

LAB 02 - Inheritance, Abstraction, and Interface

Bank Accounts

I am providing the following interface which contains the abstract methods that should be implemented in an abstract class `BankAccount`.

```
interface BankAccountInterface {
    double getBalance();
    String getName();
    void deposit(double amount) throws Exception;
    void withdraw(double amount) throws Exception;
    void transfer(double amount, BankAccount destination) throws
Exception;
}
```

The abstract class `BankAccount` should look like this:

```
abstract class BankAccount {
    private String name;
    private double balance;

    public BankAccount(String name, double initialDeposit)
throws Exception {
        // WRITE YOUR CODE HERE
        // If the name does not have more than 2 characters
        // the constructor should throw an Exception
        // Also, if the initialDeposit is negative or equal
        // to zero, it should throw an Exception
    }

    // With the same logic, throw an Exception in the deposit
    // method if the amount is negative or equal to zero.
    // Think and implement a condition that could throw an
    // exception in the methods: withdrawal and transfer.
}
```

Implement the class `CheckingAccount`.

Hint: you just need to implement the constructor.

It is going to use all the methods of `BankAccount` without having to override them.

```
class CheckingAccount extends BankAccount {  
  
}
```

Implement a class `SavingsAccount`, which differs from `CheckingAccount` for having an interest rate and a possibility to withdraw only a certain number of times (let's set this value at 6 times maximum). Nowadays this withdrawal restriction is no longer applied by banks, but I am setting this restriction in the problem to make this lab more challenging. Note that there are not going to be different periods where we reset this instance variable `withdrawCount`, which means when `withdrawCount` equals to 6, we cannot withdraw anymore.

```
class SavingsAccount extends BankAccount {  
    private double interestRate;  
    private int withdrawCount;  
  
    public SavingsAccount(String name, double initialDeposit,  
double interestRate) throws Exception {  
        // implement the constructor  
    }  
  
    public void addInterest() throws Exception {  
        // WRITE YOUR CODE HERE  
    }  
  
    public int getWithdrawCount() {  
        // WRITE YOUR CODE HERE  
    }  
}
```

Implement a class `SavingsAccountChild`. This account has a restriction to withdraw money: a parent/guardian should perform this transaction. The name of the parent/guardian should match the instance variable `parent_name` in order to process the transaction.

```

class SavingsAccountKid extends SavingsAccount{

    private String parentName;

    public SavingsAccountKid(String name, double initialDeposit,
double interestRate, String parentName) throws Exception {
        // implement the constructor
    }

    public void withdraw(double amount, String parentName)
throws Exception {
        // WRITE YOUR CODE HERE
        // Check if the parameter parentName is equals to the
        // instance variable parentName before withdrawing from
        // the account
    }

}

```

You must implement all the classes. I do not want to see repeated code between the classes, this is a lab based on inheritance. I named all the methods you will need to implement in the classes. You should decide when you will need to override a method in a subclass.

You must create a `Driver`, and test all the methods in your classes, trigger and catch all the exceptions, including exceptions in the `Driver`. Print descriptive messages about the actions performed by your program.