LAB 04 - Inheritance, Abstraction, Exceptions

Bank Accounts

I am providing the following **interface** which contains the 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`** must look like this:

```
// This class should implement the BankAccountInterface
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 at least 3 characters
        // the constructor must throw an Exception
        // Also, if the initialDeposit is negative or equal
        // to zero, it must throw an Exception
    }

       // With the same logic, throw an Exception in methods:
       // deposit, withdrawal, and transfer. The amount for
       // for these methods should always be greater than zero
}
```

Create a class **`CheckingAccount`.**
Hint: You just have to implement the constructor.
It will use all the Bank Account methods without having to override them.

```
class CheckingAccount extends BankAccount {

}
```

Create 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 will not 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
    }
}
```

Create 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 `parentName` in order to process the transaction.

```
class SavingsAccountChild extends SavingsAccount{

    private String parentName; // min length 3 characters

    public SavingsAccountChild(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
     }

}
```

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 implemented in your classes. Trigger and catch all the exceptions, including exceptions in the `Driver`. Print descriptive messages about the actions performed by your program.