



HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

## Java 程序设计实验报告

学号:

姓名:

专业:

班级:

## 实验一：熟悉开发环境及简单程序设计

### 一、实验目的

- 1) 认识 Java 运行环境;
- 2) 熟悉开发工具 Eclipse 的使用(创建工程、创建包、创建 java 文件、编译运行等);
- 3) 掌握 Java 程序的编辑、编译和运行过程;
- 4) 了解 Java 源代码、字节码文件;
- 5) 掌握 Java 中数据类型、变量、表达式的使用;
- 6) 掌握标准输入输出函数的使用。
- 7) 掌握自定义 static 方法及其调用
- 8) 掌握 if 和 if-else 和 switch 分支语句
- 9) 掌握 while、do-while 和 for 循环语句
- 10) 掌握 break 和 continue 程序控制语句
- 11) 掌握嵌套循环的使用方法
- 12) 掌握数组的定义和使用方法
- 13) 了解二维数组的特性
- 14) 掌握简单排序算法

### 二、实验内容

1) 利用文本编辑器编写 HelloWorld 类，保存到 D 盘根目录下，并在命令行下进行编译执行。

**Step1**，在 D 盘根目录下利用文本编辑器编写如下程序，并保存为 HelloWorld.java 文件



```
//HelloWorld.java
package edu. .... .experiment1;
public class HelloWorld{
    public static void main(String[] args)
    {
        System.out.print("Hello World!");
    }
}
```

**Step2**，启动一个命令窗口，输入下列命令：

```
C:\ C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>d:

D:\>javac -d . HelloWorld.java
```

第一个命令 `javac -d . HelloWorld.java` 表示对当前目录下的 `HelloWorld.java` 源程序进行编译。其中 `-d` 参数表示设置输出类文件的位置。如果某个类是一个包的组成部分，则 `javac` 将把该类文件放入反映包名的子目录中，必要时创建目录。

即在本例中，`javac` 在 D 盘根目录下，按照 `HelloWorld.java` 中声明的包名 `edu.fundamentals.java.experiment1`，建立 `edu` 目录，在 `edu` 目录下建立 `fundamentals` 子目录，然后又在 `fundamentals` 下建立 `experiment1` 子目录，并在该目录下编译生成 `HelloWorld.class` 文件。

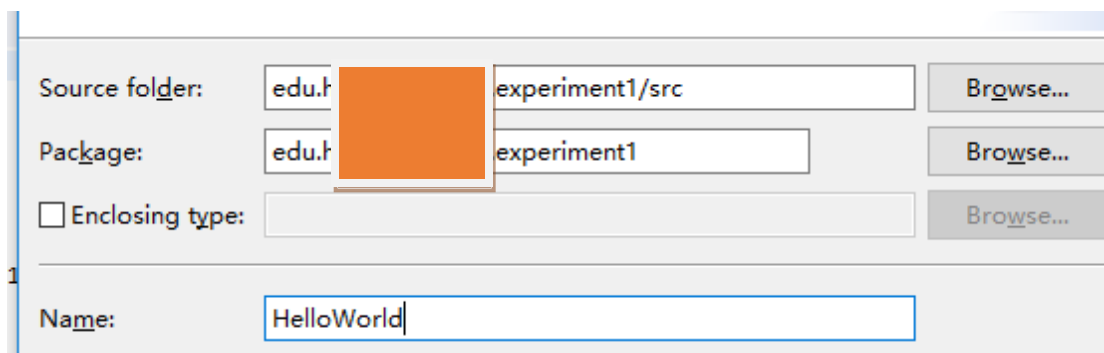
第二个命令表示执行该 Java 程序。

**Step3** 实验结果符合预期，正确输出了 `Hello World!`

```
D:\>java edu.fundamentals.java.experiment1.HelloWorld
HelloWorld!
D:\>_
```

## 2) 利用 Eclipse 开发环境，编写 HelloWorld 类，并在 Eclipse 开发环境中编译执行。

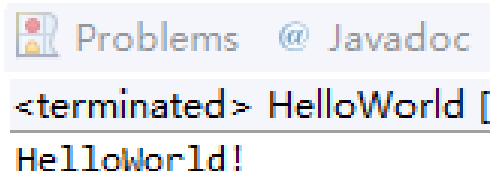
**Step1**，创建 `HelloWorld`



**Step2**，编写代码。

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

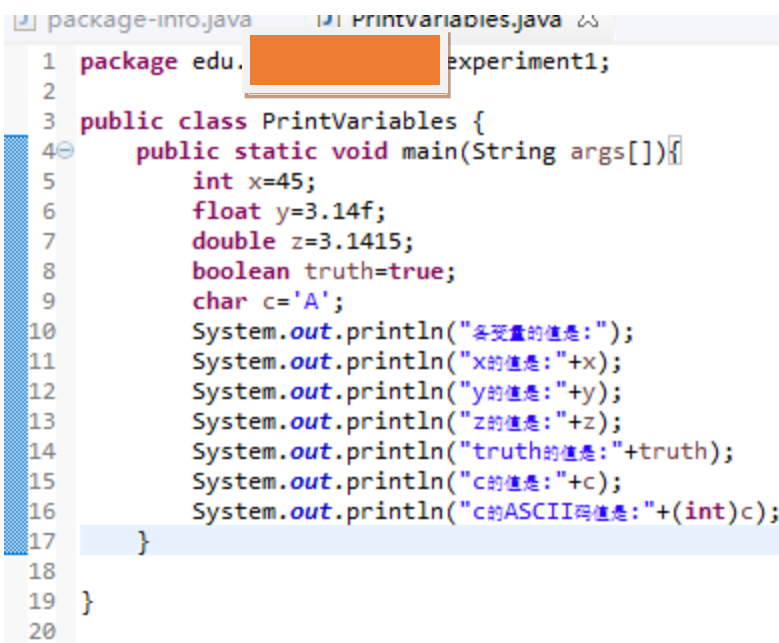
Step3, 运行得到如下结果:



Problems @ Javadoc  
<terminated> HelloWorld [  
HelloWorld!

### 3) 补全如下程序，观察并解释打印输出的结果。

Step1, 补全后的代码如图:



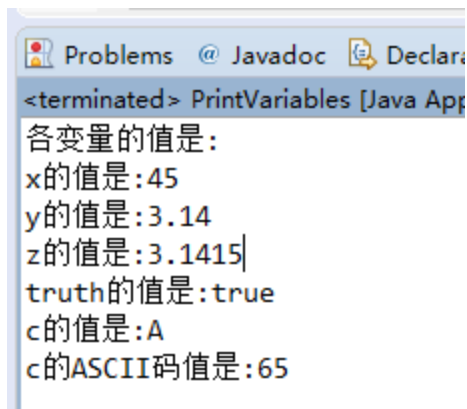
```
package edu. [redacted] experiment1;  
  
public class PrintVariables {  
    public static void main(String args[]) {  
        int x=45;  
        float y=3.14f;  
        double z=3.1415;  
        boolean truth=true;  
        char c='A';  
        System.out.println("各变量的值是:");  
        System.out.println("x的值是:"+x);  
        System.out.println("y的值是:"+y);  
        System.out.println("z的值是:"+z);  
        System.out.println("truth的值是:"+truth);  
        System.out.println("c的值是:"+c);  
        System.out.println("c的ASCII码值是:"+ (int)c);  
    }  
}
```

Step2, 运行结果及解释:

使用 println 输出，将自动在每一条输出后换行

输出 (int) c 时，将 c 在内存中的数据直接转化成整数，由此可见，字符在内存中以整数 ASCII 码存储

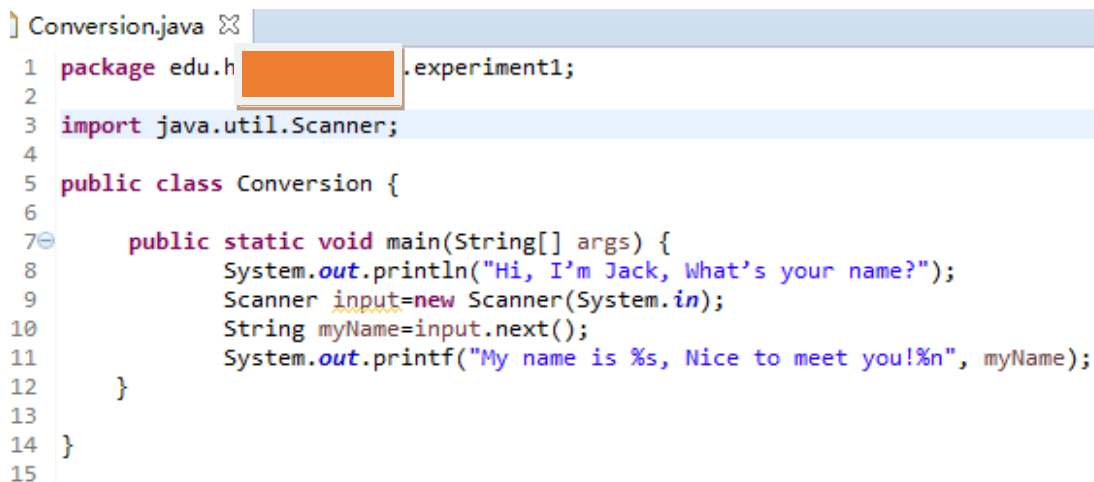
所以，结果如下:



```
<terminated> PrintVariables [Java App
各变量的值是:
x的值是:45
y的值是:3.14
z的值是:3.1415
truth的值是:true
c的值是:A
c的ASCII码值是:65
```

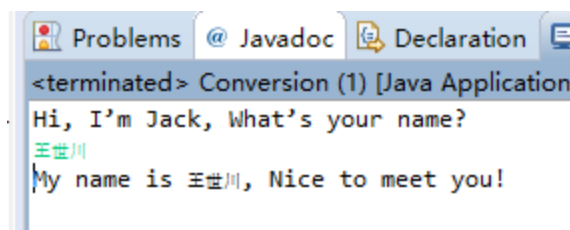
4) 编写 Conversion 类，将该程序中横线部分补充完整，并解释其中 printf 函数的作用。

Step1, 代码如下



```
Conversion.java
1 package edu.h[redacted].experiment1;
2
3 import java.util.Scanner;
4
5 public class Conversion {
6
7     public static void main(String[] args) {
8         System.out.println("Hi, I'm Jack, What's your name?");
9         Scanner input=new Scanner(System.in);
10        String myName=input.next();
11        System.out.printf("My name is %s, Nice to meet you!\n", myName);
12    }
13
14 }
15
```

Step2, 结果:



```
<terminated> Conversion (1) [Java Application
Hi, I'm Jack, What's your name?
吾世川
My name is 吾世川, Nice to meet you!
```

Step3, 解释:

其中 printf 函数的作用是格式化输出。使用一系列占位符，先将句子的格式固定，然后在传入每个占位符所对应的内容，使得文本格式可控，内容可变。

5) 编写 Averager 类，完成如下功能：

a) floatAvg 函数：从键盘输入两浮点数，计算其平均值，并打印输出。

b) 平均值的打印输出只保留 2 位小数。

Step1, 代码如下

```
import java.util.Scanner;

public class Averager {

    static float floatAvg(){

        Scanner input = new Scanner(System.in);
        float var1=input.nextFloat();
        float var2=input.nextFloat();
        float aver=(var1+var2)/2f;

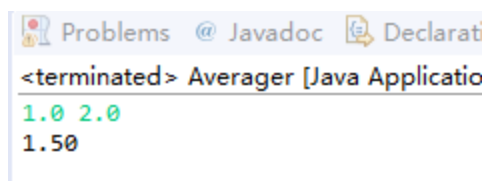
        return aver;
    }

    public static void main(String[] argv){

        System.out.printf("%.2f",floatAvg());

    }
}
```

Step2, 运行结果:



```
Problems @ Javadoc Declarat
<terminated> Averager [Java Applicatio
1.0 2.0
1.50
```

6) 编写 ControlStrcutureTest 类，为该类增加一个函数

checkLeapYear，函数参数为整数年份，返回值为 true/false，该函数用于判断某年份是否闰年，在 man 函数中，分别用 220, 618, 960,

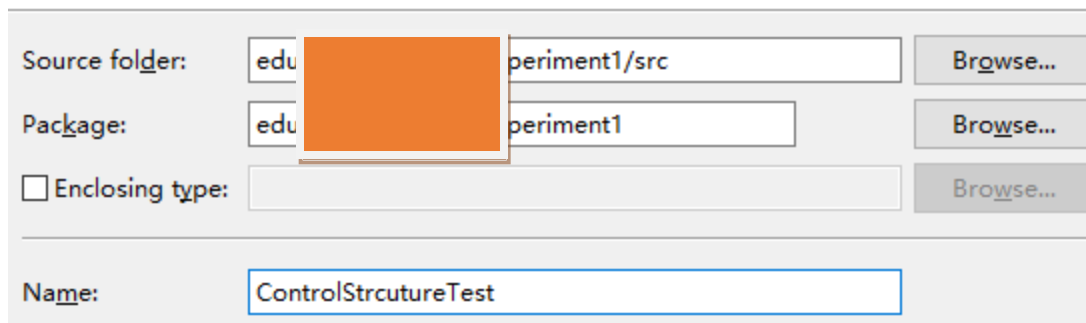
1271, 1368, 1644, 1912, 1949, 2000, 2017, 2020 等年份进行测试。

闰年判断规则如下：

①能被 4 整除且不能被 100 整除的为闰年。

②能被 400 整除的是闰年。

Step1: 增加ControlStrcutureTest类



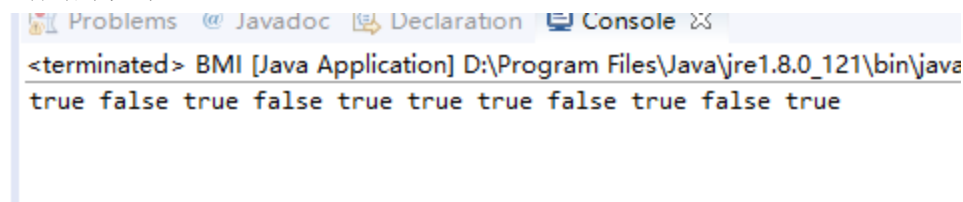
Step2: 添加checkLeapYear函数:

```
public static boolean checkLeapYear(int year){
    if(year%4==0 && year%100!=0||year%400==0)return true;
    else return false;
}
```

Step3: 在main中调用并运行观察结果

```
public static void main(String[] args){
    int i=0;
    int years[]={220, 618, 960, 1271, 1368, 1644, 1912, 1949, 2000, 2017, 2020};
    for(i=0;i<11;i++)System.out.printf("%b| ",checkLeapYear( years[i]));
}
```

运行结果如下:



7) 在 ControlStrcutureTest 中增加一个函数 printLeapYears, 该函数能够打印出开始年份到结束年份中所有的闰年; 在 mian 函数中调用该函数, 输出 2001-2050 这 50 年中所有的闰年。为 printLeapYears 函数增加一个参数, 使得该函数能够打印出开始年份

到结束年份中的第一个闰年；在 main 函数中调用该函数，输出 2001-2050 这 50 年中的第一个闰年。

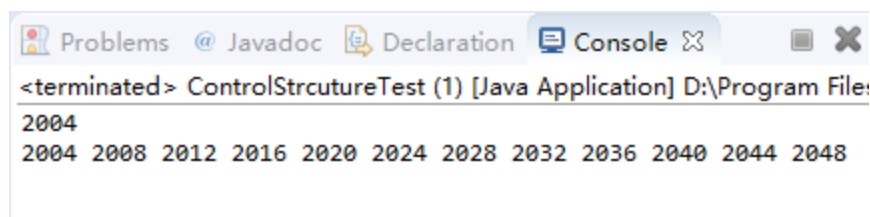
Step1: 添加函数如下

```
public static void printLeapYears(int begin, int end, boolean onlyFirst) {
    for (int year = begin; year <= end; year++)
        if (checkLeapYear(year)) {
            System.out.printf("%d ", year);
            if (onlyFirst) return;
        }
}
```

Step2: 在 main 中调用并运行观察结果

```
public static void main(String agrv){
    printLeapYears(2001,2005,true);
    System.out.println();
    printLeapYears(2001,2050,false);
}
```

运行结果如下:



```
<terminated> ControlStrcutreTest (1) [Java Application] D:\Program File:
2004
2004 2008 2012 2016 2020 2024 2028 2032 2036 2040 2044 2048
```

## 8) 在 ControlStrcutreTest 中增加一个如下猜数小游戏的函数。

Step1: 添加函数如下

```
public static void guessNumber (){
    System.out.println("给你一个1至100之间的整数,请猜测这个数");
    int realNumber=(int)(Math.random()*100)+1;
    int yourGuess=0; //你猜的数
    Scanner in=new Scanner(System.in);
    System.out.println("输入您的猜测:");
    yourGuess=in.nextInt(); //从键盘输入你猜的数
    while (yourGuess!=realNumber){
        if (yourGuess>realNumber){
            System.out.println("大了,请再猜:");
            yourGuess=in.nextInt(); //从键盘输入你猜的数
        }else if(yourGuess<realNumber){
            System.out.println("猜小了,请再猜:");
            yourGuess=in.nextInt(); //从键盘输入你猜的数
        }
    }
    System.out.println("恭喜!猜对了!");
}
```



Step2: 运行并观察结果



```

Problems @ Javadoc Declarat
<terminated> BMI [Java Application] D
给你一个1至100之间的整数,请猜测这个数
输入您的猜测:
50
大了,请再猜:
25
大了,请再猜:
12
恭喜!猜对了!

```

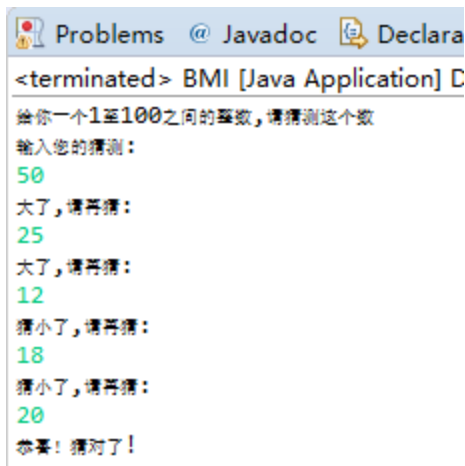
Step3, 程序可改进为 do-while 循环如下:

```

public static void guessNumber(){
    System.out.println("给你一个1至100之间的整数,请猜测这个数");
    int realNumber=(int)(Math.random()*100)+1;
    int yourGuess=0; //你猜的数
    Scanner in=new Scanner(System.in);
    System.out.println("输入您的猜测:");
    do{
        yourGuess=in.nextInt(); //从键盘输入你猜的数
        if (yourGuess>realNumber){
            System.out.println("大了,请再猜:");
        }else if(yourGuess<realNumber){
            System.out.println("猜小了,请再猜:");
        }
    }while (yourGuess!=realNumber);
    System.out.println("恭喜!猜对了!");
}

```

运行结果如下:



```

Problems @ Javadoc Declarat
<terminated> BMI [Java Application] D
给你一个1至100之间的整数,请猜测这个数
输入您的猜测:
50
大了,请再猜:
25
大了,请再猜:
12
猜小了,请再猜:
18
猜小了,请再猜:
20
恭喜!猜对了!

```

9) 在 ControlStrcutureTest 中增加 2 个函数, 根据不同的参数, 可实现打印如下六个用星号组成的三角形。

Step1. 分析题意可知, 应当增加的两个函数分别是 用于选择打印的三角形类型控制结构函数, 和一个用于实现打印动作的功能函数。

根据对题意的理解, 完成如下函数的代码:

1、ChoiceTriangular(int n);

```
public static void ChoiceTriangular(int n){
    switch(n){
        case 1:PrintTriangular(" ", "*", 1);break;
        case 2:PrintTriangular(" ", " * ", 1);break;
        case 3:PrintTriangular(" * ", " ", 2);break;
        case 4:PrintTriangular(" ", " * ", 2);break;
        case 5:PrintTriangular(" ", " * ", 2);break;
        case 6:PrintTriangular(" * ", " ", 1);break;
        default:System.out.println("错误参数");
    }
}
```

2、PrintTriangular(String left,String right,int mode);

```
public static void PrintTriangular(String left,String right,int mode){

    final int N=10;
    if(mode==1){
        for(int i=0;i<N;i++){
            for(int j=0;j<N-i;j++){
                System.out.printf("%s",left);
            }
            for(int j=0;j<i;j++){
                System.out.printf("%s",right);
            }
            System.out.println();
        }
    }else{
        for(int i=0;i<N;i++){
            for(int j=0;j<i;j++){
                System.out.printf("%s",left);
            }
            for(int j=0;j<N-i;j++){
                System.out.printf("%s",right);
            }
            System.out.println();
        }
    }
}
```

Step2.在主函数中调用并引导用户选择想要的三角形:

```
public static void main(String []argv){
    Scanner input=new Scanner(System.in);
    System.out.println("请输入参数");
    ChoiceTriangular(input.nextInt());
}
```

Step3.观察结果如下:



10) 在 ControlStrcutreTest 中增加函数 ChineseTrangle(int N), 利用二维数组实现 N(小于等于 10)阶杨辉三角的计算和显示。提示：代码 System.out.printf("%4s", "");可以打印 4 个空格；System.out.printf("%4d", 1);打印显示 4 个字符宽度的数值。

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1

```

Step1.增加 ChineseTrangle(int N) 函数如下：

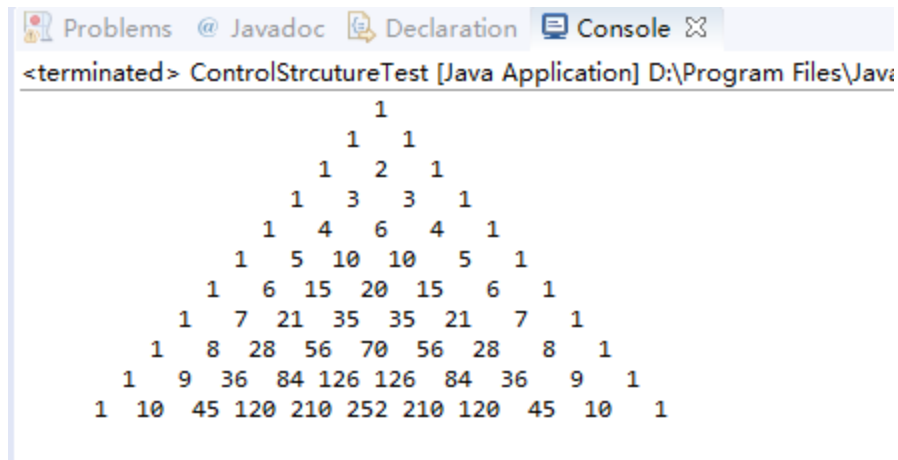
```
public static void ChineseTrangle(int N){
    int[][] number=new int[N][N];

    for(int i=0;i<N;i++)number[i][0]=1;

    for(int i=1;i<N;i++){
        for(int j=1;j<=i;j++){
            number[i][j]=number[i-1][j-1]+number[i-1][j];
        }
    }
    for(int i=0;i<N;i++){
        for(int j=0;j<N-i;j++)System.out.printf("%2s","");
        for(int j=0;j<=i;j++)System.out.printf("%4d",number[i][j]);
        System.out.println();
    }
}
```

Step2.在主函数中调用并观察结果：

```
public static void main(String []argv){
    //Scanner input=new Scanner(System.in);
    //System.out.println("请输入参数");
    //ChoiceTriangular(input.nextInt());
    ChineseTrangle(11);
}
```



```
<terminated> ControlStrcutureTest [Java Application] D:\Program Files\Java
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
```