实验一 初识 Java

实验目的

1. 掌握 JDK 的安装与环境变量的配置方法。
2. 了解 Java 源代码、字节码文件，熟悉 Java 程序编辑、编译和运行的方法。
3. 运行简单的 Java Application 程序，了解程序的结构。

实验预习

Java 的工作方式

Java 应用程序会历经代码的编辑、编译、运行三个步骤，最后得到运行结果。图 1.1 为一个简单 Java 程序的执行过程。

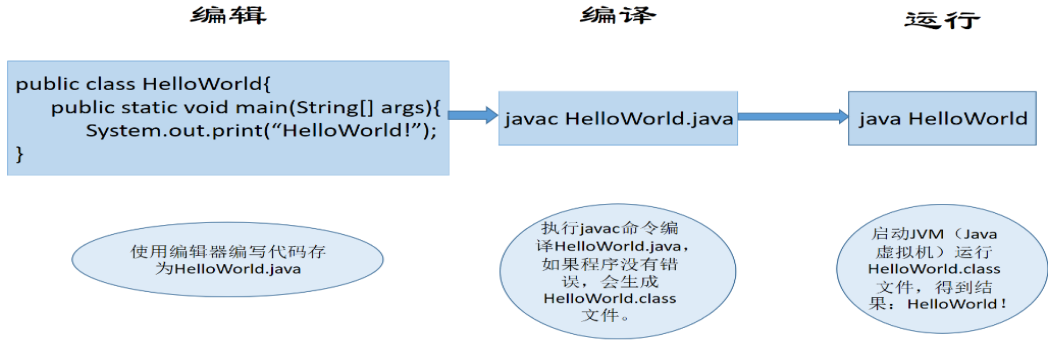

图 1.1 HelloWorld.java 的执行过程

基础练习

1. JDK、IDE 的安装

实现 Java 程序的编译和运行，我们需要安装 Java Development Kit（JDK）Java 软件开发工具包。安装方法及步骤需要同学们自主研学，JDK 版本及 IDE 可任选，但需要满足以下几点要求：

1. 官网下载正版软件；
2. 软件版本选择较新的长期支持版；
3. 尽量选择免费软件或收费软件的免费教育版；
4. 安装软件时注意提示信息。

2. 配置环境变量

环境变量的配置涉及三个变量，为别为 JAVA_HOME、PATH 和 CLASSPATH。在“命令提示符”程序中输入 javac 和 java 命令，检查环境变量设置是否正确。

3. 简单的 Java 应用程序

1. 编辑程序

编写 Java 程序，可以打开“记事本”进行编写，也可使用其他编辑软件进行程序的编写。完成代码的编写后，需以类的名称为文件命名，例如以下程序被命名为 HelloWorld.java。

```
public class HelloWorld{
  public static void main(String[] args){
    System.out.println("HelloWorld!");
  }
}
```

2. 编译程序

使用“命令提示符”进行程序的编译。打开“命令提示符”后，首先需要将显示路径切换到 HelloWorld.java 文件所在目录，再输入编译程序命令：javac HelloWorld.java。如程序没有错误，编译成功后，在同一目录下产生 HelloWorld.class 文件，并出现以下提示，如图 1.2 所示。

```
D:\>javac HelloWorld.java

D:\>
```

图 1.2 编译图示

3. 运行程序

编译成功后，输入运行命令 java HelloWorld。这时 JVM 启动，在命令行寻找指定的类（与文件名同名的类），执行得到以下运行结果，如图 1.3 所示。

```
D:\>java HelloWorld
HelloWorld!

D:\>
```

图 1.3 运行图示

扩展练习

从提供的语句序列中选择适当的语句填入程序的空白处，必须要能够编译并运行出如图 1.4 所示的结果。注：每条语句只能使用一次。

```
D:\>javac Puzzle1.java

D:\>java Puzzle1
a noise
annoys
an oyster
```

System.out.print(" ");	x = x + 1;
System.out.print("a");	x = x + 2;
System.out.print("n");	x = x - 2;
System.out.print("an");	x = x - 1;
x > 0;	System.out.print("noys");
x < 1;	System.out.print("oise");
x > 1;	System.out.print(" oyster");
x > 3;	System.out.print("annoys");
x < 4;	System.out.print("noise");

图 1.4 Puzzle1 运行结果

```
class Puzzle1{
  public static void main(String[] args){
    int x = 0;
    while(__1__){
        _____2_____
        if(x < 1){
            _____3_____
        }
        _____4_____
        if(__5__){
            _____6_____
            ___7___
        }
        if(x == 1){
            _____8_____
        }
        if(__9__){
            _____10_____
        }
        System.out.println(" ");
        ___11___
    }
  }
```

实验二 Java 基本语法

实验目的

1. 熟悉 Java 的程序结构。
2. 熟练掌握 Java 的 8 种基本数据类型，包括它们各自的取值范围及类型之间的转换。
3. 熟练掌握各种运算符及其优先级次序。
4. 熟练掌握三种语句类型。

实验预习

1. Java 的程序结构

Java 程序的源文件（扩展名为.java）包含类的定义，类存于源文件中。类用来表示程序的一个组件，小程序或许只会有一个类，类的内容必须包在花括号中。方法存于类中，类中可以带有一个或多个方法，方法必须在类的内部声明。如图 2.1 所示，Dog 这个类中，bark 方法包含如何“汪汪”的指令。在方法的花括号中编写方法应该执行的指令，方法代码由一组语句所组成，你可以把方法想象成一个函数或过程。

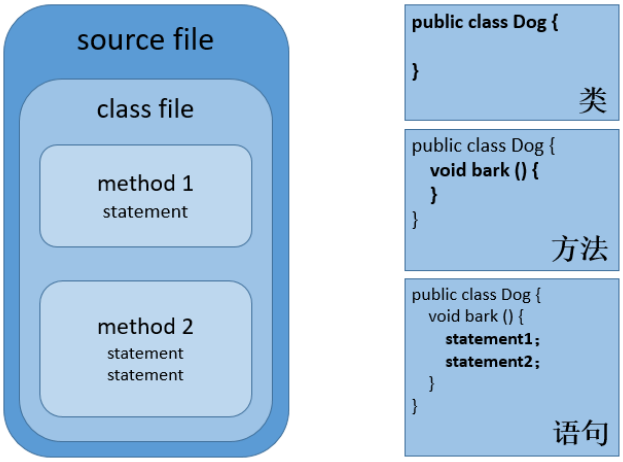

图 2.1 Java 的程序结构

2. 剖析一个 Java 类

当 Java 虚拟机启动时，它会寻找你在命令行所指定的类，找到之后它会在其中找到“main”这样一个特定的方法，接着 Java 虚拟机就会执行 main 方法在花括号间的所有指令。每个 Java 程序最少都会有一个类，每个应用程序只能有一个 main() 函数。一个简单的 HelloWorld 程序每部分具体的说明如图 2.2 所示。

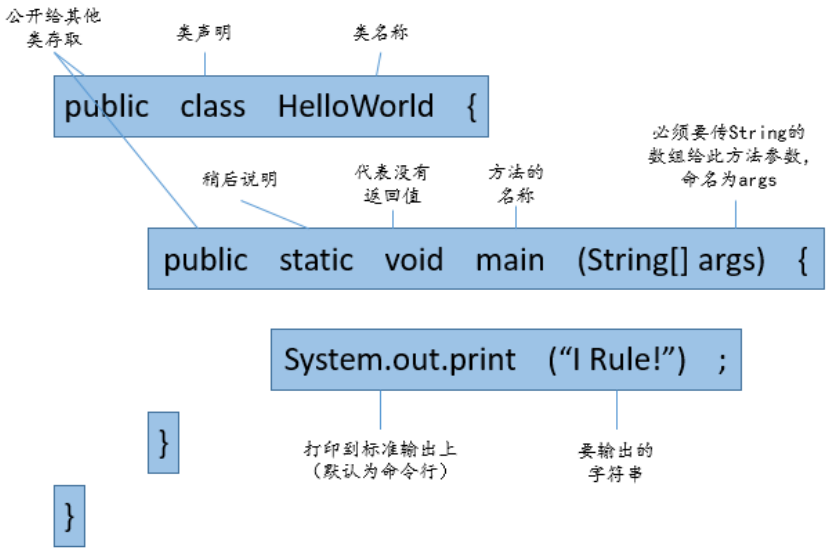

图 2.2 HelloWorld 程序说明

3. 基本数据类型及运算符

1. 基本数据类型及类型转换运算

基本数据类型共有 8 种，类型名称、位数、取值范围、声明及赋值方法如表 2.1 所示。

将一种基本数据类型变量值赋给另一个基本类型变量时，就涉及数据转换。下列基本类型会涉及数据转换，精度从低到高排列为：byte→short→char→int→long→float→double。

将低精度的变量值赋给高精度的变量时，系统自动完成数据类型转换。将高精度的变量值赋给低精度的变量时，必须要使用类型转换运算，格式为:（类型名）要转换的值，例如：int x = (int)12.34;。

表 2.1 基本数据类型

名称	boolean	char	byte	short	int	long	float	double
位数	Java虚拟机决定	16	8	16	32	64	32	64
值域	true或false	0 ~ 65535	-128 ~ 127	-32768 ~ 32767	-2147483648 ~ 2147483647	- 很大 ~ 很大	范围规模可变	范围规模可变
声明与赋值	boolean x = ture;	char x = 'a';	byte x = 123;	short x = 1234;	int x = 12345;	long x = 123456;	float x = 12.34f;	double x = 123.45;

2. 运算符及其优先级

Java 所有运算符、运算符的优先级和结合方向可查看表 2.2。在编写程序时尽量使用括号运算符号来实现想要的运算次序，以免产生难以阅读的计算顺序。要注意表中的结合性，它决定了优先级相同的运算符的先后顺序。

表 2.2 运算符的优先级及结合性

优先级	描述	运算符	结合性
1	分隔符	[] () . , ;	右到左
2	对象归类、自增、自减运算、逻辑非	Instanceof ++ -- !	左到右
3	“算术乘除”运算	* / %	左到右
4	“算术加减”运算	+ -	左到右
5	“移位”运算	>> << >>>	左到右
6	“大小关系”运算	< <= > >=	左到右
7	“相等关系”运算	== !=	左到右
8	“按位与”运算	&	左到右
9	“按位异或”运算	^	左到右
10	“按位或”运算	|	左到右
11	“逻辑与”运算	&&	左到右
12	“逻辑或”运算	||	左到右
13	“三目条件”运算	?:	左到右
14	“赋值”运算	=	右到左

4. 语句基本结构

Java 语句的基本结构分为顺序、分支和循环三种。顺序结构代码可以理解为让 Java 虚拟机去依次做某些事（包括声明、赋值、调用方法等普通语句）；分支结构代码可以理解为让 Java 虚拟机在适当的条件下做某件事，结构如图 2.3，包括 if 和 switch 语句；循环结构代码可以理解为让 Java 虚拟机在满足某个条件时反复做某件事，结构如图 2.4，包括 while、do-while 和 for 语句。

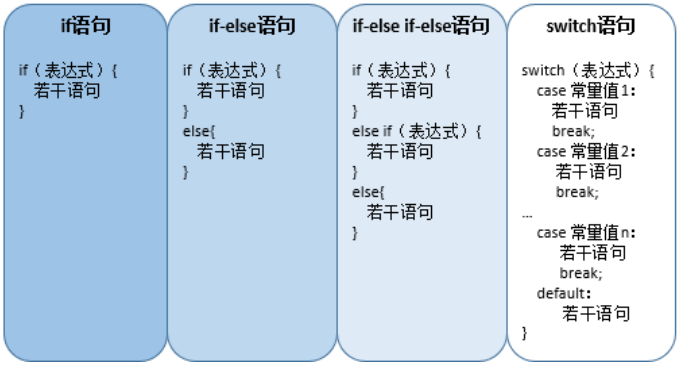

图 2.3 分支结构语句

分支语句：

if、if-else 和 if-else if-else 语句，条件满足“表达式”要求时，运行随后大括号中包含的若干语句；不满足表达式要求时，执行 else 语句后大括号中的若干语句，如果没有 else 部分，则继续执行分支结构后面的语句。

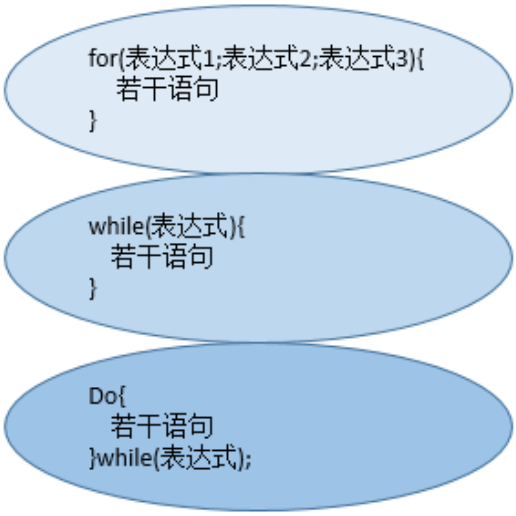

图 2.4 循环结构语句

循环语句：

for 中“表达式 1”完成变量初始化；“表达式 2”设置循环条件；“表达式 3”用来改变循环条件。

while 中“表达式”为循环条件，满足则执行一次大括号中语句。

do-while 中“表达式”也表示循环条件，但与 while 不同点在于大括号中语句先执行一次后再判断循环条件。在循环中包含“break;”和“continue;”两个语句时，如果某次循环中执行了“break;”语句，那么整个循环结束；如果某次循环中执行了“continue;”语句，那么本次循环结束。

加强版的 for 循环

for 循环还有一种形式：for(String name: nameArray){……}

这行程序的意思是：对 nameArray 中的每个元素执行一次。

编译器理解为：

① 创建名称为 name 的 String 变量。

② 将 nameArray 的第一个元素值赋给 name。

③ 执行循环体中的内容。

④ 再将 nameArray 的第二个元素值赋给 name，并执行循环体中的内容。

⑤ 重复执行至所有元素都被运行为止。

一个简单的加强版 for 循环例子，说明 main()方法中参数的含义。main()方法的参数是在运行时传入的，例如下面程序在运行 “java main”命令时，后面所带的即是 main()方法的参数。先分析运行结果，然后上机操作查看结果。

```
public class Main{
  public static void main(String[] args){
    for(String str: args){
      System.out.println(str);
    }
  }
}
```

编译运行如下：

```
javac Main.java
```

```
java Main hello java hello java
```

5. 读程序练习

读下列程序，写出运行结果。

1. 输出汉字位置

```
public class Exercise2_1{
  public static void main(String args[]){
    char chinaWord = '好';
    System.out.println("汉字:" + chinaWord + "的位置:" + (int)chinaWord
+ "\n");
  }
}
```

运行结果：_____

2. 输出指定位置上的汉字

```
public class Exercise2_2{
  public static void main(String args[]){
    int position = 20320;
    System.out.println(position + "位置上的字符是：" + (char)position);
    position = 21319;
    System.out.println(position + "位置上的字符是：" + (char)position +
"\n");
  }
}
```

运行结果：_____

3. 分析“u4F60”与 20320 的关系

```
public class Exercise2_3{
  public static void main(String args[]){
    char you = '\u4F60';
      System.out.println("you:" + you + "\n");
  }
}
```

运行结果：_____

4. 思考初值与输出为何不同

```
public class Exercise2_4{
  public static void main(String args[]){
    float a = 12345.123456789f;
      System.out.println("a=" + (float)a + "\n");
  }
}
```

运行结果：_____

5. 分析两个赋值语句的差别

```
public class Exercise2_5{
  public static void main(String args[]){
    double height = 23.345d;
```

```
    double width = 34.56789;
    System.out.println("height=" + (double)height);
    System.out.println("width=" + (double)width + "\n");
  }
}
```

运行结果：_____

6. 类型的精度从低至高，系统自动完成转换

```
public class Exercise2_6{
  public static void main(String args[]){
    int b = 50;
    float c;
    c = b;
    System.out.println("b=" + (int)b);
    System.out.println("c=" + (float)c + "\n");
  }
}
```

运行结果：_____

7. 类型的精度从高至低，必须使用类型转换运算，对比 e,f,g,h 的初值与输出结果

```
public class Exercise2_7{
  public static void main(String args[]){
    byte e = 22;
    float f = 123.45678f;
    System.out.println("e=" + (byte)e);
    System.out.println("f=" + (float)f);
    int g = 129;
    double h = 123456.456789;
    g = (byte)g;
    h = (float)h;
    System.out.println("g=" + g);
    System.out.println("h=" + h + "\n");
  }
}
```

运行结果：_____

8. 输入数据练习

```
import java.util.Scanner;
public class Exercise2_8{
  public static void main(String args[]){
    Scanner reader = new Scanner(System.in);
    double i = 0;
    double j = reader.nextDouble();
    while(j!=0){
      i = i + j ;
      j = reader.nextDouble();
    }
```

```
        System.out.println("i=" + i + "\n");
      }
    }
```

运行结果：_____

9. 数组的使用

```
public class Exercise2_9{
  public static void main(String args[]){
    int k[] = {1,2,3,4};
    int l[] = {100,200,300};
    System.out.println("数组 k 的元素个数=" + k.length);
    System.out.println("数组 l 的引用=" + l);
    k = l;
    System.out.println("数组 k 的元素个数=" + k.length);
    System.out.println("k[0]=" + k[0] + ",k[1]=" + k[1] + ",k[2]=" +
k[2] + "\n");
  }
}
```

运行结果：_____

10. 枚举类型

```
public class Exercise2_10{
  public static void main(String args[]){
    Season season = Season.spring;
    System.out.println(season);
  }
}
enum Season //枚举类型声明
   {spring,summer,autumn,winter}
```

运行结果：_____

基础练习

1. 补全程序练习

根据程序说明及缺失部分的代码注释，补写代码完成程序功能。

1. 设计程序要求输入一个日期后，可以显示下一天日期。部分代码如下，在空白处根据注释补全程序。

```
import java.util.Scanner;
public class GetNextDate{
  public static void main(String[] args){
    int y,m,d;    //年、月、日
    int dnum = 0;  //月里的天数
    boolean isLeap = false;  //是否为闰年
    System.out.println("请输入年月日信息，格式为：年-月-日");
      Scanner scanner = new Scanner(System.in);
```

```
      //以字符串方式输入日期并分离出年月日
      String date = scanner.nextLine();
      String ymd[] = date.split("-");
      y = Integer.parseInt(ymd[0]);
      m = Integer.parseInt(ymd[1]);
      d = Integer.parseInt(ymd[2]);
      //用 if 语句判断输入的年、月是否合法，要求 1000<y<9999
      此处填写程序段，满足上行的注释要求。
      if((y % 400 == 0)||(y % 4 == 0 && y % 100 != 0)){
        isLeap = true;
      }
      //用 switch 语句判断月份，求出该月份的天数
      此处填写程序段，满足上行的注释要求。
      if(d < 1 || d > dnum){
        System.out.println("日期不合法! ");
        return;
      }
      if(d != dnum){
        d++;
      }
      else{
        if(m == 12){
          y++;
          m = 1;
          d =1;
        }
        else{
          m++;
          d = 1;
        }
      }
      System.out.println(y + "-" + m + "-" + d);
  }
}
```

2. 设计程序，求 2~100 之间的所有素数，要求输出时每行显示 5 个素数部分代码如下，在空白处根据注释补全程序。

```
public class FindPrime{
  public static void main(String[] args){
    int t = 0;
    boolean f = true;
    //用两层循环语句求出素数
    //第一层循环为遍历 2~100（变量设为：i）
    //第二层循环为遍历 2~i-1
    此处填写程序段，满足上面的注释要求。
```

```
    if(f){
      t++;
      if(t % 5 == 0){
        System.out.println(i + " ");
      }
      else{
        System.out.print(i + " ");
      }
      }
    }//此处括号为需补全程序中第一层循环结束
  }
}
```

2. 编写程序练习

1. 用循环语句计算 1 + 1/2! + 1/3! + 1/4! …的前 20 项之和。（提示：结果要求误差小于 0.0001）

2. 输入不小于 10 个数，再将这些数按从小到大的顺序输出。（提示：输入数的类型可自由定义）

3. 设计程序实现输入日期及机票张数，计算出应付金额。假设北京至上海的机票全价为 1200 元/张，以 2017 年为例进行程序编写，所有的法定假日，机票无折扣；除法定假日之外的周末，机票价格为 8 折；除法定假日及周末之外的工作日，机票价格为 6 折。（提示：只需考虑月份及日期）