//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Generic Bank Account Class

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

public class BankAccount {

final int MONTHS = 12 ;

protected int deposits, // number of deposits made for the month

withdraws ; // number of withdraws made for the month

protected double balance, // balance of account

interestRate, // annual interest rate

serveCharge ; // monthly service charges

// BankAccount constructor

// Accepts start balance and starting rate parameters

public BankAccount(double startBal, double startRate) {

deposits = 0 ;

withdraws = 0 ;

serveCharge = 0 ;

balance = startBal;

interestRate = startRate ;

} // end constructor

// function: deposit

// pre: must include amount that is to be deposited

// post: adds amount to balance and increments number of deposits for the month

public void deposit(double depo){

balance += depo ;

deposits++ ;

} // end deposit

// function: withdraw

// pre: must include amount that is to be withdrawn

// post: subtracts amount from balance and increments number of withdraws for the month

public void withdraw(double draw){

balance -= draw ;

withdraws++ ;

} // end withdraw

// function: calcInterest

// pre: none

// post: adds interest earned for the month to the account

public void calcInterest() {

double monthRate = interestRate / MONTHS ; // calculates interest rate for month

double interest = balance \* monthRate ; // calculates interest earned

balance = balance + interest ; // adds interest to balance

} // end calcInterest

// function: monthlyProcess

// pre: none

// post: deducts service charge from account, adds interest to account, and zeros out deposits, withdraws, and service charges

public void monthlyProcess() {

balance -= serveCharge ; // deducts service charge from account

this.calcInterest() ; // adds interest to account

deposits = 0 ; // zeros out deposits, withdraws, and service charges

withdraws = 0 ;

serveCharge = 0 ;

} // end monthlyProcess

} // end class bankAccount

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Chris Dang CSCI 1125

// Checking Account Class

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

public class SavingsAccount extends BankAccount {

final int MINIMUM = 25 ; // MINIMUM amount for active to stay active is $25

final int CHARGE = 1 ;

final int MAX\_DRAW = 4 ;

protected boolean active ;

// SavingsAccount constructor

// Accepts start balance and starting rate parameters

public SavingsAccount(double startBal, double startRate) {

super(startBal, startRate) ;

if (startBal >= MINIMUM)

active = true ;

else

active = false ;

} // end SavingsAccount constructor

// function: deposit

// pre: must include amount that is to be deposited

// post: if new balance brings total to > MINIMUM, reactivates account, and deposits money

public void deposit(double depo){

double newSum = depo + balance ; // calculate new sum

if (!active && newSum > MINIMUM) // reactivates account if

active = true ; // account is not active and

// new sum is > MINIMUM

super.deposit(depo); // deposits with parent class deposit

} // end deposit

// function: withdraw

// pre: must include amount that is to be withdrawn

// post: if active, subtracts amount from balance and if balance falls below MINIMUM, deactivates account

public void withdraw(double draw){

if (active) { // no withdrawal is made if not active

super.withdraw(draw); // withdraws with parent class withdraw

if (balance < MINIMUM) // checks if balance falls below MINIMUM

active = false ; // deactivates account if balance < MINIMUM

} // end if

} // end withdraw

// function: monthlyProcess

// pre: none

// post: CHARGEs if too many withdraws, processes account, and disables if amount falls below MINIMUM

public void monthlyProcess() {

if (withdraws > MAX\_DRAW) { // if more than MAX\_DRAW number of withdraws

int deduct = CHARGE \* (withdraws - MAX\_DRAW) ;

balance -= deduct ; // deduct a dollar for each withdraw above MAX\_DRAW

} // end if

super.monthlyProcess();

if (balance < MINIMUM) // checks if balance falls below MINIMUM

active = false ; // deactivates account if balance < MINIMUM

} // end monthlyProcess

} // end SavingsAccount

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Checking Account Class

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

import java.awt.\* ;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\* ;

public class CheckingAccount extends BankAccount{

final int CHARGE = 15 ; // service CHARGE for overdraw

final int FEE = 5 ; // monthly FEE

final double FEE\_RATE = 0.10 ; // CHARGEs 10 cents

// CheckingAccount constructor

// Accepts start balance and starting rate parameters

public CheckingAccount(double startBal, double startRate) {

super(startBal, startRate) ;

} // end constructor

// function: withdraw

// pre: must include amount that is to be withdrawn

// post: CHARGEs if account if overdrawn and denies withdrawal, otherwise withdraws

public void withdraw(double draw){

double newSum = balance - draw ; // calculates how much account if withdrawn

if (newSum < 0) // if negative,

balance -= CHARGE ; // apply CHARGE and deny withdraw

else

super.withdraw(draw) ; // else, withdraw

} // end withdraw

// function: monthlyProcess

// pre: none

// post: CHARGEs if too many withdraws, processes account

public void monthlyProcess() {

serveCharge = FEE + (FEE\_RATE \* withdraws) ; // calculates FEE total and applies to monthly FEE

super.monthlyProcess() ; // processes account

} // end monthlyProcess

} // end CheckingAccount

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Generic Bank Account GUI

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

import java.awt.\* ;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\* ;

public class BankAccountGUI {

final int EX\_BAL = 25 ; // example balance and rate to start construction

final int EX\_RATE = 10 ;

final int WIDTH = 230 ;

final int HEIGHT = 350 ;

// constructs new bank, savings, and checking accounts

BankAccount bank = new BankAccount(EX\_BAL, EX\_RATE) ;

//SavingsAccount save = new SavingsAccount(EX\_BAL, EX\_RATE) ;

//CheckingAccount check = new CheckingAccount(EX\_BAL, EX\_RATE) ;

// Frame of Banking

JFrame frame = new JFrame("Java Banking") ;

SavingsPanel save = new SavingsPanel() ;

CheckingPanel check = new CheckingPanel() ;

JTabbedPane tabPane = new JTabbedPane() ;

BankAccountGUI() {

// closes account

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setPreferredSize(new Dimension(WIDTH, HEIGHT));

save.setBackground(Color.cyan) ;

check.setBackground(Color.pink) ;

// // sets background

// JPanel back = new JPanel() ;

// back.setBackground(Color.gray) ;

// back.setPreferredSize(new Dimension(WIDTH, HEIGHT));

//

//

//

tabPane.addTab("Savings", save) ;

tabPane.addTab("Checking", check) ;

// frame.getContentPane().add(back) ;

frame.getContentPane().add(tabPane) ;

frame.pack() ;

frame.setVisible(true) ;

} // end BankAccountGUI

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Generic Bank Account GUI

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

import java.awt.\* ;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\* ;

public class BankAccountGUI {

final int EX\_BAL = 25 ; // example balance and rate to start construction

final int EX\_RATE = 10 ;

final int WIDTH = 230 ;

final int HEIGHT = 350 ;

// constructs new bank, savings, and checking accounts

BankAccount bank = new BankAccount(EX\_BAL, EX\_RATE) ;

//SavingsAccount save = new SavingsAccount(EX\_BAL, EX\_RATE) ;

//CheckingAccount check = new CheckingAccount(EX\_BAL, EX\_RATE) ;

// Frame of Banking

JFrame frame = new JFrame("Java Banking") ;

SavingsPanel save = new SavingsPanel() ;

CheckingPanel check = new CheckingPanel() ;

JTabbedPane tabPane = new JTabbedPane() ;

BankAccountGUI() {

// closes account

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setPreferredSize(new Dimension(WIDTH, HEIGHT));

save.setBackground(Color.cyan) ;

check.setBackground(Color.pink) ;

// // sets background

// JPanel back = new JPanel() ;

// back.setBackground(Color.gray) ;

// back.setPreferredSize(new Dimension(WIDTH, HEIGHT));

//

//

//

tabPane.addTab("Savings", save) ;

tabPane.addTab("Checking", check) ;

// frame.getContentPane().add(back) ;

frame.getContentPane().add(tabPane) ;

frame.pack() ;

frame.setVisible(true) ;

} // end BankAccountGUI

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Generic Bank Account GUI

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

import java.awt.\* ;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\* ;

public class BankAccountGUI {

final int EX\_BAL = 25 ; // example balance and rate to start construction

final int EX\_RATE = 10 ;

final int WIDTH = 230 ;

final int HEIGHT = 350 ;

// constructs new bank, savings, and checking accounts

BankAccount bank = new BankAccount(EX\_BAL, EX\_RATE) ;

//SavingsAccount save = new SavingsAccount(EX\_BAL, EX\_RATE) ;

//CheckingAccount check = new CheckingAccount(EX\_BAL, EX\_RATE) ;

// Frame of Banking

JFrame frame = new JFrame("Java Banking") ;

SavingsPanel save = new SavingsPanel() ;

CheckingPanel check = new CheckingPanel() ;

JTabbedPane tabPane = new JTabbedPane() ;

BankAccountGUI() {

// closes account

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setPreferredSize(new Dimension(WIDTH, HEIGHT));

save.setBackground(Color.cyan) ;

check.setBackground(Color.pink) ;

// // sets background

// JPanel back = new JPanel() ;

// back.setBackground(Color.gray) ;

// back.setPreferredSize(new Dimension(WIDTH, HEIGHT));

//

//

//

tabPane.addTab("Savings", save) ;

tabPane.addTab("Checking", check) ;

// frame.getContentPane().add(back) ;

frame.getContentPane().add(tabPane) ;

frame.pack() ;

frame.setVisible(true) ;

} // end BankAccountGUI

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Chris Dang CSCI 1125

// Generic Bank Account GUI

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package javaProg4;

import java.awt.\* ;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\* ;

public class BankAccountGUI {

final int EX\_BAL = 25 ; // example balance and rate to start construction

final int EX\_RATE = 10 ;

final int WIDTH = 230 ;

final int HEIGHT = 350 ;

// constructs new bank, savings, and checking accounts

BankAccount bank = new BankAccount(EX\_BAL, EX\_RATE) ;

//SavingsAccount save = new SavingsAccount(EX\_BAL, EX\_RATE) ;

//CheckingAccount check = new CheckingAccount(EX\_BAL, EX\_RATE) ;

// Frame of Banking

JFrame frame = new JFrame("Java Banking") ;

SavingsPanel save = new SavingsPanel() ;

CheckingPanel check = new CheckingPanel() ;

JTabbedPane tabPane = new JTabbedPane() ;

BankAccountGUI() {

// closes account

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setPreferredSize(new Dimension(WIDTH, HEIGHT));

save.setBackground(Color.cyan) ;

check.setBackground(Color.pink) ;

// // sets background

// JPanel back = new JPanel() ;

// back.setBackground(Color.gray) ;

// back.setPreferredSize(new Dimension(WIDTH, HEIGHT));

//

//

//

tabPane.addTab("Savings", save) ;

tabPane.addTab("Checking", check) ;

// frame.getContentPane().add(back) ;

frame.getContentPane().add(tabPane) ;

frame.pack() ;

frame.setVisible(true) ;

} // end BankAccountGUI

}

