# 代码8

GitHub地址：https://github.com/HuaZhouyang/Course\_JavaProgramming

———————————————————————————————————————

## Unit 11:

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

### AdditionQuiz.java:

package Unit\_11;

import java.util.Scanner;

import java.util.TreeSet;

/\*\*

\* 11.16

\*/

public class AdditionQuiz {

public static void main(String[] args) {

int number1 = (int)(Math.random() \* 10);

int number2 = (int)(Math.random() \* 10);

// Create a Scanner

Scanner input = new Scanner(System.in);

System.out.print("What is " + number1 + " + " + number2 + "? ");

TreeSet<Integer> set = new TreeSet<>();

int answer = input.nextInt();

set.add(answer);

while (number1 + number2 != answer) {

System.out.print("Wrong answer. Try again. What is "

+ number1 + " + " + number2 + "? ");

answer = input.nextInt();

if (set.contains(answer)) {

System.out.println("You already entered " + answer);

}

}

System.out.println("You got it!");

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

### BinPackingUsingFirstFit.java:

package Unit\_11;

import java.util.ArrayList;

import java.util.Scanner;

/\*\*

\* 11.19

\*/

public class BinPackingUsingFirstFit {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter the number of objects: ");

int n = sc.nextInt(), i;

ArrayList<Integer> weights = new ArrayList<>();

ArrayList<Integer> sizes = new ArrayList<>();

System.out.print("Enter the weights of the objects: ");

for (i = 0; i < n; i++) {

weights.add(sc.nextInt());

}

i = 1;

while (!weights.isEmpty()) {

sizes.clear();

int size = 0;

for (Integer weight : weights) {

if (size + weight <= 10) {

size += weight;

sizes.add(weight);

}

}

System.out.print("Container " + i++ + " contains objects with weight");

for (Integer weight : sizes) {

System.out.print(" " + weight);

weights.remove(weight);

}

System.out.println();

}

}

}

/\*

Enter the number of objects: 6

Enter the weights of the objects: 7 5 2 3 5 8

Container 1 contains objects with weight 7 2

Container 2 contains objects with weight 5 3

Container 3 contains objects with weight 5

Container 4 contains objects with weight 8

\*/

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

### NewAccount.java:

package Unit\_11;

import java.util.ArrayList;

import java.util.Date;

/\*\*

\* 11.8

\*/

public class NewAccount extends Account {

public static void main(String[] args) {

NewAccount.setAnnualInterestRate(1.5);

NewAccount georgeAccount = new NewAccount(1122, 1000, "George");

georgeAccount.deposit(30);

georgeAccount.deposit(40);

georgeAccount.deposit(50);

georgeAccount.withdraw(5);

georgeAccount.withdraw(4);

georgeAccount.withdraw(2);

georgeAccount.printAccountSummary();

}

private String name;

private ArrayList<Transaction> transactions = new ArrayList<>();

public NewAccount(int id, double balance, String name) {

super(id, balance);

this.name = name;

}

@Override

public void withdraw(double amount) {

super.withdraw(amount);

transactions.add(new Transaction('W', amount, getBalance(), "从账户提取"+amount+"元"));

}

@Override

public void deposit(double amount) {

super.deposit(amount);

transactions.add(new Transaction('D', amount, getBalance(), "向账户存储"+amount+"元"));

}

public void printAccountSummary() {

System.out.println("Name: " + name);

System.out.println("Annual Interest Rate: " + getAnnualInterestRate() + "%%");

System.out.println("Balance: " + getBalance());

System.out.println("All Transactions: ");

for (Transaction transaction : transactions) {

System.out.println("\t" + transaction.toString());

}

}

}

class Transaction {

private Date date; // 该交易的日期

private char type; // 交易类型

private double amount; // 交易量

private double balance; // 交易后的新余额

private String description; // 交易的描述

public Transaction(char type, double amount, double balance, String description) {

this.date = new Date();

this.type = type;

this.amount = amount;

this.balance = balance;

this.description = description;

}

@Override

public String toString() {

return "Transaction{" +

"date=" + date +

", type=" + type +

", amount=" + amount +

", balance=" + balance +

", description='" + description + '\'' +

'}';

}

}

class Account {

private int id = 0; // 标识账号

private double balance = 0; // 余额

private static double annualInterestRate = 0; // 当前利率

private Date dateCreated; // 账户开户日期

public Account() {

}

public Account(int id, double balance) {

this.id = id;

this.balance = balance;

this.dateCreated = new Date();

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public double getBalance() {

return balance;

}

public void setBalance(double balance) {

this.balance = balance;

}

public static double getAnnualInterestRate() {

return annualInterestRate;

}

public static void setAnnualInterestRate(double annualInterestRate) {

Account.annualInterestRate = annualInterestRate;

}

public Date getDateCreated() {

return dateCreated;

}

// 返回月利率

public double getMonthlyInterestRate() {

return annualInterestRate / 100 / 12;

}

// 返回月利息

public double getMonthlyInterest() {

return balance \* getMonthlyInterestRate();

}

// 取款

public void withdraw(double amount) {

if (this.balance >= amount) this.balance -= amount;

}

// 存款

public void deposit(double amount) {

this.balance += amount;

}

}

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

### PerfectSquare.java:

package Unit\_11;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

/\*\*

\* 11.17

\*/

public class PerfectSquare {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int m, n = 1;

System.out.print("Enter an integer m: ");

m = sc.nextInt();

int tmp = m;

Map<Integer, Integer> map = new HashMap<>();

int i = 2;

while (m != 1) {

if (m % i == 0) {

m = m / i;

if (map.containsKey(i)) {

map.put(i, map.get(i) + 1);

} else {

map.put(i, 1);

}

} else {

i++;

}

}

for (i = 0; i < 1000; i++) {

if (map.containsKey(i) && map.get(i) % 2 == 1)

n = n \* i;

}

System.out.println("The smallest number n for m \* n to be a perfect square is " + n);

System.out.println("m \* n is " + (tmp \* n));

}

}