



Soluções exercícios da semana



Solução Slot Machine

Arquivos anexados:  [slotMachine.zip](#) (22,025 KB)



5.2 Solution to the Bank account

```
package bank;

public class Account {
    int balance;

    public Account(int balance) {
        this.balance = balance;
    }

    public synchronized void deposit(int amount) {
        balance += amount;
    }

    public synchronized boolean withdraw(int amount) {
        if (amount < balance) {
            balance -= amount;
            return true;
        } else {
            return false;
        }
    }

    public synchronized int getBalance() {
        return balance;
    }
}

package bank;

import java.util.Random;

public class Depositor extends Thread {
    private Account account;
    int fundsDeposited;

    public Depositor(Account account_to_deposit_to) {
        this.account = account_to_deposit_to;
        this.fundsDeposited = 0;
    }
}
```

```

    public void run() {
        while (!Thread.interrupted()) {
            int nextDeposit = (new Random()).nextInt(10) + 1;
            System.out.println("Depositing: " + nextDeposit);
            account.deposit(nextDeposit);
            fundsDeposited += nextDeposit;
        }
    }

    public int getFundsDeposited() {
        return fundsDeposited;
    }
}

package bank;

import java.util.Random;

public class Withdrawer extends Thread {
    private Account account;
    int fundsWithdrawn;

    public Withdrawer(Account account_to_withdraw_from) {
        this.account = account_to_withdraw_from;
        this.fundsWithdrawn = 0;
    }

    public void run() {
        while (!Thread.interrupted()) {
            int nextWithdrawalTry = (new Random()).nextInt(10) + 1;
            if (account.withdraw(nextWithdrawalTry)) {
                System.out.println("Succeeded withdrawing: " + nextWithdrawalTry);
                fundsWithdrawn += nextWithdrawalTry;
            } else {
                System.out.println("Failed to withdraw: " + nextWithdrawalTry);
            }
        }
    }

    public int getFundsWithdrawn() {
        return fundsWithdrawn;
    }
}

package bank;

public class Bank {

    public static void main(String[] args) throws InterruptedException {
        int initialFunds = 1000;

        Account account = new Account(initialFunds);
        Depositor depositor = new Depositor(account);
        Withdrawer withdrawer = new Withdrawer(account);

        depositor.start();
        withdrawer.start();

        Thread.sleep(10 * 1000);

        depositor.interrupt();
        withdrawer.interrupt();
    }
}

```

```
depositor.join();
withdrawer.join();

System.out.println("Initial funds: " + initialFunds);
System.out.println("Total withdrawn: " + (withdrawer.getFundsWithdrawn()));
System.out.println("Total deposited: " + depositor.getFundsDeposited());

int computedBalance = initialFunds - withdrawer.getFundsWithdrawn() + depositor.getFundsDeposited();

System.out.println("Balance should be: " + computedBalance);
System.out.println("Balance is: " + account.getBalance());

if (computedBalance != account.getBalance()) {
    System.out.println("Something is wrong! The two balances are not the same!");
} else {
    System.out.println("All fine - the balances are the same.");
}
}
}
```



Solução exercício 5.3