

```
/**
 *严靖炜,2017210346,2017211107
 * Exercise 01
 */
```

1.1

将 droid 文件夹复制到 D:\BUPT\大三上\JAVA\Homework\ch1 文件夹后:

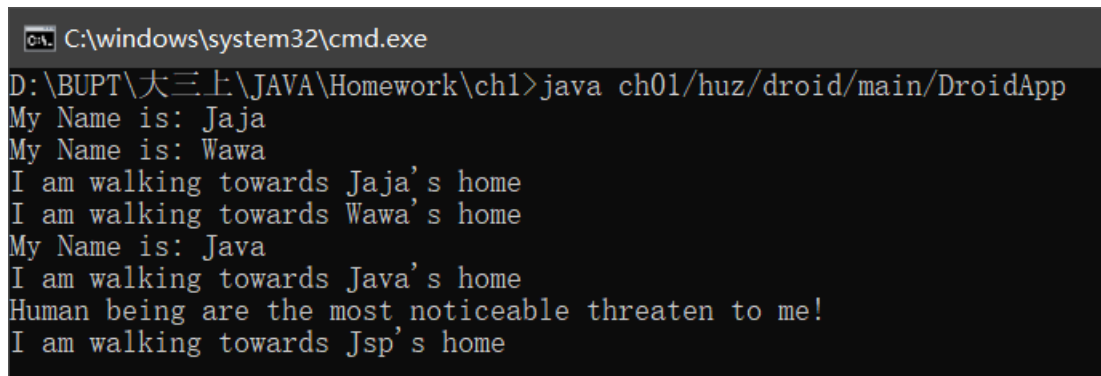
Source Code

Cmd

```
C:\Users\yju98>cd /d d:
D:\>cd D:\BUPT\大三上\JAVA\Homework\ch1\
D:\BUPT\大三上\JAVA\Homework\ch1>javac -d .
droid/adroid/Adroid.java
D:\BUPT\大三上\JAVA\Homework\ch1>javac -d . -classpath .
droid/adroid/smartdroid/SmartDroid.java
D:\BUPT\大三上\JAVA\Homework\ch1>javac -d . -classpath .
droid/main/DroidApp.java
D:\BUPT\大三上\JAVA\Homework\ch1>java ch01/huz/droid/main/DroidApp
```

Result

```
My Name is: Jaja
My Name is: Wawa
I am walking towards Jaja's home
I am walking towards Wawa's home
My Name is: Java
I am walking towards Java's home
Human being are the most noticeable threaten to me!
I am walking towards Jsp's home
```



The screenshot shows a Windows command prompt window with the title bar "C:\windows\system32\cmd.exe". The command prompt shows the following sequence of commands and outputs:

```
D:\BUPT\大三上\JAVA\Homework\ch1>java ch01/huz/droid/main/DroidApp
My Name is: Jaja
My Name is: Wawa
I am walking towards Jaja's home
I am walking towards Wawa's home
My Name is: Java
I am walking towards Java's home
Human being are the most noticeable threaten to me!
I am walking towards Jsp's home
```

小结:

- (1).何时需要使用 `-classpath`: 当要编译或执行的类引用了其它的类, 但被引用类的 `.class` 文件不在当前目录下时, 就需要通过 `-classpath` 来引入类
- (2).何时需要指定路径: 当要编译的类所在的目录和执行 `javac` 命令的目录不是同一个目录时, 就需要指定源文件的路径(`CLASSPATH` 是用来指定 `.class` 路径的, 不是用来指定 `.java` 文件的路径的)

1.2

Source Code

Linear.java

```
public class Linear {
    public static void main(String[] args) {
        double a = 3.4, b = 50.2;
        double c = 2.1, d = 0.55;
        double e = 44.5, f = 5.9;
        //Cramer 法则
        double x = (e * d - b * f) / (a * d - b * c);
        double y = (a * f - e * c) / (a * d - b * c);
        //System.out.println("x=" + x + ",y=" + y);
        System.out.println("x=" + String.format("%.2f", x) + ",y="
+ String.format("%.2f", y)); //格式化输出, 避免输出小数过长
    }
}
```

Result

x=2.62,y=0.71

1.3

Source Code

Invest.java

```
package ch1;

public class Invest {
    private double annualInterestRate;
    private int numberOfYears;
    private double investAmount;
    private java.util.Date investDate;

    /** Default constructor */
    public Invest() {
        this(3.25, 1, 1000);
    }

    /** Construct a investment with specified annual interest
rate,
    number of years, and investment amount
    */
    public Invest(double annualInterestRate, int numberOfYears,
        double investAmount) {
        this.annualInterestRate = annualInterestRate;
    }
}
```

```

        this.numberOfYears = numberOfYears;
        this.investAmount = investAmount;
        investDate = new java.util.Date();
    }

    /** Return annualInterestRate */
    public double getAnnualInterestRate() {
        return annualInterestRate;
    }

    /** Set a new annualInterestRate */
    public void setAnnualInterestRate(double annualInterestRate) {
        this.annualInterestRate = annualInterestRate;
    }

    /** Return numberOfYears */
    public int getNumberOfYears() {
        return numberOfYears;
    }

    /** Set a new numberOfYears */
    public void setNumberOfYears(int numberOfYears) {
        this.numberOfYears = numberOfYears;
    }

    /** Return InvestAmount */
    public double getInvestAmount() {
        return InvestAmount;
    }

    /** Set a newInvestAmount */
    public void setInvestAmount(double investAmount) {
        this.InvestAmount = investAmount;
    }

    /** Find total Investment */
    public double getTotalInvestment() {
        double totalInvestment = investAmount *
            (Math.pow(1 + annualInterestRate/1200, numberOfYears *
12));
        return totalInvestment;
    }

    /** Return Invest date */

```

```

        public java.util.Date getInvestDate() {
            return investDate;
        }
    }
}

InvestCalculator.java
package chl;

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.geometry.HPos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.awt.event.ActionEvent;

public class InvestCalculator extends Application {
    private TextField tfAnnualInterestRate = new TextField();
    private TextField tfNumberOfYears = new TextField();
    private TextField tfInvestAmount = new TextField();
    private TextField tfTotalInvestment = new TextField();
    private Button btCalculate = new Button("Calculate");

    @Override // Override the start method in the Application
class
    public void start(Stage primaryStage) {
        // Create UI
        GridPane gridPane = new GridPane();
        gridPane.setHgap(5);
        gridPane.setVgap(5);
        gridPane.add(new Label("Annual Interest Rate:"), 0, 0);
        gridPane.add(tfAnnualInterestRate, 1, 0);
        gridPane.add(new Label("Number of Years:"), 0, 1);
        gridPane.add(tfNumberOfYears, 1, 1);
        gridPane.add(new Label("Invest Amount:"), 0, 2);
        gridPane.add(tfInvestAmount, 1, 2);
        gridPane.add(new Label("Total Investment:"), 0, 4);
        gridPane.add(tfTotalInvestment, 1, 4);
        gridPane.add(btCalculate, 1, 5);

        // Set properties for UI

```

```

        gridPane.setAlignment(Pos.CENTER);
        tfAnnualInterestRate.setAlignment(Pos.BOTTOM_RIGHT);
        tfNumberOfYears.setAlignment(Pos.BOTTOM_RIGHT);
        tfInvestAmount.setAlignment(Pos.BOTTOM_RIGHT);
        tfTotalInvestment.setAlignment(Pos.BOTTOM_RIGHT);
        tfTotalInvestment.setEditable(false);
        GridPane.setHalignment(btCalculate, HPos.RIGHT);

        // Process events
        btCalculate.setOnAction(e -> calculateInvestInvestment());

        // Create a scene and place it in the stage
        Scene scene = new Scene(gridPane, 400, 250);
        primaryStage.setTitle("InvestCalculator"); // Set title
        primaryStage.setScene(scene); // Place the scene in the
stage
        primaryStage.show(); // Display the stage
    }

    private void calculateInvestInvestment() {
        // Get values from text fields
        double interest =
            Double.parseDouble(tfAnnualInterestRate.getText());
        int year = Integer.parseInt(tfNumberOfYears.getText());
        double investAmount =
            Double.parseDouble(tfInvestAmount.getText());

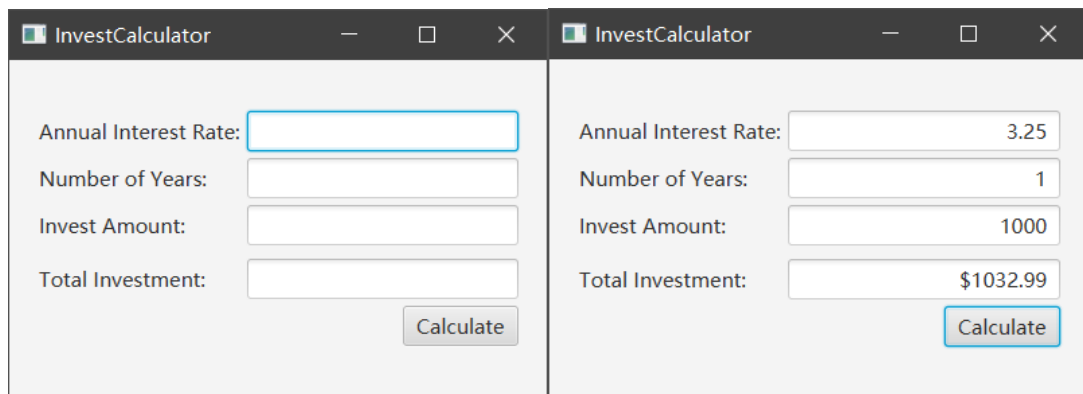
        // Create a invest object. Invest defined in Listing 10.2
        Invest invest = new Invest(interest, year, investAmount);

        // Display total Investment
        tfTotalInvestment.setText(String.format("$%.2f",
            invest.getTotalInvestment()));
    }

    /**
     * The main method is only needed for the IDE with limited
     * JavaFX support. Not needed for running from the command
line.
     */
    public static void main(String[] args) {
        launch(args);
    }
}

```

Result



Field	Value
Annual Interest Rate	3.25
Number of Years	1
Invest Amount	1000
Total Investment	\$1032.99

1.4

Source Code

YearCalculator.java

```
import java.io.*;

public class YearCalculator {
    public static void main(String[] args) throws IOException {
        //从标准输入读取一个字符串
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        String str;
        System.out.print("Input minute:");
        str = br.readLine();
        //类型转换
        double minute=Double.valueOf(str);
        //计算时间并输出
        double day=minute/60/24;
        System.out.println("Day:" + String.format("%.2f", day));
        double year=day/365;
        System.out.println("Year:" + String.format("%.2f", year));
    }
}
```

Result

Demo1:

Input minute:7884000

Day:5475.00

Year:15.00

Demo2:

Input minute:1000000000

Day:694444.44

Year:1902.59

1.5

Source Code

Calculator.java

```
package ch02;

public class Calculator {
    private static int result; // 静态变量, 用于存储运行结果

    public void add(int n) {
        result = result + n;
    }

    public void subtract(int n) {
        result = result - n;
    }

    public void multiply(int n) {
        result = result * n;
    }

    public void divide(int n) {
        result = result / n;
    }

    public void square() {
        result = Math.sqrt(n);
    }

    public void power(int n) {
        result = Math.pow(result, n);
    }

    public void clear() { // 将结果清零
        result = 0;
    }

    public int getResult() {
        return result;
    }
}
```

```
}
```

TestCalculator.java

```
package ch02;
```

```
public class TestCalculator {  
    public static void main(String[] args) {  
        Calculator cal=new Calculator();  
        System.out.println(cal.getResult());  
        cal.add(200);  
        System.out.println(cal.getResult());  
        cal.subtract(140);  
        System.out.println(cal.getResult());  
        cal.multiply(2);  
        System.out.println(cal.getResult());  
        cal.divide(6);  
        System.out.println(cal.getResult());  
    }  
}
```

```
}
```

Result

0

200

60

120

20