

کیف پول

توجه: این سؤال دارای دآوری خودکار نیست و نمره‌ی آن پس از چک کردن دستی کدها، اعلام می‌شود.

کیانوش در حال گذراندن تعطیلات نوروز است و به این نتیجه رسیده اگر فقط یک چیز باشد که خیلی در شرایط کنونی به آن نیاز داشته باشد، آن یک کیف پول هوشمند است! کیانوش از شما کمک می‌خواهد تا برنامه‌ای بنویسید که به او میزان موجودی کیف پولش را بگوید و در صورتی که می‌خواهد خریدی بیشتر از بودجه‌اش بکند، به او هشدار دهد. همچنین بتواند مقداری پول پس‌انداز کند و هر زمان که تنها ۵۰ هزار تومان موجودی برایش باقی ماند، به او اطلاع‌رسانی کند.

در این سوال، شما باید کلاسی با نام `Wallet` بنویسید به طوری که:

- دو ویژگی `total` و `saving` داشته باشد. `total` متغیری از جنس `int` درواقع کل موجودی کیف پول به غیر از پول‌های پس‌انداز شده است. `saving` نیز متغیری از جنس `float` است که مقدار پول پس‌انداز شده است. دقت کنید هر دو متغیر `total` و `saving` نباید از خارج از کلاس قابل دسترسی باشند.
- یک متد به نام `account_balance` داشته باشد که موجودی کیف پول را برگرداند.
- یک متد به نام `add` داشته باشد که به وسیله‌ی آن بتوانیم به حساب خود پول واریز کنیم. پس از موفقیت‌آمیز بودن عملیات واریز نیز پیام مناسب چاپ شود.
- یک متد به نام `save` داشته باشد که مقدار پول و درصد پولی که می‌خواهد از آن پول ذخیره کند، به کاربر داده شود و آن پول را به ویژگی `saving` اضافه کند.
- یک متد به نام `withdraw` داشته باشد که به وسیله‌ی آن بتوان از کیف پول برداشت کرد.

برای مثال، اگر تابع `add` را صدا بزنیم:

```
1 | int main() {
2 |     Wallet mywal(1000,0);
3 |     mywal.add(100);
4 | }
```

Copy ++C

ورودی اول مقدار پول اولیه است که به جای `total` قرار می‌گیرد. مقدار پس‌انداز اولیه را صفر در نظر بگیرید. به عنوان خروجی خواهیم داشت:

100 added!

Copy

تابع `withdraw` وظیفه‌ی برداشت از کیف پول شما را بر عهده دارد.

```
1 | int main() {
2 |     Wallet mywal(1000,0);
3 |     mywal.withdraw(4000);
4 |     mywal.withdraw(100);
5 | }
```

Copy ++C

این تابع ابتدا چک می‌کند اگر مقداری که می‌خواهید برداشت کنید بیشتر از موجودی باشد، در خروجی عبارت `low account balance` را چاپ می‌کند. درغیراین‌صورت، به اندازه‌ی داده‌شده از کل پول (`total`) برداشت می‌کند و در نهایت `withdrawn successfully` را چاپ می‌کند.

مثلا خروجی تابع بالا به صورت زیر است:

```
low account balance
100 withdrawn successfully
```

Copy

تابع `account_balance` مقدار `total` را برمی‌گرداند.

```
1 | int main() {
2 |     Wallet mywal(1000,0);
3 |     mywal.withdraw(100);
4 |     cout<< mywal.account_balance();
5 | }
```

Copy ++C

در خروجی خواهیم داشت:

```
100 withdrawn successfully  
900
```

[Copy](#)

تابع `save` مقداری از پول داده‌شده که به عنوان اولین ورودی می‌گیرد را به ویژگی `save` اضافه می‌کند.

```
1 | int main() {  
2 |     Wallet mywal(1000,0);  
3 |     mywal.save(4000,0.1);  
4 |     cout<< mywal.account_balance();  
5 | }
```

[Copy](#) ++C

در خروجی خواهیم داشت:

```
400.000000 saved!  
4600
```

[Copy](#)