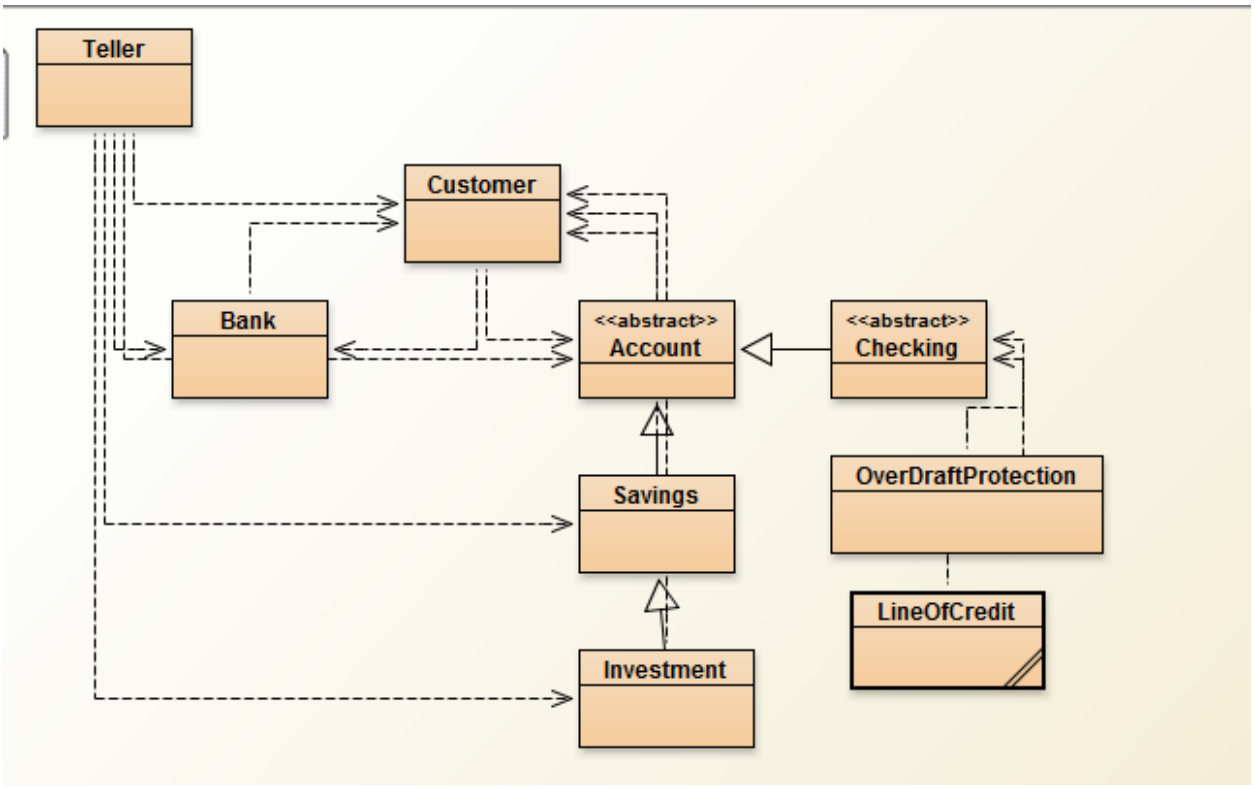
Tugas 4. Multilevel Inheritance pada Kelas Abstrak

1.



2.

```
public abstract class Checking extends Account
{
    protected double MontlyFee;
     private int x;
```

3.

```
protected double MontlyFee;
protected abstract void feeAssessment()
{
    return null;
 }
```

5.

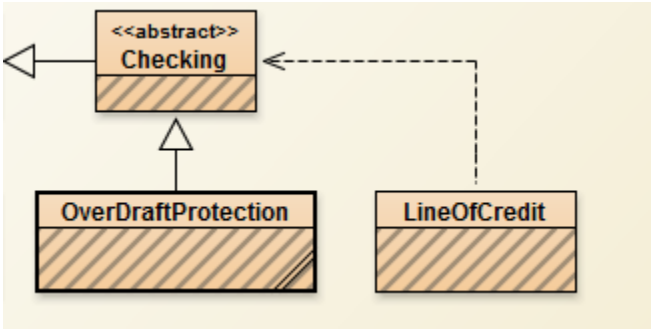
```
public double resetMonthlyFee()
{
    if(balance-amount < 0) {
        return false;
    }
    else {
        balance -= amount;
        return true;
    }
}
```

6.

```
public double getMonthlyFree()
{
    return monthlyFee;
}
```

Tugas 5. Membuat Kelas dengan Inheritance dari Kelas Abstrak 2

1.



2.

```
private double Savings savingsAccount;
```

5.

```
@Override
public abstract boolean withdraw(double amount);
{

    for(i = 0; i < accounts.length; i++) {
        if(accounts[i] != null && accounts[i].getId().endsWith(Character.toString(type))) {
            accounts[i] = null;
            accountRemoved = true;
            numOfAccounts--;
        }
        else {
            accountRemoved = false;
        }
    }
    return accountRemoved;
}
```

Tugas 6. Membuat Kelas dengan Inheritance dari Kelas Abstrak 3

1.

```
public class LineOfCredit extends Checking
{
    // instance variables - replace the example below with your ow
    private int x;

    /**
     * Constructor for objects of class LineOfCredit
     */
    public LineOfCredit()
    {
        // initialise instance variables
        x = 0;
    }
```

2.

```
private double creditBalance;
private double creditLimit;
```

3.

4.

```
public double getCreditBalance()
{
    return creditBalance;
}
public double getCreditLimit()
{
    return creditLimit;
}
public double setCreditBalance()
{
    return creditBalance;
}
public double setCreditLimit()
{
    return creditLimit;
}
```

5.

```
public double getCreditBalance()
{
    return creditBalance;
}
public double getCreditLimit()
{
    return creditLimit;
}
public double setCreditBalance()
{
    return creditBalance;
}
public double setCreditLimit()
{
    return creditLimit;
}
```