

Nama : Yuni Lestari

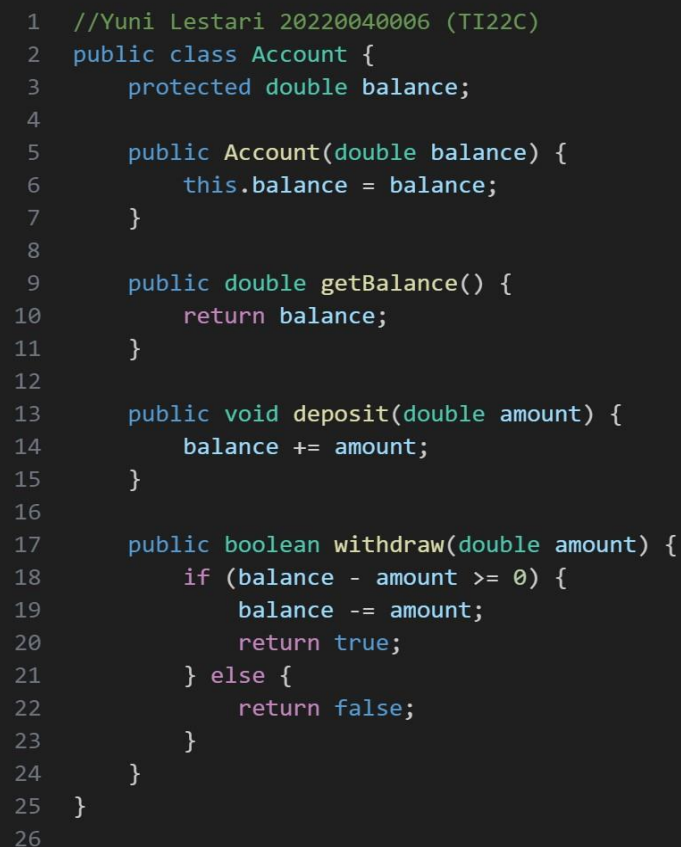
NIM : 20220040006

Kelas : TI22C

Mata Kuliah : Pemrograman Berorientasi Object Sesi 3

ANALISA SETIAP PERCOBAAN PRAKTIKUM

1. Account.java

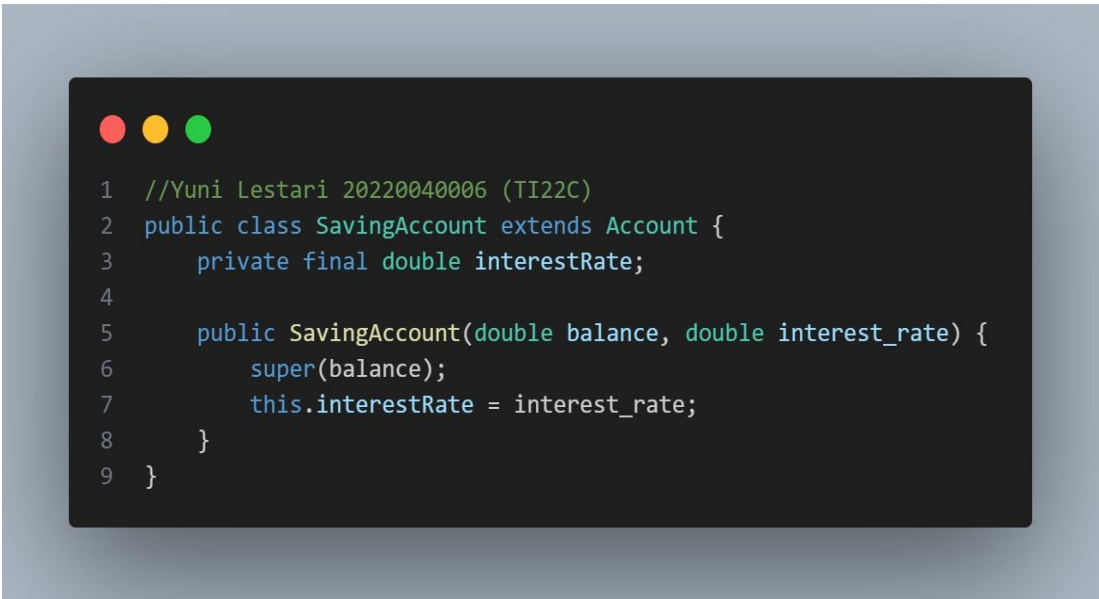


```
1 //Yuni Lestari 20220040006 (TI22C)
2 public class Account {
3     protected double balance;
4
5     public Account(double balance) {
6         this.balance = balance;
7     }
8
9     public double getBalance() {
10         return balance;
11     }
12
13     public void deposit(double amount) {
14         balance += amount;
15     }
16
17     public boolean withdraw(double amount) {
18         if (balance - amount >= 0) {
19             balance -= amount;
20             return true;
21         } else {
22             return false;
23         }
24     }
25 }
26
```

Account.java merupakan kelas dasar yang mewakili akun umum dengan saldo, dan menyediakan fungsi dasar seperti deposit dan withdraw.

- Kelas ini memiliki atribut balance yang diatur sebagai protected, yang berarti hanya kelas turunan yang dapat mengaksesnya secara langsung
- Konstruktor menerima parameter saldo awal dan menginisialisasi saldo
- Method getBalance() mengembalikan nilai saldo.
- Method deposit(double amount) menambahkan jumlah ke saldo.
- Method withdraw(double amount) mencoba untuk menarik jumlah tertentu dari saldo, mengembalikan true jika penarikan berhasil dan false jika tidak.

2. SavingAccount.java



```
1 //Yuni Lestari 20220040006 (TI22C)
2 public class SavingAccount extends Account {
3     private final double interestRate;
4
5     public SavingAccount(double balance, double interest_rate) {
6         super(balance);
7         this.interestRate = interest_rate;
8     }
9 }
```

SavingAccount.java merupakan kelas turunan dari Account yang mewakili akun tabungan dengan tambahan atribut tingkat bunga.

- Kelas ini memiliki atribut tambahan interestRate.
- Konstruktor menerima saldo awal dan tingkat bunga, kemudian memanggil konstruktor kelas induk dengan menggunakan super (balance).
- Tidak ada tambahan method pada kelas ini karena hanya memiliki atribut tambahan dan mewarisi fungsi dari kelas induk.

3. CheckingAccount.java

```
1 //Yuni Lestari 20220040006 (TI22C)
2 public class CheckingAccount extends Account {
3     private double overdraftProtection;
4
5     public CheckingAccount(double balance, double protect) {
6         super(balance);
7         this.overdraftProtection = protect;
8     }
9
10    public CheckingAccount(double balance) {
11        this(balance, -1.0); // Default overdraftProtection is -1.0
12        this.balance += overdraftProtection;
13    }
14
15    @Override
16    public boolean withdraw(double amount) {
17        if (balance - amount >= 0) {
18            balance -= amount;
19            return true;
20        } else {
21            if (overdraftProtection == -1.0 || overdraftProtection < (amount - balance)) {
22                return false;
23            } else {
24                balance = 0.0;
25                overdraftProtection -= (amount - balance);
26                return true;
27            }
28        }
29    }
30 }
31
```

CheckingAccount.java merupakan kelas turunan dari account yang mewakili akun check dengan tambahan fitur proteksi overdraft.

- Kelas ini mewakili atribut tambahan overdraftProtection
- Ada dua konstruktor yaitu, satu menerima saldo awal dan jumlah proteksi overdraft, dan yang lain menerima hanya saldo awal dengan proteksi overdraft default.
- Method withdraw (double amount) dioverride untuk menangani logika penarikan khusus akun check, yang memeriksa saldo dan proteksi overdraft.

4. Main.java

```
1 //Yuni Lestari 20220040006 (TI22C)
2 public class Main {
3     public static void main(String[] args) {
4         // Testing Account class
5         Account account = new Account(1000);
6         System.out.println("Saldo: " + account.getBalance());
7         account.deposit(500);
8         System.out.println("Saldo setelah deposit: " + account.getBalance());
9         account.withdraw(200);
10        System.out.println("Saldo setelah penarikan: " + account.getBalance());
11
12        // Testing SavingAccount class
13        SavingAccount savingAccount = new SavingAccount(2000, 0.05);
14        System.out.println("\nSaldo Rekening Tabungan: " + savingAccount.getBalance());
15
16        // Testing CheckingAccount class
17        CheckingAccount checkingAccount = new CheckingAccount(3000, 1000);
18        System.out.println("\nSaldo Rekening Checking: " + checkingAccount.getBalance());
19        checkingAccount.withdraw(4000);
20        System.out.println("Saldo setelah penarikan: " + checkingAccount.getBalance());
21    }
22 }
23
```

Main.java merupakan tempat pengujian dilakukan untuk semua kelas.

- Membuat objek dari kelas Account, SavingAccount dan CheckingAccount untuk menguji fungsionalitas mereka.
- Menggunakan method yang telah didefinisikan untuk memeriksa deposit, penarikan dan saldo setelah operasi tersebut dilakukan.

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
[Running] cd "d:\Praktikum-Inheritance-main\src\main\java\" && javac Main.java && java Main
Saldo: 1000.0
Saldo setelah deposit: 1500.0
Saldo setelah penarikan: 1300.0

Saldo Rekening Tabungan: 2000.0

Saldo Rekening Checking: 3000.0
Saldo setelah penarikan: 0.0

[Done] exited with code=0 in 1.471 seconds
```