河北大学

本科生实验报告

《Java 编程入门实验》

装

订

线

学　　院　国际学院

专　　业　软件工程

学　　号　20242605031

姓　　名　石一泽

实验一 简单 Java 程序

一、实验目的

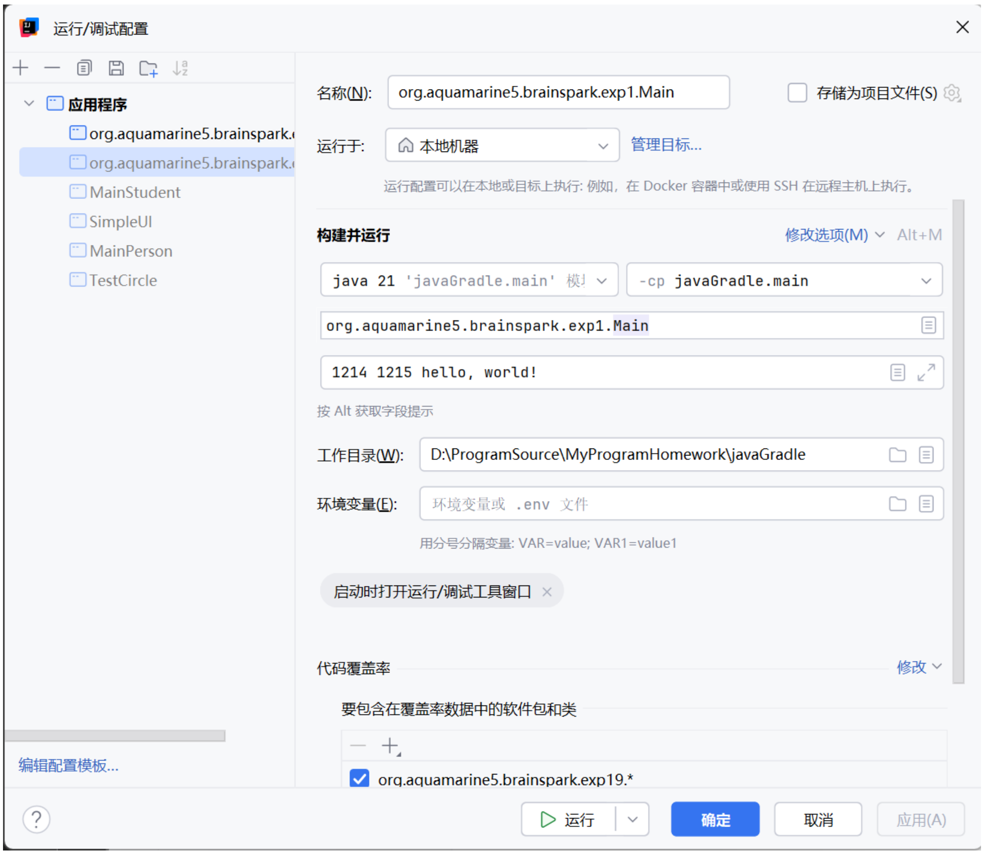
（1）掌握开发 Java 应用程序的三个步骤：编写源文件，编译源文件和运行应用程序。

二、实验内容和步骤

（1）编写一个简单的 Java 应用程序。

```java
package org.aquamarine5.brainspark.exp1;

public class Main {
    public static void main(String[] args) {
        if(args.length==0)
            System.out.println("Hello!");
        else
            for (String arg : args) System.out.println(arg);
    }
}
```

三、实验结果



名称(N): org.aquamarine5.brainspark.exp1.Main

□ 存储为项目文件(S)

运行于: 本地机器

运行配置可以在本地或目标上执行: 例如，在 Docker 容器中或使用 SSH 在远程主机上执行。

构建并运行

```
java 21 'javaGradle.main' 模
-cp javaGradle.main
org.aquamarine5.brainspark.exp1.Main
1214 1215 hello, world!
```

按 Alt 获取字段提示

工作目录(W): D:\ProgramSource\MyProgramHomework\javaGradle

环境变量(E): 环境变量或 .env 文件

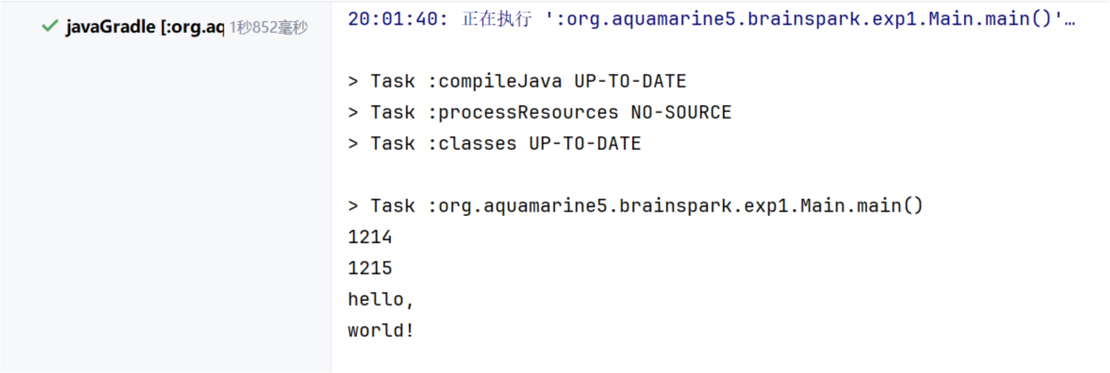
用分号分隔变量: VAR=value; VAR1=value1

启动时打开运行/调试工具窗口

代码覆盖率

要包含在覆盖率数据中的软件包和类

☑ org.aquamarine5.brainspark.exp19.*



```
20:01:40: 正在执行 ':org.aquamarine5.brainspark.exp1.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp1.Main.main()
1214
1215
hello,
world!
```

实验二　多类 Java 程序和包

一、实验目的

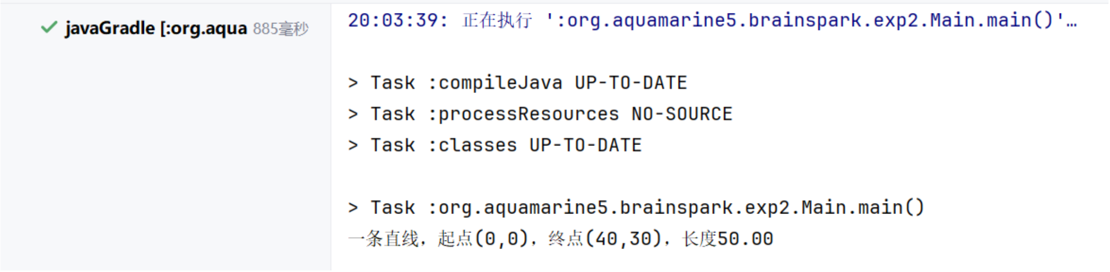
（1）　掌握开发 Java 应用程序的包

二、实验内容和步骤

（1）编写多类 Java 应用程序。

```java
package org.aquamarine5.brainspark.exp2.mypackage;

public class Line {
    public Point point1, point2;

    public Line(Point point1, Point point2)
    {
        this.point1 = point1;
        this.point2 = point2;
    }

    public double length()
    {
        int a = point1.x - point2.x, b = point1.y - point2.y;
        return Math.sqrt(a * a + b * b);
    }

    public String toString()
    {
        return "一条直线，起点" + point1.toString() + "，终点" +
point2.toString() + "，长度" + String.format("%1.2f", length());
    }
}
package org.aquamarine5.brainspark.exp2.mypackage;

public class Point
{
    public int x, y;

    public Point(int x,int y)
    {
        this.x=x;
        this.y=y;
    }
    public Point()
    {
        this(0,0);
    }

    public Point(Point p)
    {
        this(p.x,p.y);
    }

    public String toString()
    {
        return"("+this.x+","+this.y+")";
    }
}
package org.aquamarine5.brainspark.exp2;

import org.aquamarine5.brainspark.exp2.mypackage.Point;

import org.aquamarine5.brainspark.exp2.mypackage.Line;

public class Main {
    public static void main(String[] args) {
        var point1 = new Point(0,0);
        var point2=new Point(40,30);
        System.out.println(new Line(point1,point2).toString());
    }
}
```

三、实验结果



```
20:03:39: 正在执行 ':org.aquamarine5.brainspark.exp2.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp2.Main.main()
一条直线，起点(0,0)，终点(40,30)，长度50.00
```

实验三 Java 程序的输入和包的导入

一、实验目的

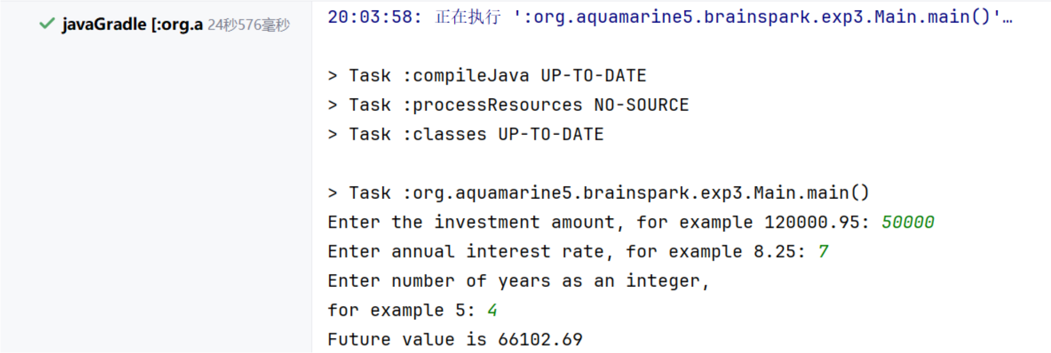
（1）掌握开发 Java 应用程序的输入和包的使用；

二、实验内容和步骤

（1）实现 Test 类，完成 Java 程序的输入和输入。

```java
package org.aquamarine5.brainspark.exp3;

import java.util.Scanner;

public class Main {
   public static void main(String[] args) {
      var input = new Scanner(System.in);
      System.out.print(
            "Enter the investment amount, for example 120000.95: ");
      double investmentAmount = input.nextDouble();
      System.out.print("Enter annual interest rate, for example 8.25:
");
      double annualInterestRate = input.nextDouble();
      double monthlyInterestRate = annualInterestRate / 1200;
      System.out.print(
            "Enter number of years as an integer, \nfor example 5:
");
      int numOfYears = input.nextInt();
      double futureValue =
            investmentAmount * Math.pow(1 + monthlyInterestRate,
                  numOfYears * 12);
      System.out.print("Future value is " +
            (int) (futureValue * 100) / 100.0);
   }
}
```

三、实验结果



```
20:03:58: 正在执行 ':org.aquamarine5.brainspark.exp3.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp3.Main.main()
Enter the investment amount, for example 120000.95: 50000
Enter annual interest rate, for example 8.25: 7
Enter number of years as an integer,
for example 5: 4
Future value is 66102.69
```

实验四　Java 程序分支语句

一、实验目的

（1）掌握开发 Java 应用程序的分支语句；

二、实验内容和步骤

（1）

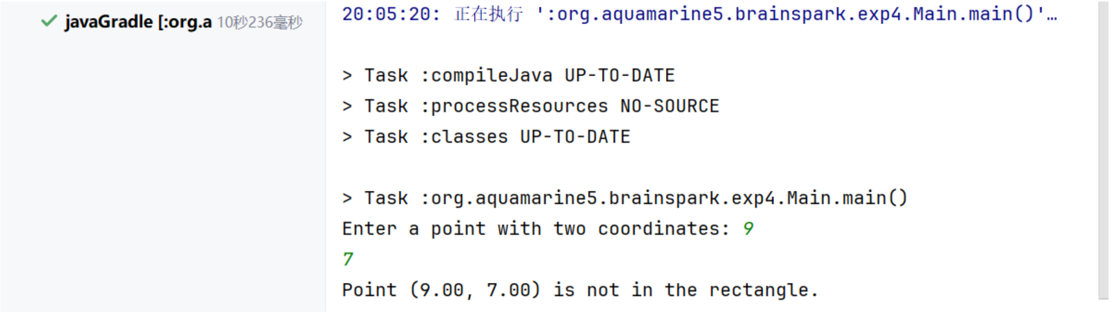
```java
package org.aquamarine5.brainspark.exp4;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        var input = new Scanner(System.in);
        System.out.print("Enter a point with two coordinates: ");
        double x = input.nextDouble();
        double y = input.nextDouble();
        double hDistance = Math.abs(x);
        double vDistance = Math.abs(y);
        System.out.println("Point (%.2f, %.2f) is ".formatted(x, y) + ((hDistance <= 5 && vDistance <= 2.5) ? "in" : "not in" + " the rectangle."));
    }
}
```

（2）

```java
package org.aquamarine5.brainspark.exp4;

public record MyDate(int year,int month,int day){
    private static int thisYear=2000;
    static {
        thisYear=3000;
    }

    @Override
    public String toString() {
        return "%d/%d/%d".formatted(year,month,day);
    }

    public static int getThisYear(){
        return thisYear;
    }
}
package org.aquamarine5.brainspark.exp4;

import java.util.Scanner;

public class MainMyDate {
    public static void main(String[] args) {
        MyDate date=new MyDate(2024,10,15);
        System.out.println(date);
        System.out.println(MyDate.getThisYear());
        System.out.println(date.getThisYear());
    }
}
```

三、实验结果

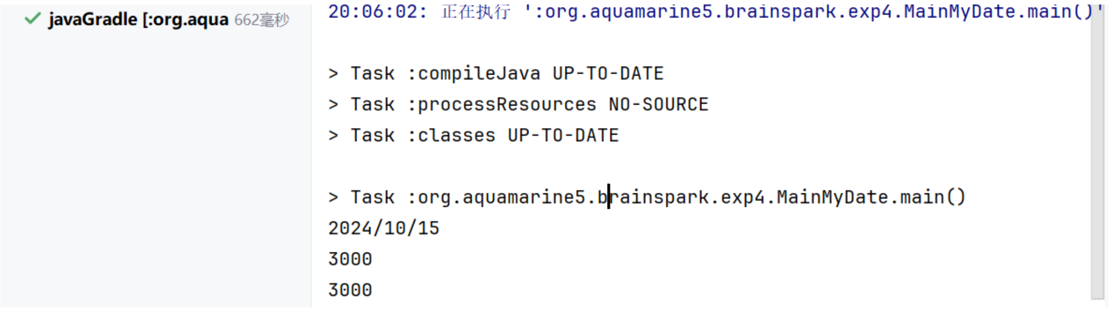
（1）



✓ javaGradle [:org.a 10秒236毫秒

```
20:05:20: 正在执行 ':org.aquamarine5.brainspark.exp4.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp4.Main.main()
Enter a point with two coordinates: 9
7
Point (9.00, 7.00) is not in the rectangle.
```

（2）



✓ javaGradle [:org.aqua 662毫秒

```
20:06:02: 正在执行 ':org.aquamarine5.brainspark.exp4.MainMyDate.main()'

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp4.MainMyDate.main()
2024/10/15
3000
3000
```

实验五　Java 程序 Switch 语句与格式化控制台输出

一、实验目的

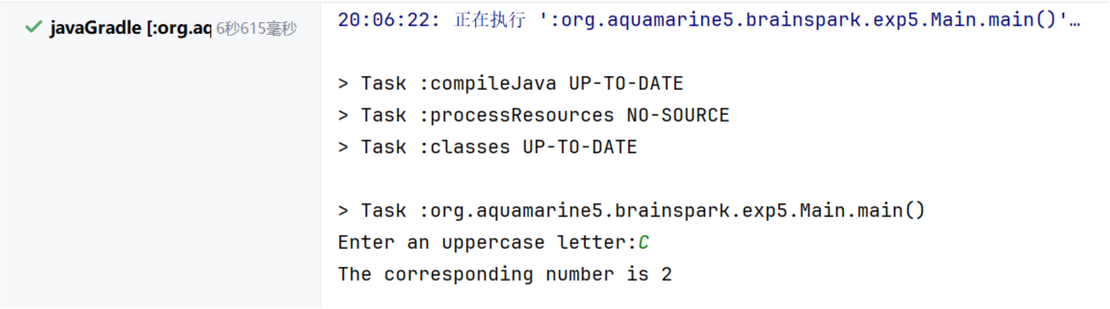
（1）掌握开发 Java 应用程序的分支语句和格式化输出；

二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp5;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        var input = new Scanner(System.in);
        System.out.print("Enter an uppercase letter:");
        char ch = input.next().charAt(0);
        int number = 0;
        switch (Character.toUpperCase(ch)) {
            case 'A':
            case 'B':
            case 'C':
                number = 2;
                break;
            case 'D':
            case 'E':
            case 'F':
                number = 3;
                break;
            case 'G':
            case 'H':
            case 'I':
                number = 4;
                break;
            case 'J':
            case 'K':
            case 'L':
                number = 5;
                break;
            case 'M':
            case 'N':
            case 'O':
                number = 6;
                break;
            case 'P':
            case 'Q':
            case 'R':
            case 'S':
                number = 7;
                break;
            case 'T':
            case 'U':
            case 'V':
                number = 8;
                break;
            case 'W':
            case 'X':
            case 'Y':
            case 'Z':
                number = 9;
                break;
            default:
                System.out.println(ch + "is an invalid input ");
                System.exit(1);
        }
        System.out.println("The corresponding number is " + number);
    }
}
```

三、实验结果



```
20:06:22: 正在执行 ':org.aquamarine5.brainspark.exp5.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp5.Main.main()
Enter an uppercase letter:C
The corresponding number is 2
```

实验六 Java 程序循环语句与输入输出重定向

一、实验目的

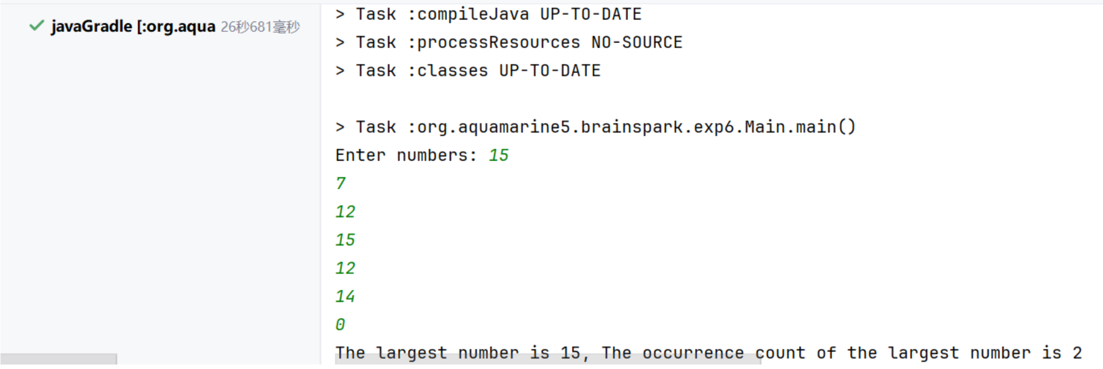
（1）掌握开发 Java 应用程序的循环语句和输入输出重定向；

二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp6;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        var input = new Scanner(System.in);
        System.out.print("Enter numbers: ");
        int number = input.nextInt();
        int max = number;
        int count = 1;
        while (number != 0) {
            number = input.nextInt();
            if (number > max) {
                max = number;
                count = 1;
            } else if (number == max)
                count++;
        }
        System.out.printf("The largest number is %d, The occurrence count of the largest number is %d%n", max, count);
    }
}
```

三、实验结果



```
> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp6.Main.main()
Enter numbers: 15
7
12
15
12
14
0
The largest number is 15, The occurrence count of the largest number is 2
```

实验七 Java 程序一维数组和循环语句

一、实验目的

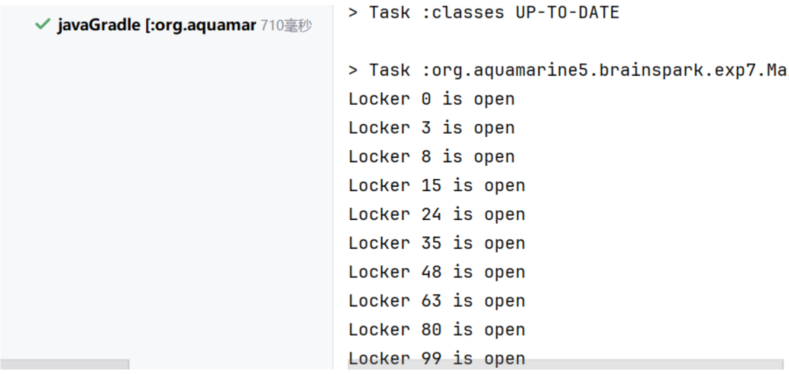
（1）掌握开发 Java 应用程序的一维数组；

二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp7;

public class Main {
    public static void main(String[] args) {
        final int NUMBER_OF_STUDENTS = 100;
        boolean[] lockers = new boolean[NUMBER_OF_STUDENTS];
        for (int student = 1; student <= NUMBER_OF_STUDENTS;
student++) {
            for (int locker = student - 1; locker < NUMBER_OF_STUDENTS;
locker += student) {
                lockers[locker] = !lockers[locker];
            }
        }
        for (int i = 0; i < lockers.length; i++) {
            if (lockers[i]) {
                System.out.printf("Locker %d is open\n", i);
            }
        }
    }
}
```

三、实验结果



```
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp7.Main.main()
Locker 0 is open
Locker 3 is open
Locker 8 is open
Locker 15 is open
Locker 24 is open
Locker 35 is open
Locker 48 is open
Locker 63 is open
Locker 80 is open
Locker 99 is open
```

实验八 Java 程序--类和对象

一、实验目的

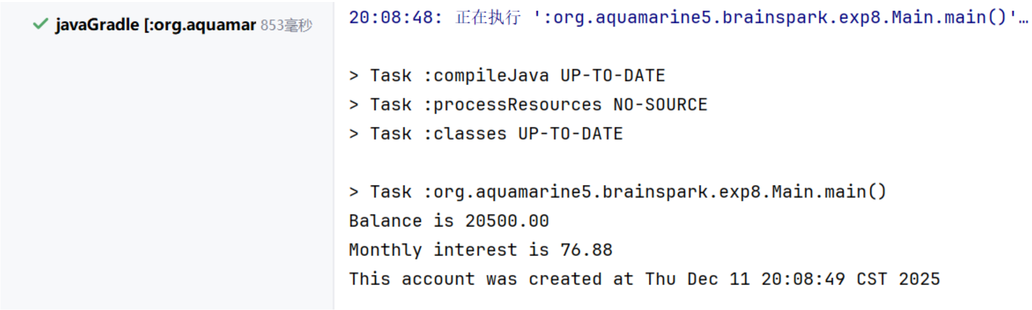
（1）掌握开发 Java 应用程序的一类和对象；

二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp8;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

import java.util.Date;

@AllArgsConstructor
@Getter
@Setter
public class Account {
    @Setter
    private static double annualInterestRate;
    private final Date dateCreated = new Date();
    private int id;
    private double balance;

    public double getMonthlyInterest() {
        return balance * annualInterestRate / 1200;
    }

    public void withdraw(double amount) {
        balance -= amount;
    }

    public void deposit(double amount) {
        balance += amount;
    }
}
package org.aquamarine5.brainspark.exp8;

public class Main {
    public static void main(String[] args){
        Account account=new Account(1122,20000);
        Account.setAnnualInterestRate(4.5);

        account.withdraw(2500);
        account.deposit(3000);

        System.out.printf("Balance is %.2f\n",account.getBalance());
        System.out.printf("Monthly interest 
is %.2f\n",account.getMonthlyInterest());
        System.out.println("This account was created at 
"+account.getDateCreated());
    }
}
```

三、实验结果



```
20:08:48: 正在执行 ':org.aquamarine5.brainspark.exp8.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp8.Main.main()
Balance is 20500.00
Monthly interest is 76.88
This account was created at Thu Dec 11 20:08:49 CST 2025
```

实验九 类和对象

一、实验目的

（1）掌握开发 Java 应用程序的一类和对象；

二、实验内容和步骤

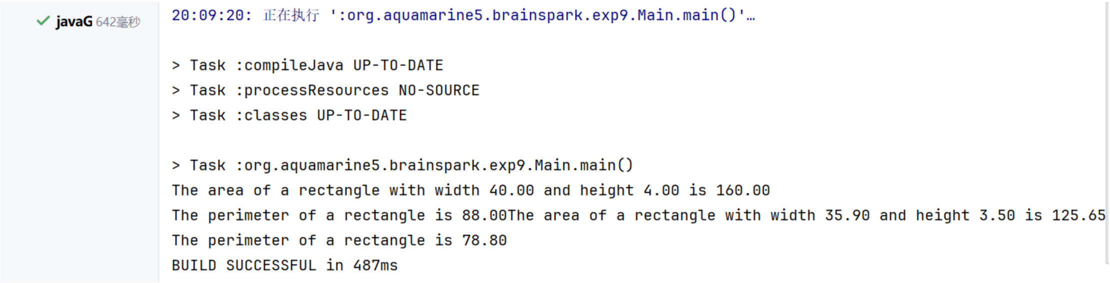
（1）

```
package org.aquamarine5.brainspark.exp9;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;

@AllArgsConstructor
@NoArgsConstructor
public class MyRectangle {
    @Getter
    double height=1;
    double width=1;

    public double getPerimeter(){
        return 2*(height+width);
    }
    public double getArea(){
        return height*width;
    }
}
package org.aquamarine5.brainspark.exp9;


public class Main {
    public static void main(String[] args) {
        var myRectangle = new MyRectangle(4d, 40d);
        System.out.printf("The area of a rectangle with width %.2f and
height %.2f is %.2f",
                myRectangle.width, myRectangle.height,
myRectangle.getArea());
        System.out.printf("\nThe perimeter of a rectangle
is %.2f",myRectangle.getPerimeter());
        var yourRectangle = new MyRectangle(3.5d, 35.9d);
        System.out.printf("The area of a rectangle with width %.2f and
height %.2f is %.2f",
                yourRectangle.width, yourRectangle.height,
yourRectangle.getArea());
        System.out.printf("\nThe perimeter of a rectangle is %.2f",
yourRectangle.getPerimeter());
    }
}
```

（2）

```
package org.aquamarine5.brainspark.exp9;

import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.Setter;

@Setter
@Getter
@EqualsAndHashCode
@AllArgsConstructor
public class MyDate {
    private static final int thisYear;

    static{
        thisYear=2025;
    }

    private int year = 1970;
    private int month = 1;
    private int day = 1;

    public void set(int year, int month, int day) {
        this.year = year;
        this.month = (month >= 1 && month <= 12) ? month : 1;
        this.day = (day >= 1 && day <= 31) ? day : 1;
    }

    public void set(MyDate date){
        this.year = date.year;
        this.month = date.month;
        this.day = date.day;
    }

    public String toString() {
        return String.format("%04d年%02d月%02d日", year, month, day);
    }

    public static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 ==
0);
    }

    public boolean isLeapYear() {
        return isLeapYear(this.year);
    }

    public static int daysOfMonth(int year,int month){
        return switch (month) {
            case 1, 3, 5, 7, 8, 10, 12 -> 31;
            case 4, 6, 9, 11 -> 30;
            case 2 -> isLeapYear(year) ? 29 : 28;
            default -> 0;
        };
    }

    public int daysOfMonth(){
        return daysOfMonth(this.year,this.month);
    }

    public void tomorrow(){
        if(this.day<daysOfMonth()){
            this.day++;
        }else{
            this.day=1;
            if(this.month<12){
                this.month++;
            }else{
                this.month=1;
                this.year++;
            }
        }
    }

    public void yesterday(){
        if(this.day>1){
            this.day--;
        }else{
            if(this.month>1){
                this.month--;
            }else{
                this.month=12;
                this.year--;
            }
            this.day=daysOfMonth();
        }
    }
}
```

三、实验结果



```
20:09:20: 正在执行 ':org.aquamarine5.brainspark.exp9.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp9.Main.main()
The area of a rectangle with width 40.00 and height 4.00 is 160.00
The perimeter of a rectangle is 88.00The area of a rectangle with width 35.90 and height 3.50 is 125.65
The perimeter of a rectangle is 78.80
BUILD SUCCESSFUL in 487ms
```

实验十 Java 程序--面向对象编程

一、实验目的

（1）掌握开发 Java 应用程序的一类和对象；

二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp10;

import lombok.*;

import java.util.Date;

@Getter
@NoArgsConstructor
@AllArgsConstructor
public class Loan {
    private final Date loanDate = new Date();
    @Setter
    private double annualInterestRate = 2.5;
    @Setter
    private int numberOfYears = 1;
    @Setter
    private double loanAmount = 1000;

    public double getMonthlyPayment() {
        double monthlyInterestRate = annualInterestRate / 1200;
        return loanAmount * monthlyInterestRate /
               (1 - 1 / Math.pow(1 + monthlyInterestRate,
numberOfYears * 12));
    }
    public double getTotalPayment() {
        return getMonthlyPayment() * numberOfYears * 12;
    }
}
```

（2）

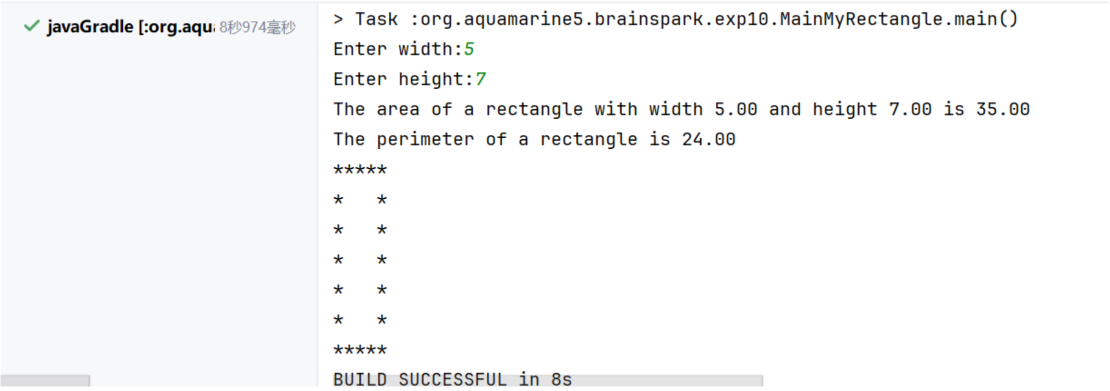
```
package org.aquamarine5.brainspark.exp10;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;

@AllArgsConstructor
@NoArgsConstructor
public class MyRectangle {
    @Getter
    double height=1;
    double width=1;

    public MyRectangle(double w){
        this.width=w;
        this.height=w;
    }

    public double getPerimeter(){
        return 2*(height+width);
    }
    public double getArea(){
        return height*width;
    }

    public void draw(){
        for (int i = 0; i < this.width; i++)
            System.out.print("*");
        System.out.println();
        for (int i = 0; i < this.height - 2; i++) {
            System.out.print("*");
            for (int j = 0; j < this.width-2; j++)
                System.out.print(" ");
            System.out.print("*");
            System.out.println();
        }
        for (int i = 0; i < this.width; i++)
            System.out.print("*");
    }
}
package org.aquamarine5.brainspark.exp10;

import lombok.AllArgsConstructor;

@AllArgsConstructor
public class Person {
    public String name;
    public int age;
    public void getInfo(){
        System.out.println("Name: " + name + ", Age: " + age);
    }
}
package org.aquamarine5.brainspark.exp10;


public class Teacher extends Person{
    public String teacherID;
    public Teacher(String name,int age,String id){
        super(name,age);
        this.teacherID=id;
    }

    public void getInfo(){
        System.out.println("Name: " + name + ", Age: " + age + ",
Teacher ID: " + teacherID);
    }
}
package org.aquamarine5.brainspark.exp10;

import java.util.Scanner;

public class MainMyRectangle {
    public static void main(String[] args){
        var scanner=new Scanner(System.in);
        System.out.print("Enter width:");
        double width=scanner.nextDouble();
        System.out.print("Enter height:");
        double height=scanner.nextDouble();
        var myRectangle=new MyRectangle(height,width);
        System.out.printf("The area of a rectangle with width %.2f and
height %.2f is %.2f",
                myRectangle.width, myRectangle.height,
myRectangle.getArea());
        System.out.printf("\nThe perimeter of a rectangle
is %.2f\n",myRectangle.getPerimeter());
        myRectangle.draw();
    }
}
```

（3）

```java
package org.aquamarine5.brainspark.exp10;

public class MainString {
    public static void main(String[] args){
        String str1 = "Hello";
        String str2 = "Hello";
        String str3 = new String("Hello");
        String str4 = new String("Hello");
        System.out.println(str1 == str2);      // true
        System.out.println(str1.equals(str2)); // true
        System.out.println(str1 == str3);      // false
        System.out.println(str1.equals(str3)); // true
        System.out.println(str3 == str4);      // false
    }
}
```

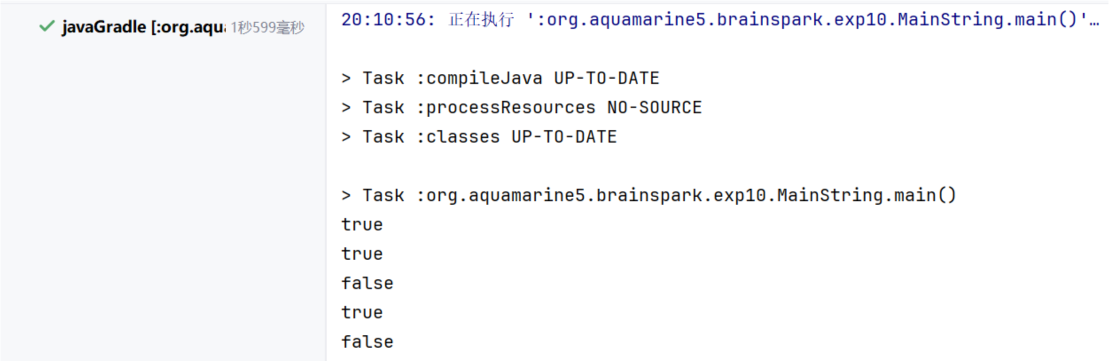
三、实验结果

（2）



```
> Task :org.aquamarine5.brainspark.exp10.MainMyRectangle.main()
Enter width:5
Enter height:7
The area of a rectangle with width 5.00 and height 7.00 is 35.00
The perimeter of a rectangle is 24.00
*****
*   *
*   *
*   *
*   *
*   *
*****
BUILD SUCCESSFUL in 8s
```

（3）



```
20:10:56: 正在执行 ':org.aquamarine5.brainspark.exp10.MainString.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp10.MainString.main()
true
true
false
true
false
```

实验十一 Java 程序--面向对象编程

一、实验目的

（1）掌握开发 Java 应用程序的一继承和多态；

二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp11;

import java.util.Date;

public abstract class GeometricObject {
    private final Date dateCreated;
    private String color = "white";
    private boolean filled;

    protected GeometricObject() {
        dateCreated = new Date();
    }

    protected GeometricObject(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
        dateCreated = new Date();
    }

    public String getColor() {
        return color;
    }

    public void setColor(String color) {
        this.color = color;
    }

    public boolean isFilled() {
        return filled;
    }

    public void setFilled(boolean filled) {
        this.filled = filled;
    }
    public Date getDateCreated() {
        return dateCreated;
    }

    @Override
    public String toString() {
        return "GeometricObject{" +
                "color='" + color + '\'' +
                ", filled=" + filled +
                ", dateCreated=" + dateCreated +
                '}';
    }

    public abstract double getArea();

    public abstract double getPerimeter();
}
package org.aquamarine5.brainspark.exp11;

public class Rectangle extends GeometricObject {
    private double width, height;

    public Rectangle() {
    }

    public Rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    public Rectangle(double width, double height, String color,
boolean filled) {
        super(color, filled);
        this.width = width;
        this.height = height;
    }

    public double getWidth() {
        return width;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public double getHeight() {
        return height;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    @Override
    public double getArea() {
        return width * height;
    }

    @Override
    public double getPerimeter() {
        return 2 * (width + height);
    }
}
package org.aquamarine5.brainspark.exp11;

public class Circle extends GeometricObject {
    private double radius;

    public Circle() {
    }

    public Circle(double radius) {
        this.radius = radius;
    }

    public Circle(double radius, String color, boolean filled) {
        super(color, filled);
        this.radius = radius;
    }

    public double getRadius() {
        return radius;
    }

    public void setRadius(double radius) {
        this.radius = radius;
    }
```

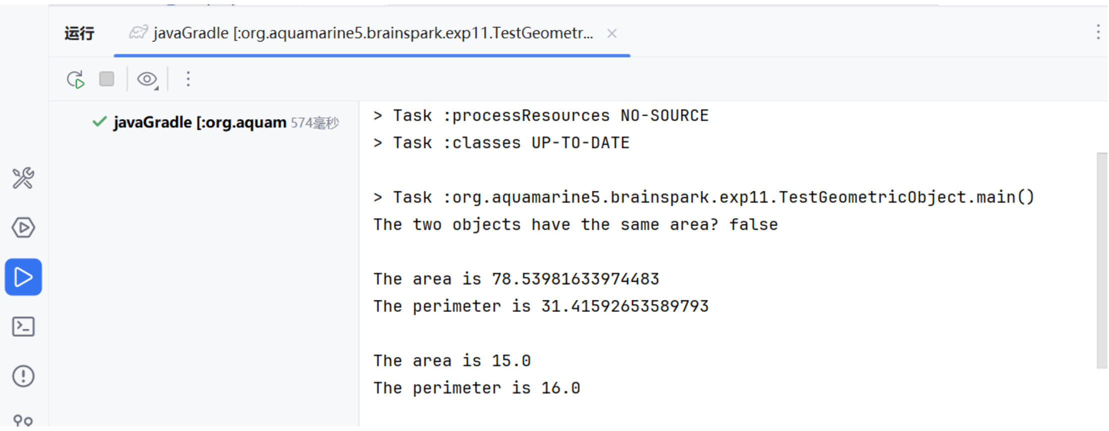
```java
    @Override
    public double getArea() {
        return radius * radius * Math.PI;
    }

    @Override
    public double getPerimeter() {
        return 2 * radius * Math.PI;
    }

    public double getDiameter() {
        return 2 * radius;
    }

    public void printCircle() {
        System.out.println("The circle is created " + getDateCreated()
+ " and the radius is " + radius);
    }
}
package org.aquamarine5.brainspark.exp11;


import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.NoArgsConstructor;
import lombok.Setter;

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
public class Triangle extends GeometricObject {
    private double side1 = 1.0, side2 = 1.0, side3 = 1.0;

    @Override
    public double getArea() {
        double s = (side1 + side2 + side3) / 2;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }

    @Override
    public double getPerimeter() {
        return side1 + side2 + side3;
    }

    @Override
    public String toString() {
        return "Triangle{" +
                "side1=" + side1 +
                ", side2=" + side2 +
                ", side3=" + side3 +
                "} ";
    }
}
package org.aquamarine5.brainspark.exp11;

public class TestGeometricObject {
    public static void main(String[] args) {
        GeometricObject geoObject1=new Circle(5);
        GeometricObject geoObject2=new Rectangle(5,3);
        System.out.println("The two objects have the same area?
"+equalArea(geoObject1,geoObject2));
        displayGeometricObject(geoObject1);
        displayGeometricObject(geoObject2);
    }

    public static boolean equalArea(GeometricObject object1,
GeometricObject object2){
        return object1.getArea()==object2.getArea();
    }
    public static void displayGeometricObject(GeometricObject
object){
        System.out.println();
        System.out.println("The area is "+object.getArea());
        System.out.println("The perimeter is "+object.getPerimeter());
    }
}
```

（2）

```
package org.aquamarine5.brainspark.exp11;

public class MainPerson {
    public static void main(String[] args) {
        A b = new B();
        b.m(5);
        System.out.println(b.i);
    }
```

```java
}

class A {
    int i = 1;

    public void m(int i) {
        this.i = i;
    }
}

class B extends A {
    // int i = 4;
    int j = 2;

    public void m(int i) {
        this.i = i;
    }
}
```

三、实验结果


运行
javaGradle [:org.aquamarine5.brainspark.exp11.TestGeometr...
javaGradle [:org.aquam 574毫秒

```
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp11.TestGeometricObject.main()
The two objects have the same area? false

The area is 78.53981633974483
The perimeter is 31.41592653589793

The area is 15.0
The perimeter is 16.0
```

实验十二 Java 程序--面向对象编程

一、实验目的

（1）掌握开发 Java 应用程序的一继承和多态；

二、实验内容和步骤

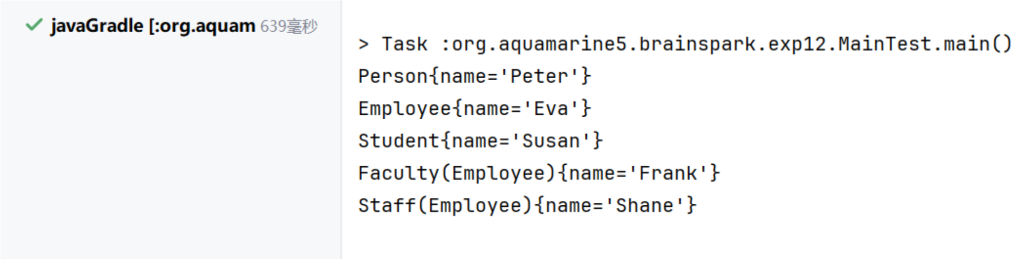
（1）

```java
package org.aquamarine5.brainspark.exp12;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

@AllArgsConstructor
@Getter
@Setter
public class Person {
    private String name;

    @Override
    public String toString() {
        return "Person{" +
                "name='" + name + "'}";
    }
}
```

```java
package org.aquamarine5.brainspark.exp12;

public class Employee extends Person{
    public Employee(String name) {
        super(name);
    }

    @Override
    public String toString() {
        return "Employee{" +
                "name='" + getName() + "'}";
    }
}
```

```java
package org.aquamarine5.brainspark.exp12;

public class Student extends Person{
    public Student(String name) {
        super(name);
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + getName() + "'}";
    }
}
package org.aquamarine5.brainspark.exp12;


public class Staff extends Employee{
    public Staff(String name) {
        super(name);
    }

    @Override
    public String toString() {
        return "Staff(Employee){" +
                "name='" + getName() + "'}";
    }
}
package org.aquamarine5.brainspark.exp12;

public class Faculty extends Employee{
    public Faculty(String name) {
        super(name);
    }

    public String toString() {
        return "Faculty(Employee){" +
                "name='" + getName() + "'}";
    }
}
package org.aquamarine5.brainspark.exp12;

public class MainTest {
    public static void main(String[] args) {
        Person person = new Person("Peter");
        Student student = new Student("Susan");
        Employee employee = new Employee("Eva");
        Faculty faculty = new Faculty("Frank");
        Staff staff = new Staff("Shane");

        System.out.println(person);
        System.out.println(employee);
        System.out.println(student);
        System.out.println((Person)faculty);
        System.out.println((Employee)staff);
    }
}
```

三、实验结果



```
> Task :org.aquamarine5.brainspark.exp12.MainTest.main()
Person{name='Peter'}
Employee{name='Eva'}
Student{name='Susan'}
Faculty(Employee){name='Frank'}
Staff(Employee){name='Shane'}
```

实验十三 Java 程序--面向对象编程

一、实验目的

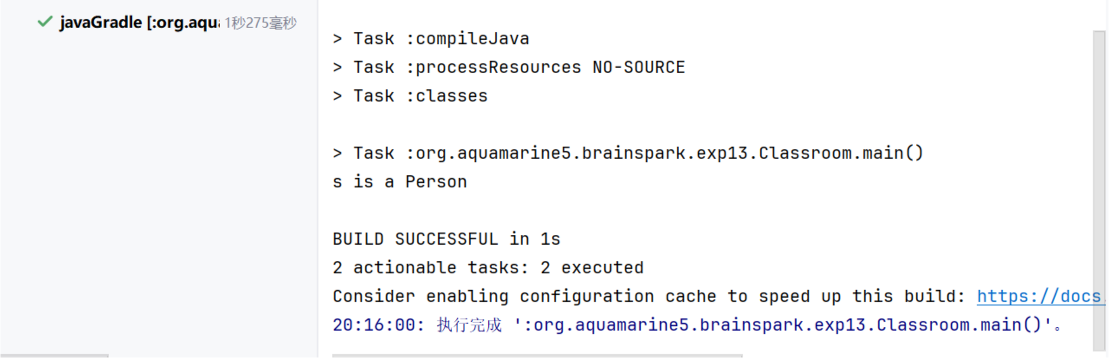
（1）掌握开发 Java 应用程序的一继承和多态；

二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp13;

import java.util.ArrayList;

public class Classroom {
    private final ArrayList<Person> people = new ArrayList<>();

    public void Main() {
        Person p = new Student("John", 150, 1);
        Student s = new Student("Alice", 200, 2);
        Student ps = (Student) p;
        if(s instanceof Person){
            System.out.println("s is a Person");
        }
    }
}
package org.aquamarine5.brainspark.exp13;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

@AllArgsConstructor
@Setter
@Getter
public class Person extends Object{
    private String name;
    protected int money;

    public void earnMoney(int amount) {
        this.money += amount;
    }

    protected final void spendMoney(int amount){
        this.money -= amount;
    }

    public static int getBasicMoney(){
        return 100;
    }
}
package org.aquamarine5.brainspark.exp13;

import lombok.Getter;

public class Student extends Person {
    @Getter
    private final int id;

    public Student(String name,int money, int id) {
        super(name,money);
        this.id = id;
    }

//    @Override
//    public void spendMoney(int amount) {
//        super.spendMoney(amount - 5);
//    }

    public void earnMoney(int amount) {
        var basic=getBasicMoney();
        super.earnMoney(amount + 10);
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + getName() + "'}";
    }

}
```

三、实验结果



```
> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes

> Task :org.aquamarine5.brainspark.exp13.Classroom.main()
s is a Person

BUILD SUCCESSFUL in 1s
2 actionable tasks: 2 executed
Consider enabling configuration cache to speed up this build: https://docs
20:16:00: 执行完成 ':org.aquamarine5.brainspark.exp13.Classroom.main()'。
```

实验十四 Java 程序--面向对象编程

一、实验目的

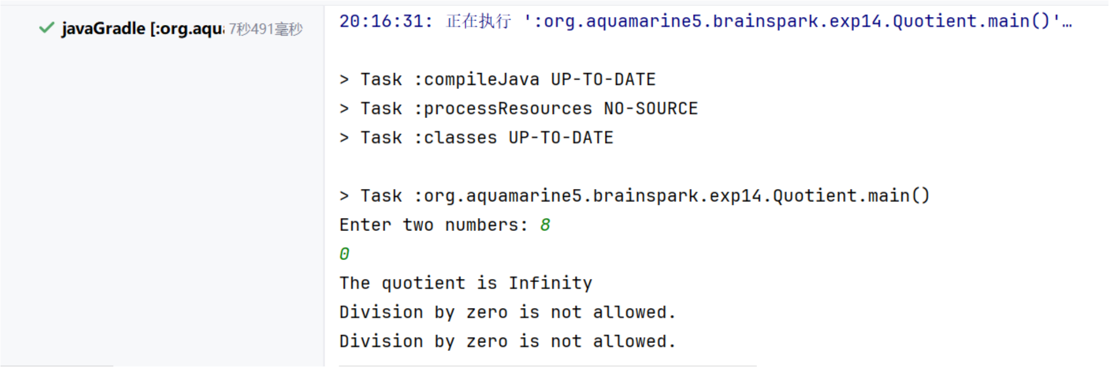
（1）掌握开发 Java 应用程序异常处理；

二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp14;

import java.util.Scanner;

public class Quotient {
    public static void quotient(double a, double b) {
        System.out.println("The quotient is " + (a / b));
    }

    public static void quotientIf(double a, double b) {
        if (b == 0)
            System.out.println("Division by zero is not allowed.");
        else System.out.println("The quotient is " + (a / b));
    }

    public static void quotientWithException(double a, double b)
            throws ArithmeticException {
        if (b == 0)
            throw new ArithmeticException("Division by zero is not
allowed.");
        else System.out.println("The quotient is " + (a / b));
    }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter two numbers: ");
        int a = scan.nextInt();
        int b = scan.nextInt();
        quotient(a, b);
        quotientIf(a, b);
        try {
            quotientWithException(a, b);
        } catch (ArithmeticException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

三、实验结果



✓ javaGradle [:org.aqu 7秒491毫秒

```
20:16:31: 正在执行 ':org.aquamarine5.brainspark.exp14.Quotient.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp14.Quotient.main()
Enter two numbers: 8
0
The quotient is Infinity
Division by zero is not allowed.
Division by zero is not allowed.
```

实验十五 Java 程序--面向对象编程

一、实验目的

（1）掌握开发 Java 应用程序异常处理；

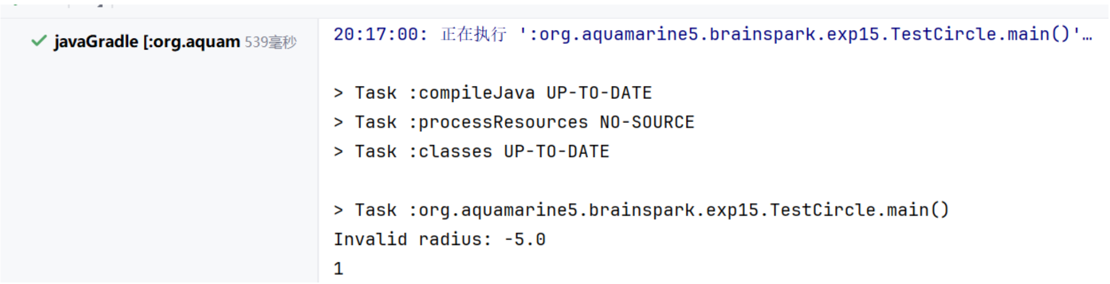
二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp15;

public class Rectangle {
    private double weight;
    private double height;
    private boolean square;


    public double getWeight() {
        return weight;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    public double getHeight() {
        return height;
    }

    public boolean isSquare() {
        return square;
    }
}
package org.aquamarine5.brainspark.exp15;

import lombok.Getter;

public class Circle {
    @Getter
    private double radius;
    @Getter
    private static int numberOfObjects = 0;
    public Circle(){
```

```java
        this(1.0);
    }
    public Circle(double radius){
        try{
            setRadius(radius);
            numberOfObjects++;
        } catch (InvalidRadiusException e){
            System.out.println(e.getMessage());
        }
    }
    public void setRadius(double radius)
            throws InvalidRadiusException{
        if(radius < 0)
            throw new InvalidRadiusException(radius);
        this.radius = radius;
    }
    public double findArea(){
        return radius * radius * Math.PI;
    }
}
package org.aquamarine5.brainspark.exp15;

import lombok.Getter;

public class InvalidRadiusException extends Exception {
    @Getter
    private double radius;

    public InvalidRadiusException(double radius) {
        super("Invalid radius: " + radius);
        this.radius = radius;
    }
}

package org.aquamarine5.brainspark.exp15;

public class TestCircle {
    public static void main(String[] args){
        try{
            Circle c1 = new Circle(5);
            c1.setRadius(-5);
            Circle c2 = new Circle(0);
        } catch (InvalidRadiusException e) {
            System.out.println(e.getMessage());
        }
        System.out.println(Circle.getNumberOfObjects());
    }
}
```

三、实验结果



```
20:17:00: 正在执行 ':org.aquamarine5.brainspark.exp15.TestCircle.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp15.TestCircle.main()
Invalid radius: -5.0
1
```

实验十六 Java 程序--面向对象编程

一、实验目的

（1）掌握开发 Java 应用程序--抽象类和接口；

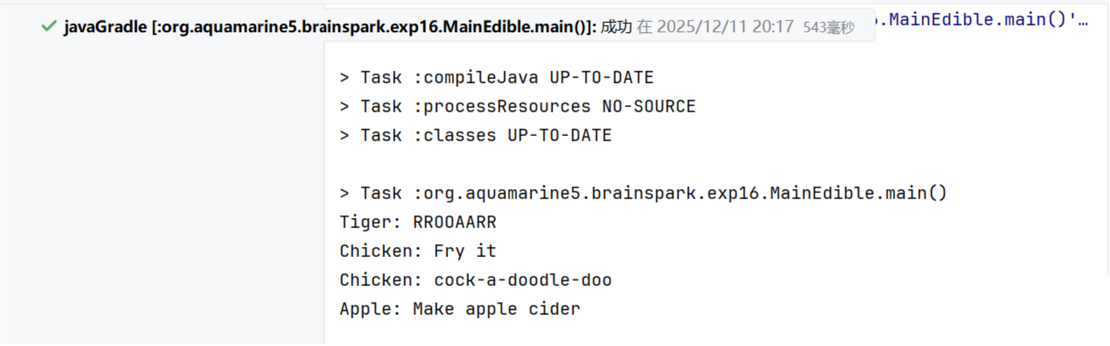
二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp16;

import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
public abstract class Animal {
    private double weight;
    public abstract String sound();
}
package org.aquamarine5.brainspark.exp16;

public class Apple extends Fruit{
    @Override
    public String howToEat() {
        return "Apple: Make apple cider";
    }
}
package org.aquamarine5.brainspark.exp16;

public class Chicken extends Animal implements Edible{
    @Override
    public String sound() {
        return "Chicken: cock-a-doodle-doo";
    }

    @Override
    public String howToEat() {
        return "Chicken: Fry it";
    }
}
package org.aquamarine5.brainspark.exp16;
```

```
public interface Edible {
    public abstract String howToEat();
}
package org.aquamarine5.brainspark.exp16;

public abstract class Fruit implements Edible {
}
package org.aquamarine5.brainspark.exp16;

public class MainEdible {
    public static void main(String[] args){
        Object[] objects=
                {new Tiger(),new Chicken(),new Apple()};
        for(Object obj:objects){
            if(obj instanceof Edible edible){
                System.out.println(edible.howToEat());
            }
            if(obj instanceof Animal animal){
                System.out.println(animal.sound());
            }
        }
    }
}
package org.aquamarine5.brainspark.exp16;

public class Orange extends Fruit{
    @Override
    public String howToEat() {
        return "Orange: Make orange juice";
    }
}
package org.aquamarine5.brainspark.exp16;

public class Tiger extends Animal{
    @Override
    public String sound() {
        return "Tiger: RROOAARR";
    }
}
```

三、实验结果



✓ javaGradle [:org.aquamarine5.brainspark.exp16.MainEdible.main()]: 成功 在 2025/12/11 20:17 543毫秒

```
.MainEdible.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp16.MainEdible.main()
Tiger: RROOAARR
Chicken: Fry it
Chicken: cock-a-doodle-doo
Apple: Make apple cider
```

实验十七 Java 程序--面向对象编程

一、实验目的

（1）编写程序，对常用的两个接口 Comparable，Cloneable 进行实现，并给出演示样例。

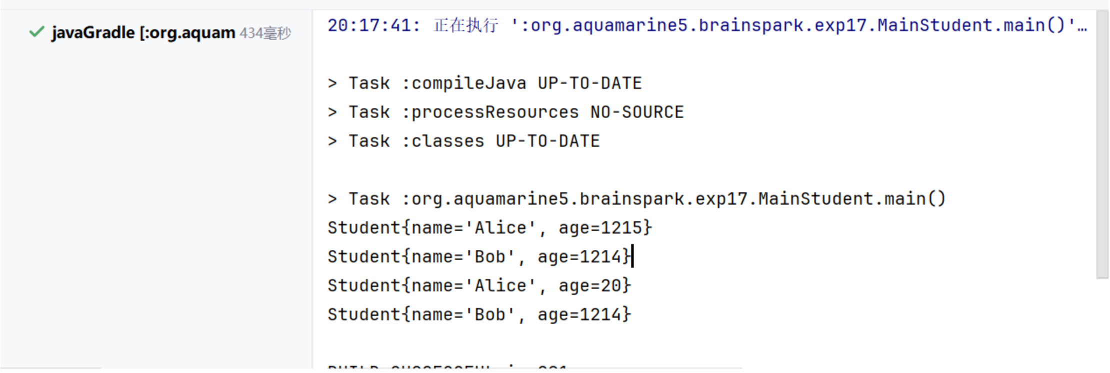
二、实验内容和步骤

（1）

```
package org.aquamarine5.brainspark.exp17;

public class People {
    private String major;
    private String name;
    private long id;

    public People(String major, String name, long id) {
        this.major = major;
        this.name = name;
        this.id = id;
    }
    public int love(){
        return 1;
    }
    public boolean love(int a){
        return true;
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
package org.aquamarine5.brainspark.exp17;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;
```

```java
@AllArgsConstructor
@Getter
@Setter
public class Student implements Cloneable, Comparable<Student> {
    private String name;
    private int age;

    @Override
    public int compareTo(Student o) {
        return Integer.compare(this.age, o.age);
    }

    @Override
    public Student clone() {
        try {
            return (Student) super.clone();
        } catch (CloneNotSupportedException e) {
            throw new RuntimeException(e);
        }
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + name + '\'' +
                ", age=" + age +
                '}';
    }
}
package org.aquamarine5.brainspark.exp17;

public class MainStudent {
    public static void main(String[] args) {
        Student s1 = new Student("Alice", 20);
        Student s2 = new Student("Bob", 22);
        Student s3 = s1.clone();
        Student s4 = s2;
        s1.setAge(1215);
        s4.setAge(1214);
        System.out.println(s1);
        System.out.println(s2);
        System.out.println(s3);
        System.out.println(s4);
    }
}
```

三、实验结果



```
20:17:41: 正在执行 ':org.aquamarine5.brainspark.exp17.MainStudent.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp17.MainStudent.main()
Student{name='Alice', age=1215}
Student{name='Bob', age=1214}
Student{name='Alice', age=20}
Student{name='Bob', age=1214}

BUILD SUCCESSFUL in 331
```

实验十八 Java 程序--面向对象编程

一、实验目的

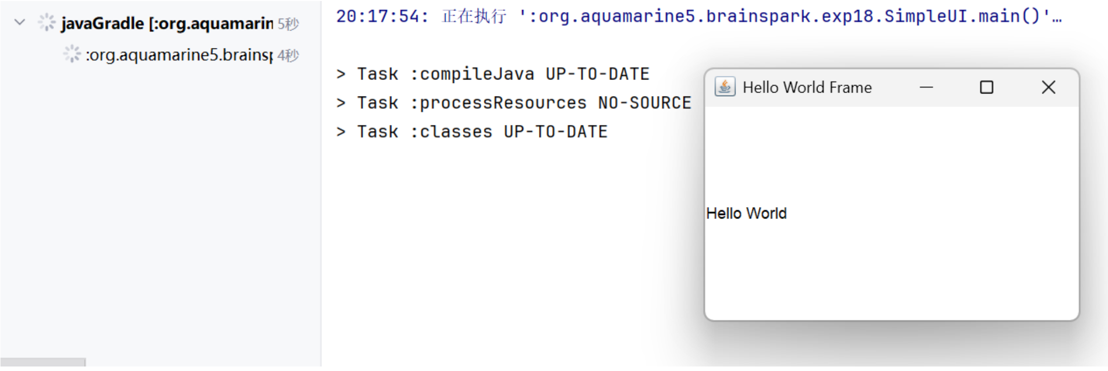
（1）编写一个简单的 GUI 程序，完成"Hello world"的显示。

二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp18;

import java.awt.*;
import java.awt.event.WindowAdapter;
import java.awt.event.WindowEvent;

public class SimpleUI extends Frame {
    public SimpleUI() {
        super("Hello World Frame");
        Label label = new Label("Hello World");
        add(label);
        setSize(300, 200);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent we) {
                System.exit(0);
            }
        });
    }

    public static void main(String[] args) {
        SimpleUI frame = new SimpleUI();
        frame.setVisible(true);
    }
}
```

三、实验结果



```
20:17:54: 正在执行 ':org.aquamarine5.brainspark.exp18.SimpleUI.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE
```

实验十九 Java 程序--面向对象编程

一、实验目的

（1）Binary I/O 与 Java 对象序列化

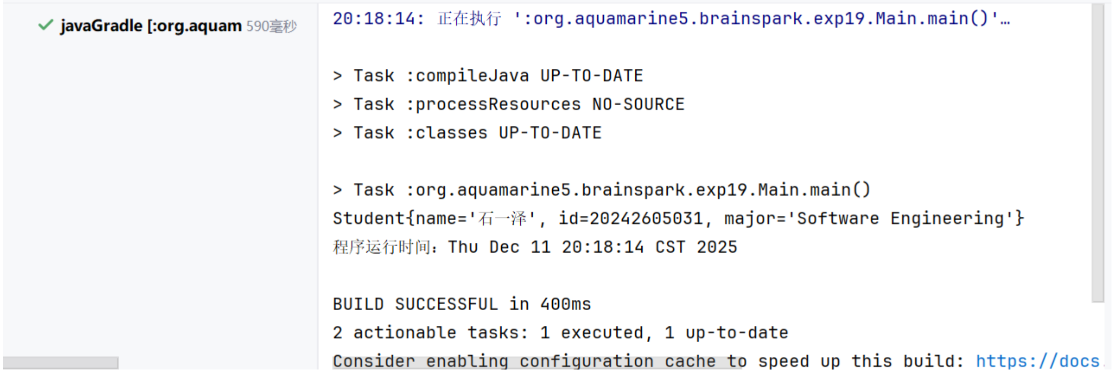
二、实验内容和步骤

（1）

```java
package org.aquamarine5.brainspark.exp19;

import lombok.AllArgsConstructor;
import lombok.Getter;
import lombok.Setter;

@AllArgsConstructor
@Getter
@Setter
public class Student implements java.io.Serializable {
    private String name;
    private long id;
    private String major;
    public Student() {
        this.name = "石一泽";
        this.id = 20242605031L;
        this.major = "Software Engineering";
    }

    @Override
    public String toString() {
        return "Student{" +
                "name='" + name + '\'' +
                ", id=" + id +
                ", major='" + major + '\'' +
                '}';
    }
}
```

```java
package org.aquamarine5.brainspark.exp19;

import java.io.*;
import java.util.Date;

public class Main {
  public static void main(String[] args) {
      Student student = new Student();
      try (FileOutputStream fos = new
FileOutputStream("student.dat");
           ObjectOutputStream oos = new ObjectOutputStream(fos)) {
          oos.writeObject(student);
      } catch (IOException e) {
          e.printStackTrace();
      }

      Student diskStudent = null;
      try (var fis = new FileInputStream("student.dat");
           var ois = new ObjectInputStream(fis)) {
          diskStudent = (Student) ois.readObject();
      } catch (IOException | ClassNotFoundException e) {
          e.printStackTrace();
      }finally {
          if (diskStudent != null) {
              System.out.println(diskStudent);
          }
      }
      System.out.println("程序运行时间："+new Date());
  }
}
```

三、实验结果



```
20:18:14: 正在执行 ':org.aquamarine5.brainspark.exp19.Main.main()'…

> Task :compileJava UP-TO-DATE
> Task :processResources NO-SOURCE
> Task :classes UP-TO-DATE

> Task :org.aquamarine5.brainspark.exp19.Main.main()
Student{name='石一泽', id=20242605031, major='Software Engineering'}
程序运行时间：Thu Dec 11 20:18:14 CST 2025

BUILD SUCCESSFUL in 400ms
2 actionable tasks: 1 executed, 1 up-to-date
Consider enabling configuration cache to speed up this build: https://docs
```