سعی کنید از قابلیت‌هایی که در فاز قبل توسعه داده‌اید به خوبی استفاده کنید و تا جای ممکن از کد تکراری پرهیز کنید.

شرح تمرین

در این فاز هر کاربر مقداری پول به عنوان سرمایه دارد و با توجه به آن می‌تواند غذا رزرو کند. همچنین رستوران‌های مختلف می‌توانند تخفیف‌های متنوعی را (که در ادامه توضیح داده می‌شوند) به صورت درصدی یا مقداری بر روی سفارش‌ها اعمال کنند. موارد ذکر نشده در فاز دو باید مشابه فاز یک در نظر گرفته شوند.

انواع تخفیف‌ها

1. **تخفیف بر روی یک غذای خاص:** هر رستوران می‌تواند بر روی بعضی از غذاهای منو، تخفیف اعمال کند. تخفیف در نظر گرفته شده برای هر غذا می‌تواند متفاوت با غذاهای دیگر باشد. همچنین رستوران‌های مختلف می‌توانند برای یک غذای یکسان، تخفیف‌های متفاوتی داشته باشند.
2. **تخفیف اولین سفارش:** هر رستوران می‌تواند بر روی اولین سفارش هر فرد این تخفیف را اعمال کند. این تخفیف بر روی مبلغ نهایی سفارش اعمال می‌شود.
3. **تخفیف مبلغ کل سفارش:** این تخفیف بر اساس مبلغ سفارش پس از اعمال دو تخفیف بالا مشخص می‌شود. به این صورت که اگر مبلغ کل سفارش پس از اعمال تخفیف‌های قبلی از کف سفارش مشخص شده رستوران بیشتر باشد، این نوع تخفیف اعمال می‌شود.

فرایند اعمال تخفیف‌ها حتما باید به صورت شی‌گرا و با تکیه بر مفاهیم ارث‌بری و چندریختی پیاده‌سازی شود.

- تمامی تخفیف‌ها به غیر از تخفیف نوع اول بر روی مبلغ کل سفارش اعمال می‌شود.
- هر کدام از تخفیف‌های بالا می‌تواند به صورت درصدی (کاهش درصد مشخصی از مبلغ مورد نظر) و یا مقداری (کاهش مقداری مشخصی از مبلغ مورد نظر) اعمال شود.
- در تخفیف نوع اول، تخفیف به هر قلم سفارش که شامل تخفیف باشد تعلق می‌گیرد. به عنوان مثال اگر از یک آیتم منو که شامل تخفیف است سه عدد سفارش داده باشیم، به ازای هر کدام تخفیف جداگانه اعمال می‌شود.
- این تخفیف‌ها به انتخاب خود رستوران اعمال می‌شوند و ممکن است رستورانی نخواهد تمامی این تخفیف‌ها را برای سفارشات خود در نظر بگیرد. این که هر رستوران چه تخفیفاتی بر روی سفارشات خود اعمال می‌کند در فایل ورودی تخفیف‌ها مشخص شده است.
- ترتیب اعمال تخفیف‌ها، به همان ترتیب بالا می‌باشد. به این صورت که ابتدا تخفیف هر غذا (در صورت وجود) اعمال می‌شود و سپس تخفیف سفارش اول اعمال می‌شود. نهایتاً اگر مبلغ باقی‌مانده در شرط تخفیف مبلغ کل صادق باشد، این تخفیف هم اعمال می‌شود.

قالب فایل‌های ورودی

در این فاز یک فایل دیگر به لیست فایل‌های ابتدای برنامه اضافه می‌شود که شامل اطلاعات تخفیف‌ها می‌باشد.

پیکربندی نحوه اجرای برنامه

```
./UTaste </path/to/restaurants/file.csv>  
</path/to/districts/file.csv> <path/to/discounts/file.csv>
```

فایل تخفیف‌ها

پیکربندی فایل تخفیف‌ها

```
restaurant_name, total_price_discount, first_order_discount, food_discount  
[restaurant_name], [type;minimum;value], [type;value], [type;food1:value1|type;food2:value2|...|type;food_n:value_n]
```

اگر تخفیف مورد نظر به صورت درصدی باشد، فیلد type با کلمه percent پر می‌شود و در غیر این صورت از کلمه amount استفاده می‌شود که نشان‌دهنده تخفیف به صورت مقداری است. فیلد minimum نیز مربوط به کف میزان خرید می‌باشد. همچنین اگر رستوران مورد نظر فاقد یکی از این تخفیف‌ها باشد، عبارت none در فیلد مورد نظر قرار خواهد گرفت.

نمونه فایل رستوران‌ها

```
restaurant_name, total_price_discount, first_order_discount, food_discount  
sib, percent;800;7, none, amount;burger:20|percent;pizza:50
```

دستورات جدید

افزایش موجودی

این دستور موجودی کیف کاربر را به اندازه مشخص شده افزایش می‌دهد. در صورتی که مقدار درخواستی منفی یا چیزی به غیر از عدد باشد پیغام **Bad Request** نمایش داده خواهد شد.

ورودی

`POST increase_budget ? amount <amount>`

خروجی

`OK | Bad Request | Permission Denied`

نمونه ورودی اول

`POST increase_budget ? amount 15000`

نمونه خروجی اول

`OK`

نمونه ورودی دوم

`POST increase_budget ? amount pizza`

نمونه خروجی دوم

`Bad Request`

مشاهده موجودی

با این دستور، کاربر می‌تواند موجودی کیف پول خود را مشاهده کند.

ورودی
GET show_budget ?
خروجی
<budget> Permission Denied
نمونه ورودی اول
GET show_budget ?
نمونه خروجی اول
15000
نمونه ورودی دوم
GET show_budget ?
نمونه خروجی دوم
0

دستورات تغییر یافته

مشاهده اطلاعات یک رستوران

در این فاز، کاربر باید علاوه بر اطلاعاتی که در فاز قبل نمایش داده می‌شد، اطلاعات تخفیف‌های موجود در رستوران را نیز مشاهده کند. اگر رستورانی تخفیفی را نداشته باشد لازم به نمایش آن نیست.

برای هر تخفیف باید اطلاعات زیر نمایش داده شود:

- نوع تخفیف (درصدی/مقداری)
 - درصد/مقدار تخفیف
 - برای تخفیف مبلغ کل سفارش، حداقل مبلغ کل سفارش که تخفیف به آن تعلق می‌گیرد
 - برای تخفیف روی یک آیتم خاص، باید میزان تخفیف برای هر آیتم منو وارد شود.
- اگر رستوران مورد نظر تخفیفی را شامل نشود، تخفیف مورد نظر به طور کل نباید چاپ شود. برای مثال اگر رستوران مورد نظر هیچ نوع تخفیفی نداشته باشد، خروجی کاملاً مشابه فاز قبل می‌شود.

ورودی

```
GET restaurant_detail ? restaurant_name "<restaurant_name>"
```

خروجی

```
Name: <restaurant_name>
District: <restaurant_district_name>
Time: <start_time>--<end_time>
Menu: <food_name1>(<food_price1>), <food_name2>(<food_price2>)
<table_id1>: (<reserve_start11>,
<reserve_end11>),(<reserve_start12>, <reserve_end12>)
<table_id2>: (<reserve_start21>,
<reserve_end21>),(<reserve_start22>, <reserve_end22>)
Order Amount Discount: <percentage_or_amount>, <min_order_amount>,
<discount_amount>
Item Specific Discount: <food_name1>(<percentage_or_amount1>:
<discount_value1>), <food_name2>(<percentage_or_amount2>:
<discount_value2>)
First Order Discount: <percentage_or_amount>, <discount_amount>
... | Not Found | Bad Request
```

نمونه ورودی اول

GET restaurant_detail ? restaurant_name "Chicken Family"

نمونه خروجی اول

Name: Chicken Family
District: Yousef Abad
Time: 12-22
Menu: Sib Zamini(70000), Sokhari(200000)
1: (14-16), (17-19)
2:
3:
4: (21-22)
Order Amount Discount: percentage, 400000, 10
Item Specific Discount: Sokhari(amount: 40000), Pizza(percentage: 30)
First Order Discount: amount, 60000

نمونه ورودی دوم

GET restaurant_detail ? restaurant_name "Na Kojab Abad"

نمونه خروجی دوم

Not Found

نمونه ورودی سوم

GET restaurant_detail ? restaurant_name "Porteghal"

نمونه خروجی سوم

Name: Porteghal
District: Yousef Abad
Time: 12-22
Menu: Sib Zamini(70000), Dough(2000)
1: (14-16), (17-19)
2:
3:
4: (21-22)
First Order Discount: amount, 60000

رزرو رستوران

با وارد کردن این دستور، منطق فاز قبل به طور کامل اجرا می‌شود، اما باید بخش‌های زیر به آن اضافه شود.

- مبلغ خام سفارش (قبل از هر گونه تخفیف)
 - مقدار تخفیف به ازای هر یک از سه تخفیف گفته شده (برای تخفیف‌هایی که درصدی هستند باید مقدار تخفیف اعمال شده را محاسبه و نمایش دهید)
 - مقدار کل تخفیف (جمع کل تخفیف‌های مورد قبل)
 - مبلغ پرداختی نهایی (پس از اعمال تخفیف‌ها)
- همچنین باید مبلغ پرداختی نهایی (در صورتی که خطایی رخ ندهد)، از کیف پول فرد کم شود. اگر فرد موجودی کافی را نداشته باشد، تنها عبارت **Bad Request** به عنوان خطا چاپ می‌شود.

ورودی

```
POST reserve ? restaurant_name "Khoshmaze" table_id "3" start_time "15" end_time "17" foods "burger"
```

خروجی

```
Reserve ID: <reserve_id>
Table <table_id> for <start_time> to <end_time> in
<restaurant_name>
Original Price: <price_of_foods>
Order Amount Discount: <value1>
Total Item Specific Discount: <value2>
First Order Discount: <value3>
Total Discount: <value4>
Total Price: <price_after_discounts> |
Bad Request | Permission Denied | Not Found
```

نمونه ورودی اول

```
POST reserve ? restaurant_name "Khoshmaze" table_id "3" start_time "15" end_time "17" foods "Mahi,Torshi"
```

نمونه خروجی اول

```
Reserve ID: 10
Table 3 for 15 to 17 in Khoshmaze
Original Price: 300000
Order Amount Discount: 6000
Total Item Specific Discount: 20000
First Order Discount: 4000
Total Discount: 30000
Total Price: 270000
```


نمونه ورودی دوم

```
POST reserve ? restaurant_name "Khoshmaze" table_id "3" start_time "15" end_time "17" foods "Kabab,Torshi"
```

نمونه خروجی دوم

Bad Request

نمایش رزروهای انجام شده توسط کاربر

در این فاز باید علاوه بر اطلاعات فاز قبل، باید مبلغ کل یک رزرو قبل و بعد از اعمال تخفیف هم نمایش داده شود. اگر به رزروی هیچ تخفیفی تعلق نگرفته بود، این دو مبلغ برابر می‌باشد. همچنین در صورتی که رستورانی رزرو شده باشد اما غذایی سفارش داده نشده باشد، مبلغ رزرو قبل و بعد از صفر می‌باشد. قالب ورودی و سایر اطلاعات خروجی دستور مشابه فاز قبل می‌باشد.

ورودی

```
GET reserves ? restaurant_name "<restaurant_name>" reserve_id "<reserve_id>"
```

خروجی

```
<reserve_id1>: <restaurant_name1> <table_id1>  
<start_time1>-<end_time1> <food11(number11)> <food12(number12)>  
<price_before_discount1> <price_after_discount1>  
<reserve_id2>: <restaurant_name2> <table_id2>  
<start_time2>-<end_time2> <food21(number21)> <food22(number(22))>  
<price_before_discount2> <price_after_discount2>  
...| Permission Denied | Bad Request | Empty
```

نمونه ورودی اول

```
GET reserves ?
```

نمونه خروجی اول

```
1: Gotham 12 18-19  
1: Espinas Pallas 7 20-21 Steak(1) French Fries(1) 1000000 1000000  
3: Espinas Pallas 9 21-23 Pizza(3) 2000000 1850000
```

نمونه ورودی دوم

```
GET reserves ? restaurant_name "Espinas Pallas"
```

نمونه خروجی دوم

```
1: Espinas Pallas 7 20-21 0 0
3: Espinas Pallas 9 21-23 Pizza(3) 2000000 1850000
```

نمونه ورودی سوم

```
GET reserves ? restaurant_name "Espinas Pallas" reserve_id "3"
```

نمونه خروجی سوم

```
3: Espinas Pallas 9 21-23 Pizza(3) 2000000 1850000
```

نمونه ورودی چهارم

```
GET reserves ? restaurant_name "sib"
```

نمونه خروجی چهارم

Empty

حذف رزرو

تفاوتی که این دستور با فاز قبل دارد در این است که بعد از حذف رزرو توسط کاربر، باید 60 درصد از مبلغ کل سفارش به موجودی کیف پول کاربر برگردانده شود. قالب ورودی و خروجی دستور مشابه فاز قبل می‌باشد.

ورودی

```
DELETE reserve ? restaurant_name "<restaurant_name>" reserve_id
"<reserve_id>"
```

خروجی

OK | Not Found | Bad Request

نمونه ورودی اول

```
DELETE reserve ? restaurant_name "Haida" reserve_id "3"
```

نمونه خروجی اول

OK

نمونه ورودی دوم

```
DELETE reserve ? restaurant_name "Haida" reserve_id "6785"
```

نمونه خروجی دوم

Not Found

نکات و نحوه تحویل

- برای تحویل این پروژه، لازم است کد خود را در یک مخزن³ در GitHub بارگذاری کنید و سپس لینک مخزن به همراه Hash آخرین کامیت⁴ را در صفحه eLearn درس بارگذاری نمایید.
نمونه متن خواسته شده در سامانه ای‌لرن (سه بخش <username> و <repository_name> و <last_commit_hash> را جایگزین کنید):

```
https://github.com/<username>/<repository_name>  
<last_commit_hash>
```

- دقت کنید که پروژه شما باید Multi-file باشد و Makefile داشته باشد. همین‌طور در Makefile خود مشخص کنید که از استاندارد c++20 استفاده می‌کنید.
- فایل‌های خود را در مخزن قبلی به نام <SID>-F03-AP به نام **AP-UT@** را به مخزن اضافه کنید. دقت داشته باشید که برای هر سه فاز از این مخزن استفاده خواهید کرد و همه تغییرات خود را در شاخه⁵ main انجام دهید و نیازی به پیاده‌سازی شاخه‌ها نیست (هر چند برای کارایی بهتر توصیه می‌شود انجام دهید ولی در پایان تغییرات خود را به شاخه main نیز منتقل کنید). دقت داشته باشید که چون تحویل هر سه فاز با یکدیگر خواهد بود مشخصات کامیت خود را در هر فاز به درستی وارد کنید.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++20 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده است اجتناب کنید.
- دقت کنید که نام فایل اجرایی شما باید UTaste (بدون هیچ پسوندی مانند exe یا out) باشد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.
- توجه داشته باشید که حالت‌های خاصی که در صورت پروژه ذکر نشده است در تست‌های خودکار نخواهد بود و می‌توانید به هر شکلی که مد نظر دارید آن‌ها را مدیریت کنید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.
- توجه کنید که رعایت نکردن ساختار گفته شده در نام‌گذاری مخزن، فایل‌کد، فایل اجرایی و آپلود موارد خواسته شده باعث کسر 5 درصد از نمره شما خواهد شد.

³ Repository

⁴ Commit

⁵ Branch

نمرات

- تمیزی کد

- رعایت کردن نام‌گذاری صحیح و انسجام⁶
- عدم وجود کد تکراری
- رعایت دندان‌گذاری⁷
- عدم استفاده از متغیرهای گلوبال
- استفاده صحیح از متغیرهای ثابت⁸ به جای Magic Value-ها
- ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند

- درستی کد

- آزمون‌های خودکار
- پیاده‌سازی صحیح کارکردهای خواسته شده

- طراحی

- طراحی صحیح و منطقی در شی‌گرایی و ارث‌بری (پیاده‌سازی مکانیزم‌های مربوط به تخفیف با کمک ارث‌بری و چندریختی)
- رعایت Encapsulation
- جداسازی منطق کد از ورودی/خروجی و استفاده از کلاس جداگانه برای مدیریت دستورات
- استفاده مناسب از استثناها برای مدیریت خطا
- میک‌فایل و چندفایلی

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.

⁶ Consistency

⁷ Indentation

⁸ Constant