ورد جادویی

سارا تابستان گذشته بالاخره نامه‌ای که مدت‌ها انتظارش را کشیده بود، دریافت کرد و الان چندهفته‌ای است که مشغول تحصیل در مدرسه‌ی جادوگری هاگوارتز است. همه چیز داشت خوب پیش می‌رفت تا اینکه یک روز که سارا در کتابخانه مشغول مطالعه و تمرین وردهای جدید از روی کتاب وردها بود، یکی از وردها را اشتباه تلفظ کرد و ناگهان کتاب پر از انبوهی از کلمات درهم و نامفهوم شد. او درحالی‌که بسیار آشفته بود، برای حل این مشکل به سراغ پروفسور مک گونگال رفت. پروفسور به او گفت که راه‌حل این مشکل، در میان انبوه کلمات مبهم موجود در کتاب مخفی شده، فقط کافی است کلماتی که تنها یک‌بار در هر صفحه تکرار شده‌اند را کنار هم بگذارد، و ورد به‌دست‌آمده را بخواند تا کتاب به حالت اولیه‌اش بازگردد.

ازآنجاکه سارا وقت ندارد تمام کلمات کتاب را بخواند و کلمات منحصربه‌فرد را کنار هم بچیند، از شما می‌خواهد که به او کمک کنید. برنامه‌ای بنویسید که یک‌رشته از کلمات را به‌عنوان ورودی بگیرد و کلماتی را که تنها یکبار در آن رشته تکرار شده‌اند، در خروجی چاپ کند.

ورودی

رشته‌ای از کلمات را به‌عنوان ورودی دریافت کنید. کلمات ممکن است شامل حروف کوچک و بزرگ انگلیسی، اعداد و علائم نگارشی باشند. کلمات با یک فاصله از هم جدا می‌شوند. حروف کوچک و بزرگ باید یکسان در نظر گرفته شوند (یعنی "book" و "BOOK" به‌عنوان یک کلمه حساب شوند). علائم نگارشی نباید بخشی از کلمات حساب شوند؛ یعنی کلماتی مانند "Spell" و "Spell! " باید به‌عنوان یک کلمه در نظر گرفته شوند.

خروجی

خروجی باید کلمات منحصربه‌فرد را به‌صورت یک جمله به ترتیبی که زودتر در کتاب قرار گرفته اند در یک خط نشان دهد. خروجی به بزرگی و کوچکی حروف حساس نیست.

مثال

ورودی نمونه

```
Upon the murmurous! whisp, to summon as forth of forgotten shadows12 of Lum
```

خروجی نمونه

```
Luminara Aetheris Ignis Eterna254
```

بانکداری مدرن

عصر یک روز سرد پاییزی، آقای محتشمی، رئیس یکی از بزرگ‌ترین بانک‌های شهر، درحالی‌که در دفترش نشسته بود، چای می‌نوشید و از پنجره به بیرون نگاه می‌کرد. فکرش حسابی مشغول بود. فرزندش دیروز ده‌ساله شد و دیگر وقت آن بود که مدیریت مالی را به او بیاموزد. دلش می‌خواست که در راستای آموزش این امر مهم، یک حساب پس‌انداز برای فرزندش در بانکشان افتتاح کند، اما واگذاری مدیریت حساب بانکی به یک کودک کار عاقلانه‌ای نبود. پس از مدتی فکرکردن، ایده‌ای به ذهنش رسید. او تصمیم گرفت سیستم بانکشان را به‌گونه‌ای تغییر دهد که بتوان برای کودکان، حساب ویژه‌ای باز کرد که دارای محدودیت‌هایی باشد و والدین بتوانند با خیال راحت، مدیریت مالی را به فرزندانشان آموزش دهند. او سپس با شما تماس گرفت و درخواست کرد که پیاده‌سازی این سیستم بانکی جدید را به عهده بگیرید. آقای محتشمی کلاس ها و متود های پایه برای پیاده سازی این سیستم را نیز در اختیار شما گذاشت. شما می‌توانید فایل اولیه این پروژه را از این لینک دانلود کنید.

برای پیاده‌سازی این مسئله، دو کلاس به نام‌های BankAccount و KidsAccount تعریف شده است. کلاس BankAccount به‌عنوان پایه‌ای برای انواع حساب‌ها عمل می‌کند و شامل متدهای واریز و برداشت و نمایش تاریخچه است. کلاس KidsAccount از BankAccount ارث می‌برد و محدودیت‌های خاصی را برای حساب‌های پس‌انداز کودکان پیاده‌سازی می‌کند.

کلاس BankAccount

این کلاس شامل ویژگی‌ها و متدهای زیر است:

ویژگی balance : موجودی حساب که به‌صورت خصوصی (private) نگهداری می‌شود.

متد deposit (double amount) : این متد مقدار ورودی را به موجودی اضافه می‌کند. اگر مبلغ واریزی منفی باشد، باید پیامی نمایش داده شود که `Invalid Amount` . اگر عملیات موفقیت آمیز باشد عبارت `Deposit: $amount. Current Balance: $_balance` را نمایش می‌دهد.

متد withdraw (double amount) : این متد مقدار ورودی را از موجودی کسر می‌کند. اگر موجودی کافی نباشد، پیامی مبنی بر ناکافی بودن موجودی نمایش می‌دهد `Not Enough Balance` . اگر مبلغ منفی باشد، پیامی مبنی بر `Invalid Withdraw` نشان داده می‌شود. اگر عملیات موفقیت آمیز باشد عبارت `Withdrew: $amount. Current Balance: $_balance` را نمایش می‌دهد.

متد showTransactionHistory () : این متد تاریخچه انتقالات حساب را نمایش می‌دهد. قالب نمایش تاریخچه به صورت زیر می‌باشد:

```
Transaction History:
Deposited: 20.0
Withdrew: 10.0
Withdrew: 5.0
Deposited: 20.0
```

اگر تاریخچه ای موجود نباشد عبارت `No Transactions Yet` نمایش داده می‌شود.

کلاس KidsAccount

این کلاس از BankAccount ارث می‌برد و محدودیت برداشت را به آن اضافه می‌کند.

متد withdraw (double amount) : این متد از BankAccount بازنویسی (override) می‌شود. علاوه بر بررسی موجودی کافی، محدودیت حداکثر 100 واحد پولی را برای هر تراکنش برداشت اعمال می‌کند. اگر مبلغ برداشت بیشتر از 100 باشد، پیامی نمایش داده می‌شود که `Withdraw Is Not Possible` .

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده شده. توجه شود که ورودی های داده شده بر اساس پروژه اولیه که در اختیار شما قرار گرفته در نظر گرفته شده است.

ورودی نمونه ۱

```
1
1
200
2
-2
2
100
4
```

در اینجا با وارد کردن عدد 1 یک اکانت معمولی ساخته می‌شود. سپس با وارد کردن 1 بعدی، عملیات واریز به حساب به مبلغ 200 انجام می‌شود. سپس با انتخاب 2 عملیات برداشت از حساب انجام می‌شود که چون مبلغ وارد شده نامعتبر است، باید پیغام Invalid Withdraw نمایش داده شود. سپس دوباره عملیات برداشت انتخاب شده و اینبار مبلغ برداشتی معتبر و مقدار 100 می‌باشد. در نهایت با وارد کردن عدد 4 از سیستم بانکی خارج می‌شود.

خروجی نمونه ۱

```
Deposited: 200.0. Current Balance: 200.0
Invalid Withdraw
Withdrew: 100.0. Current Balance: 100.0
```

توجه داشته باشید که در اینجا فقط خروجی متدها نمایش داده شده و خروجی لیست های انتخابی نمایش داده نشده است.

ورودی نمونه 2

```
2
1
200
2
105
2
90
3
4
```

در اینجا با وارد کردن عدد 2 یک اکانت کودک ساخته می‌شود. سپس با وارد کردن عدد 1 عملیات واریز به حساب به مبلغ 200 انجام می‌شود. سپس با انتخاب 2 عملیات برداشت از حساب انجام می‌شود که مبلغ وارد شده بیش از 100 بوده و برای حساب کودکان امکان‌پذیر نیست، پس باید پیغام Withdraw Is Not Possible نشان داده شود. سپس دوباره عملیات برداشت انتخاب شده و اینبار مبلغ برداشتی معتبر و مقدار 90 می‌باشد. سپس عدد 3 وارد شده که تاریخچه‌ی حساب را نمایش می‌دهد. در نهایت با وارد کردن عدد 4 از سیستم بانکی خارج می‌شود.

خروجی نمونه 2

```
Deposited: 200.0. Current Balance: 200.0
Withdraw Is Not Possible
Withdrew: 90.0. Current Balance: 110.0
Transaction History:
Deposited: 200.0
Withdrew: 90.0
```

توجه داشته باشید که در اینجا فقط خروجی متدها نمایش داده شده و خروجی لیست های انتخابی نمایش داده نشده است.

متد های خواسته شده را در پروژه اولیه کامل کنید و در نهایت فایل main.dart را به صورت زیپ‌شده آپلود کنید.

معجون جادویی برای همه

به‌محض اینکه شما از بانک آقای محتشمی خارج شدید، یک نامه‌ی پرنده به‌سرعت به‌طرف شما آمد و در دستانتان قرار گرفت. روی نامه نوشته بود: "ارسال شده از مدرسه‌ی جادوگری هاگوارتز".

آهی کشیدید و شروع به خواندن نامه کردید. همان‌طور که حدس می‌زدید، نامه از طرف سارا بود. او بالاخره به کمک شما موفق شده بود کتاب وردش را به حالت عادی برگرداند و از پروفسور مک گونگال نمره‌ی خوبی بگیرد. همچنین نوشته بود که در کلاس معجون‌سازی جزو شاگردهای ممتاز است و تعدادی مشتری از سرتاسر دنیا پیدا کرده که خواهان معجون‌های جادویی برای رفع مشکلاتشان هستند. او این بار از شما می‌خواست که یک فروشگاه معجون فروشی آنلاین برای او پیاده‌سازی کنید، تا بتواند به کمک آن از سرتاسر دنیا، سفارش‌ها را دریافت و معجون‌های جادویی‌اش را به مشتریان ارسال کند. هرچند که شما باور نمی‌کردید کسی واقعاً معجون‌های سارا را بخرد، اما تصمیم گرفتید در پیاده‌سازی این سیستم فروشگاهی نیز به او کمک کنید.

برای این کار دو کلاس Order و Store نیاز دارید که جزئیات مربوط به پیاده‌سازی آنها به شرح زیر است:

کلاس Order

کلاس Order مسئول ذخیره اطلاعات هر سفارش است و شامل ویژگی‌ها و متدهای زیر خواهد بود:

ویژگی orderId (شناسه سفارش): یک عدد که شماره سفارش را ذخیره می‌کند و برای هر سفارش منحصر به فرد است.

ویژگی customerName (نام مشتری): نام مشتری که سفارش را ثبت کرده است.

ویژگی totalAmount (مبلغ کل): مبلغ کل سفارش را نگه می‌دارد.

ویژگی status (وضعیت سفارش): وضعیت سفارش که می‌تواند یکی از مقادیر زیر باشد:

```
Processing
Sent
Delivered
Canceled
```

متد updateStatus (String newStatus) : این متد وضعیت سفارش را به مقدار جدیدی که به آن ارسال شده تغییر می‌دهد. همچنین اگر وضعیت ورودی معتبر نباشد، پیامی مبنی بر `Invalid Status` نمایش می‌دهد.

کلاس Store

کلاس Store برای مدیریت لیستی از سفارش‌ها استفاده می‌شود. این کلاس شامل ویژگی‌ها و متدهای زیر است:

ویژگی orders (لیست سفارش‌ها): یک لیست از تمام سفارش‌های موجود در فروشگاه را ذخیره می‌کند.

متد addOrder (Order order) : برای اضافه‌کردن سفارش جدید به لیست orders. اگر سفارش از قبل موجود باشد پیام `Order already exists` نمایش داده می‌شود. اگر اضافه کردن سفارش جدید موفقیت آمیز باشد پیام `Order added successfully` را نمایش می‌دهد.

متد findOrderById (int orderId) : جستجو برای یافتن سفارش بر اساس orderId. اگر سفارش پیدا شود، آن را به صورت زیر نمایش می‌دهد.

```
Order ID: 1
Customer Name: sara
Total Amount: $10.00
Status: Sent
```

در غیر این صورت، پیامی مبنی بر `Order not found` چاپ می‌کند.

متد calculateTotalSales () : مجموع مبالغ سفارشاتی که وضعیت آن‌ها "Canceled" نیست را محاسبه و به صورت زیر نمایش می‌دهد.

```
Total Sales (excluding canceled orders): $10.00
```

متد displayOrdersByStatus (String status) : تمام سفارش‌هایی که وضعیت موردنظر را دارند به ترتیب مبلغ سفارش (از بیشترین به کمترین) به صورت زیر نمایش داده می‌شوند: برای مثال status مورد نظر در اینجا Sent است:

```
Orders with status Sent
---------
Order ID: 1
Customer Name: ali
Total Amount: $20.00
Status: Sent
---------
Order ID: 12
Customer Name: sara
Total Amount: $15.00
Status: Sent
```

متد removeOrder (int orderId) : حذف سفارش از لیست بر اساس orderId. اگر حذف کردن سفارش جدید موفقیت آمیز باشد پیام `Order removed successfully` نمایش داده می‌شود. اگر آیدی یافت نشود پیام `Order not found` نمایش داده می‌شود.

متد updateOrderStatus (int orderId, String newStatus) :این متود وضعیت یک سفارش را با استفاده از شناسه‌ی آن تغییر می‌دهد. اگر آپدیت کردن سفارش جدید موفقیت آمیز باشد پیام `Order updated successfully` نمایش داده می‌شود. اگر سفارشی با شناسه‌ی موردنظر یافت نشود پیام `Order not found` نمایش داده می‌شود.

ورودی

هر یک از دستورات زیر می‌توانند به عنوان ورودی داده شوند:

`Add` : پس از وارد شدن این دستور، در خط بعدی مشخصات سفارش جدید به صورت زیر از کاربر گرفته شده و متد addOrder فراخوانی می‌شود.

```
order ID_customer name_total amount_order status
```

`Find` : پس از وارد شدن این دستور، در خط بعدی شناسه‌ی سفارش مورد نظر داده شده و متد findOrderById صدا زده می‌شود.

`Total` : پس از وارد شدن این دستور، باید متد calculateTotalSales فراخوانی شود.

`Status` : پس از وارد شدن این دستور، در خط بعدی status مورد نظر دریافت شده و سپس متد displayOrdersByStatus فراخوانی می‌شود.

`Remove` : پس از وارد شدن این دستور، در خط بعدی شناسه‌ی سفارش مورد نظر داده شده و متد removeOrder فراخوانی می‌شود.

`Update` : با وارد شدن این دستور، در خط بعدی، شناسه و وضعیت جدید به صورت زیر داده شده و متد updateOrderStatus فراخوانی می‌شود.

```
orderId_newStatus
```

`Exit` : با وارد کردن این دستور در ورودی، کاربر از برنامه فروشگاه خارج می‌شود.

مثال

در اینجا چند نمونه برای فهم بهتر صورت سوال و قالب ورودی و خروجی تست‌ها داده شده.

ورودی نمونه ۱

```
Add
1_sara_100_Sent
Add
2_sara_50_Sent
Add
3_sara_11_Processing
Find
1
Total
Status
Sent
Remove
2
Update
3_Sent
Exit
```

خروجی نمونه ۱

```
Order added successfully
Order added successfully
Order added successfully
Order ID: 1
Customer Name: sara
Total Amount: $100.00
Status: Sent
Total Sales (excluding canceled orders): $161.00
Orders with status Sent
---------
Order ID: 1
Customer Name: sara
Total Amount: $100.00
Status: Sent
---------
Order ID: 2
Customer Name: sara
Total Amount: $50.00
Status: Sent
Order removed successfully
Order updated successfully
```