学号 E21514033 专业 软件工程 姓名 何铭春

实验日期 **2018.5.10**  教师签字 成绩

实验报告

【实验名称】 Java实验四

【实验目的】

1. 熟悉处理异常的方法
2. 了解如何自定义异常类

【实验原理】

1. 异常的分离
2. 自定义异常类需要定义getMessage方法

【实验内容】

题目一：股票模拟

源代码：

public class Stock {

private String symbol;

private String name;

private double previousClosingPrice;

private double currentPrice;

public Stock(){}

public Stock(String symbol,String name) {

this.symbol = symbol;

this.name = name;

}

public double getChangePercent(){

double temp = (currentPrice - previousClosingPrice) / previousClosingPrice;

return temp \* 100;

}

public void setPre(double previousClosingPrice){

this.previousClosingPrice = previousClosingPrice;

}

public void setCurrent(double currentPrice) {

this.currentPrice = currentPrice;

}

}

class Test {

public static void main(String[] args) {

Stock hmc = new Stock("ORCL","Oracle Corporation");

hmc.setPre(34.5);

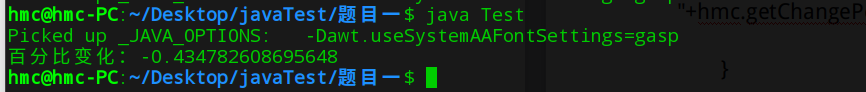
hmc.setCurrent(34.35);

System.out.println("百分比变化："+hmc.getChangePercent());

}

}

实验截图：



题目二：正n边形

源代码：

public class RegularPolygon {

private int n;

private double side;

private double x;

private double y;

public RegularPolygon(){

n = 3;

side = 1;

x = 0;

y = 0;

}

public RegularPolygon(int n,double side){

this.n = n;

this.side = side;

x = 0;

y = 0;

}

public RegularPolygon(int n,double side,double x,double y){

this.n = n;

this.side = side;

this.x = x;

this.y = y;

}

public double getPerimeter(){

return n \* side;

}

public double getArea(){

double temp = n \* side \* side;

temp = temp / (4 \* Math.tan(Math.PI / n));

return temp;

}

}

class Test {

public static void show(RegularPolygon spring){

System.out.println("周长:"+spring.getPerimeter());

System.out.println("面积:"+spring.getArea());

}

public static void main(String[] args) {

RegularPolygon hmc1 = new RegularPolygon();

RegularPolygon hmc2 = new RegularPolygon(6,4);

RegularPolygon hmc3 = new RegularPolygon(10,4,5.6,7.8);

show(hmc1);

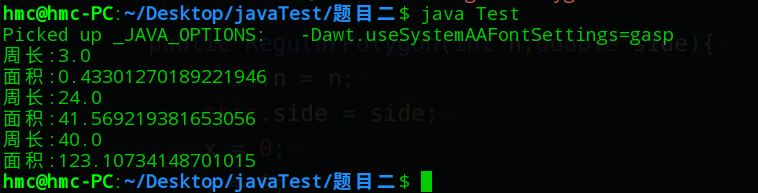
show(hmc2);

show(hmc3);

}

}

实验截图：



题目三：账户管理

源代码：

public class Account {

private int id;

private double balance;

private double annualInterestRate;

public Account() {

id = 0;

balance = 0;

annualInterestRate = 0;

}

public Account(int id,double balance) {

this.id = id;

this.balance = balance;

annualInterestRate = 0;

}

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 double getRate(){

return annualInterestRate;

}

public void setRate(double annualInterestRate){

this.annualInterestRate = annualInterestRate;

}

public double getMonthlyInterestRate(){

return (annualInterestRate / 100) / 12;

}

public String withDraw(double money){

if (money > balance) {

return "余额不足";

}

else {

balance = balance - money;

return "成功取钱";

}

}

public void deposit(double money){

balance = balance + money;

}

}

class Test {

public static void main(String[] args) {

Account hmc = new Account(1122, 20000);

hmc.setRate(4.5);

System.out.println(hmc.withDraw(2500));

hmc.deposit(3000);

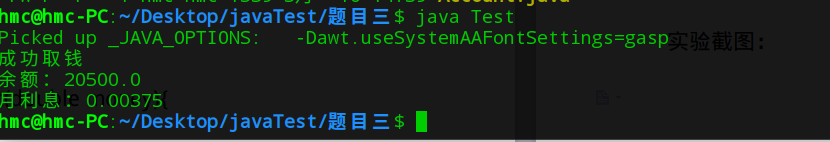
System.out.println("余额："+hmc.getBalance());

System.out.println("月利息："+hmc.getMonthlyInterestRate());

}

}

实验截图：



题目四：二次方程式

源代码：package src;

import java.util.\*;

public class QuadraticEquation {

private double a;

private double b;

private double c;

public QuadraticEquation(){

}

public QuadraticEquation(double a,double b,double c){

this.a = a;

this.b = b;

this.c = c;

}

public double getA(){

return a;

}

public double getB(){

return b;

}

public double getC(){

return c;

}

public double getDiscriminant(){

return b \* b - 4 \* a \* c;

}

public double getRoot1(){

double temp = b \* b - 4 \* a \* c;

if (temp >= 0 ) {

temp = Math.sqrt(temp);

temp = temp - b;

return temp / (2 \* a);

}

else {

return 0;

}

}

public double getRoot2(){

double temp = b \* b - 4 \* a \* c;

if (temp >= 0) {

temp = Math.sqrt(temp);

temp = (- b) - temp;

return temp / (2 \* a);

}

else {

return 0;

}

}

}

测试类：

package main;

import java.util.\*;

class Test {

public static void main(String[] args) {

System.out.println("请输入三个数：");

Scanner in = new Scanner(System.in);

double a = in.nextDouble();

double b = in.nextDouble();

double c = in.nextDouble();

in.close();

src.QuadraticEquation hmc = new src.QuadraticEquation(a,b,c);

double dis = hmc.getDiscriminant();

System.out.println("判别式结果:"+dis);

if (dis<0) {

System.out.println("The equation has no roots");

}

else if (dis==0) {

System.out.println("单根为:"+hmc.getRoot1());

}

else {

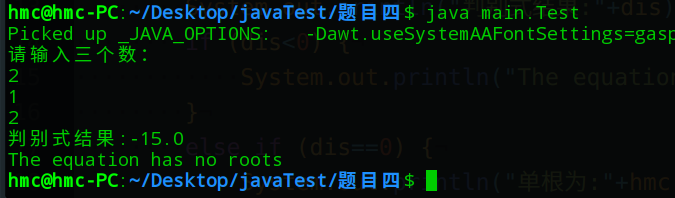
System.out.println("两个根为"+hmc.getRoot1()+"和"+hmc.getRoot2());

}

}

}

实验截图：



题目五：浮点数分离整数和小数

源代码：

import java.util.\*;

public class Juggle {

public static int show() {

System.out.println("请输入一个浮点数：");

String hmc;

Scanner in = new Scanner(System.in);

hmc = in.nextLine();

in.close();

double spring;

try {

spring = Double.parseDouble(hmc);

} catch(Exception e) {

System.out.println("请输入正确的浮点数");

return 0;

}

String[] temp = hmc.split("\\.");

int ll = hmc.length() - 1;

System.out.println("浮点数有"+ll+"位");

System.out.println("整数部分是"+temp[0]);

System.out.println("小数部分是"+temp[1]);

System.out.println("整数部分位数:"+temp[0].length());

System.out.println("小数部分位数:"+temp[1].length());

return 1;

}

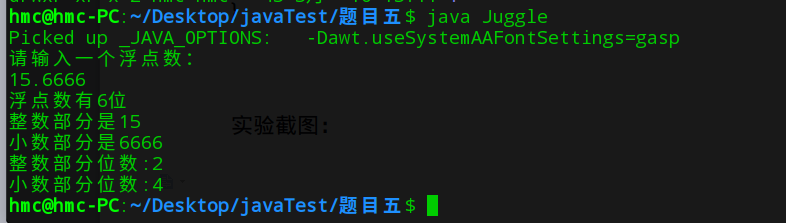
public static void main(String[] args) {

show();

}

}

实验截图：



题目六：异常处理

源代码：

import java.util.\*;

public class Test {

public static void main(String[] args) {

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

System.out.println("请输入5个整数：");

Scanner in = new Scanner(System.in);

try {

String temp = in.nextLine();

hmc.add(Integer.parseInt(temp));

temp = in.nextLine();

hmc.add(Integer.parseInt(temp));

temp = in.nextLine();

hmc.add(Integer.parseInt(temp));

temp = in.nextLine();

hmc.add(Integer.parseInt(temp));

temp = in.nextLine();

hmc.add(Integer.parseInt(temp));

} catch(NumberFormatException e) {

System.out.println("请输入整数！");

}

in.close();

try {

int out;

for (int i = 0; i<5; i++) {

out = i + 1;

System.out.println("第"+out+"个数:"+hmc.get(i));

}

} catch(ArrayIndexOutOfBoundException e) {

System.out.println("请输入至少五个整数！");

}

}

}

实验截图：



题目七：自定义异常类

源代码：

import java.util.\*;

public class Test{

public static void numberException(int x) throws Exception{

if (x > 100) {

throw new NumberTooBigException();

}

else if (x < 0) {

throw new NumberTooSmallException();

}

else {

System.out.println("没有发生异常");

}

}

public static void main(String[] args) throws Exception {

Scanner in = new Scanner(System.in);

int x = in.nextInt();

in.close();

numberException(x);

}

}

class NumberTooBigException extends Exception {

public String getMessage(){

return "发生数字太大异常";

}

}

class NumberTooSmallException extends Exception {

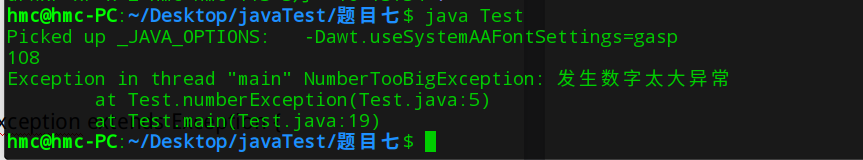
public String getMessage(){

return "发生数字太小异常";

}

}

实验截图：



【小结或讨论】

1. 第五题，浮点数分离整数和小数。一开始企图寻找Java自身方法判断。后来，直接使用字符串来分离整数部分的小数部分。
2. 自定义异常类需要getMessage方法，返回一个字符串来显示。