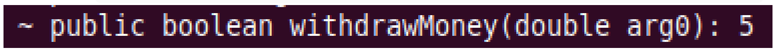
Ex 1 :

I added a column about the class size / class responsability

Class	LOC (approx.)	NOM	Short description of responsibility	Size/Responsability
Bank	393	14	Manage bank user accounts and get the average/min/max balance	Yes
BankAccount	405	20	Get/set infos about a bank account and its owner	Yes
Person	294	23	Get/Set infos about a person	No: Too much methods and attributes regarding the class responsability in the application (Like Get/Set eye color...)
BankAccountApp	447	2	Lauches the app	Yes

Ex 2 :

1. I chose `BankAccount.withdrawMoney(double withdrawAmount)` and its **cyclomatic complexity** value is 5 :



```
~ public boolean withdrawMoney(double arg0): 5
```

```
public boolean withdrawMoney(double withdrawAmount) {
    if (withdrawAmount >= 0 && balance >= withdrawAmount && withdrawAmount < withdrawLimit
          && withdrawAmount + amountWithdrawn <= withdrawLimit) { // On this line, we have 1 "if" and 3 "&&"
       balance = balance - withdrawAmount;
       success = true;
       amountWithdrawn += withdrawAmount;
    } else { // 1 "else"
       success = false;
    }
    return success;
}
// 1 + 3 + 1 = 5 CC
```

2. I would extract :

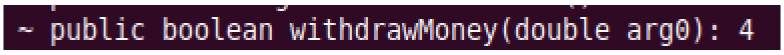
```
if (withdrawAmount >= 0 && balance >= withdrawAmount && withdrawAmount < withdrawLimit
       && withdrawAmount + amountWithdrawn <= withdrawLimit)
```

To create a `private boolean isWithdrawPossible(double withdrawAmount)
and get :

```
public boolean withdrawMoney(double withdrawAmount) {
    if isWithdrawPossible(withdrawAmount) {
       balance = balance - withdrawAmount;
       success = true;
       amountWithdrawn += withdrawAmount;
    } else {
        success = false;
    }
    return success;
}
```

So CC would be 2, I could even just return `true` instead of assigning it to `success` and waiting end of method to return it.
But the helper function is getting the old complexity so it is useless, the only choice is to refactor and get rid of `&& withdrawAmount < withdrawLimit` as the next check is `withdrawAmount + amountWithdrawn <= withdrawLimit`

3.



```
~ public boolean withdrawMoney(double arg0): 4
```

```
public boolean withdrawMoney(double withdrawAmount) {
    if (withdrawAmount >= 0 && balance >= withdrawAmount
          && withdrawAmount + amountWithdrawn <= withdrawLimit) {
        balance = balance - withdrawAmount;
        amountWithdrawn += withdrawAmount;
        return true;
    }
    return false;
}
```