

21 世纪高等学校规划教材

Java 程序设计 习题解答与实验指导

刘克成 郑珂 王晓 编著



中国电力出版社
www.infopower.com.cn

TP312/2565C

2007

21 世纪高等学校规划

Java 程序设计 习题解答与实验指导

刘克成 郑珂 王晓 编著



中国电力出版社
www.infopower.com.cn

内容提要

本书是《Java 程序设计简明教程》一书的配套教学参考书,内容全面、覆盖面广,旨在加深学生对教材内容的理解,提高学生的程序设计技巧,积累学生的程序调试经验,培养学生的动手能力。全书内容包括《Java 程序设计简明教程》一书的习题解答、上机实验项目、实验参考答案及综合实例。本书突出了《Java 程序设计简明教程》的知识重点,与理论教学紧密配合,适合作为学习《Java 程序设计简明教程》的配套教材,同时也可作为上机实验的独立教材。

图书在版编目(CIP)数据

Java 程序设计习题解答与实验指导 / 刘克成, 郑珂, 王晓编著. —北京: 中国电力出版社, 2007.8
21 世纪高等学校规划教材
ISBN 978-7-5083-5658-7

I. J… II. ①刘…②郑…③王… III. Java 语言—程序设计—高等学校—教学参考资料 IV. TP312
中国版本图书馆 CIP 数据核字 (2007) 第 088765 号

丛 书 名: 21 世纪高等学校规划教材
书 名: Java 程序设计习题解答与实验指导
出版发行: 中国电力出版社

地 址: 北京市三里河路 6 号

邮政编码: 100044

电 话: (010) 68362602

传 真: (010) 68316497, 88383619

服务电话: (010) 58383411

传 真: (010) 58383267

E-mail: infopower@cepp.com.cn

印 刷: 航远印刷有限公司

开本尺寸: 185mm×260mm

印 张: 11.5

字 数: 276 千字

书 号: ISBN 978-7-5083-5658-7

版 次: 2007 年 8 月北京第 1 版

印 次: 2007 年 8 月第 1 次印刷

印 数: 0001—4000 册

定 价: 18.00 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

前 言

上机实验是学习 Java 程序设计课程必不可少的实践环节,它是掌握基础知识、提高程序设计技巧和积累程序调试经验的基本途径。本书以实际应用为目的,组织经验丰富的教师,并结合课堂教授的内容和进度编写,它是《Java 程序设计简明教程》的配套教材,旨在加深学生对教材内容的理解,提高学生的程序设计技巧,积累学生的程序调试经验,培养学生的动手能力。

本书通过手把手的指导和按图示操作进行讲解,直观生动,让读者能够快速掌握 Java 编程的基本方法,提高学习效率,快速入门。在内容设计上,本书涵盖了《Java 程序设计简明教程》各个章节的所有知识点,与教学内容联系紧密。其中第 1 章给出了《Java 程序设计简明教程》一书中所有习题的参考答案;第 2 章给出了上机实验的目的和内容要求;第 3 章则给出了上机实验的实验结果和参考答案;第 4 章编写了 4 个综合实例,包括单机模式、C/S 模式和 B/S 模式的多种实例。在实验形式的安排上采取循序渐进的方式,为每个实验都安排了不同难度的实验题目,引导学生发现和解决问题,包括创新性实验、实践性实验、设计性实验和综合实例。

本书由刘克成、郑珂、王晓任主编,全书由刘克成、郑珂、王晓、于彬、肖飞、张松娟、张秋红、胡冰、赵东江、骆小红、张菊艳共同编写。

由于作者的水平和经验有限,疏漏之处在所难免,恳请读者不吝赐教。

编 者
2007 年 8 月

目 录

前 言

第 1 章 习题解答	1
1.1 Java 概述和开发工具	1
1.2 Java 语言基础	3
1.3 面向对象程序设计	10
1.4 Java 的核心 API	16
1.5 集合	19
1.6 异常	20
1.7 多线程	22
1.8 输入/输出处理	25
1.9 图形用户界面	32
1.10 Java 数据库连接	35
1.11 Java 网络编程	39
1.12 Java Web 应用	44
第 2 章 上机实验	47
2.1 【实验 2-1】Java 程序的编辑、编译、运行	47
2.2 【实验 2-2】Eclipse 的安装、配置和使用	48
2.3 【实验 2-3】Java 语言基础	48
2.4 【实验 2-4】类与面向对象	52
2.5 【实验 2-5】接口和包	54
2.6 【实验 2-6】数组和字符串	55
2.7 【实验 2-7】异常处理	55
2.8 【实验 2-8】多线程	57
2.9 【实验 2-9】输入/输出系统	59
2.10 【实验 2-10】Java 图形用户界面	60
2.11 【实验 2-11】使用 JDBC 连接数据库	60
2.12 【实验 2-12】使用 Hibernate 连接数据库	61
2.13 【实验 2-13】网络编程应用	61
2.14 【实验 2-14】Java Web 程序的开发	62
第 3 章 实验参考答案	63
3.1 【实验 3-1】Java 程序的编辑、编译、运行	63
3.2 【实验 3-2】Eclipse 的安装、配置和使用	64
3.3 【实验 3-3】Java 语言基础	69

3.4 【实验 3-4】类与面向对象	71
3.5 【实验 3-5】接口和包	75
3.6 【实验 3-6】数组和字符串	80
3.7 【实验 3-7】异常处理	81
3.8 【实验 3-8】多线程	84
3.9 【实验 3-9】输入/输出系统	88
3.10 【实验 3-10】Java 图形用户界面	92
3.11 【实验 3-11】使用 JDBC 连接数据库	95
3.12 【实验 3-12】使用 Hibernate 连接数据库	101
3.13 【实验 3-13】网络编程应用	105
3.14 【实验 3-14】Java Web 程序的开发	108
第 4 章 综合实例	114
4.1 【实例 4-1】简单计算器	114
4.2 【实例 4-2】学生信息管理系统	126
4.3 【实例 4-3】C/S 模式的聊天室	147
4.4 【实例 4-4】Web 通讯录	162
参考文献	176

第1章 习题解答

1.1 Java 概述和开发工具

1. Java 技术体系主要由哪三部分组成？它们分别用于什么场合？

答：Java 技术体系主要有：

- (1) J2EE——Java 2 Enterprise Edition，可扩展的企业级应用的 Java 2 平台。
- (2) J2SE——Java 2 Standard Edition，用于工作站、PC 的 Java 2 标准平台。
- (3) J2ME——Java 2 Micro Edition，嵌入式 Java 2 电子消费平台。

2. Java 平台分为哪几部分？它们的相互关系如何？

答：Java 平台有两个主要部分：

- (1) Java 虚拟机——The Java Virtual Machine (JVM)。
- (2) Java 程序设计接口——The Java Application Programming Interface (JAPI)。

Java 虚拟机 (JVM) 是一个软件模拟实现的虚拟计算机，是 Java 跨平台的基础，它可以解释并运行 Java 程序。JVM 在 Java API 中包含，Java API 是一个庞大的软件组件集合。它提供了许多实用的功能，比如用于图形设计的 AWT。Java API 把相关的类和接口组织成一个类库，这些类库称为包 (packages)。要运行 Java 程序只需要安装 JVM 即可，但是要开发 Java 程序就必须安装所必需的 Java API 和 JVM。

3. 环境变量 classpath 和 path 的作用分别是什么？

答：环境变量 classpath 的作用是告诉操作系统程序所需要的类库所放置的位置。Path 的作用是告诉操作系统可执行文件的所在位置。

4. 简述在 JDK 环境中 Java 程序的编辑、编译、运行的过程。

答：在 JDK 环境中 Java 程序的编辑、编译、运行的过程如下。

(1) 准备工作。要进行 Java 程序的开发，必须使用两种工具。

① Java 的开发包 JDK。

② 一个文本编辑器。

(2) 编写 Java 源程序。其过程如下：

① 打开记事本，在编辑窗口输入代码。

② 保存 Java 源程序。Java 源程序的扩展名为 .java，文件的名称和程序中类的名称保持完全一致，进行文件的保存时要注意文件名称的大小写，另外在保存窗口中保存类型必须选为“所有文件”。

(3) 编译源程序。进入 MS-DOS 窗口，把目录转换到 Java 源文件所在的目录，然后输入相应的编译命令“javac”，后面还要输入文件名。

注意文件名称不能写错，同时要注意区分大小写，而且要带上扩展名 .java。如果编译成

功，将会生成字节码文件 `NumberAdd.class`。

(4) 运行字节码文件。经过编译之后，生成的字节码文件就可以由 Java 的解释器解释执行，在 MS-DOS 窗口下输入 Java 程序运行命令和字节码文件名，就可以运行程序了。

5. 简述 JVM 和 JRE 的含义、作用及区别。

答：JRE 为 Java RunTime Environment 的简称，它包括两部分：Java Runtime Environment 和 Java Plug-in。可以在 JRE 上运行、测试和传输 Java 应用程序。它包括 Java 虚拟机、Java 平台核心类和一些支持文件。它不包含开发工具——编译器、调试器和其他工具。因此要开发 Java 程序除了 JRE 之外，还要安装其他组件，这些组件都包含在 `jdk1.5.0_01` 目录内。

Java 虚拟机 (JVM) 是一个软件模拟实现的虚拟计算机，是 Java 跨平台的基础，它可以解释并运行 Java 程序。Java 的源文件 (`.java`) 经编译生成字节码文件 (`.class`)，然后由 JVM 解释执行字节码文件。

6. 在 Eclipse 中设计简单的 Java 应用，能够在屏幕上输出 “Hello world!”。

答：在 Eclipse 中创建该 Java 应用的步骤如下：

(1) 新建一个空的 Java 项目。

选择【File】|【New】|【Project】命令。在弹出的对话框中选择【Java】|【Java Project】选项命令，如图 1-1 所示。

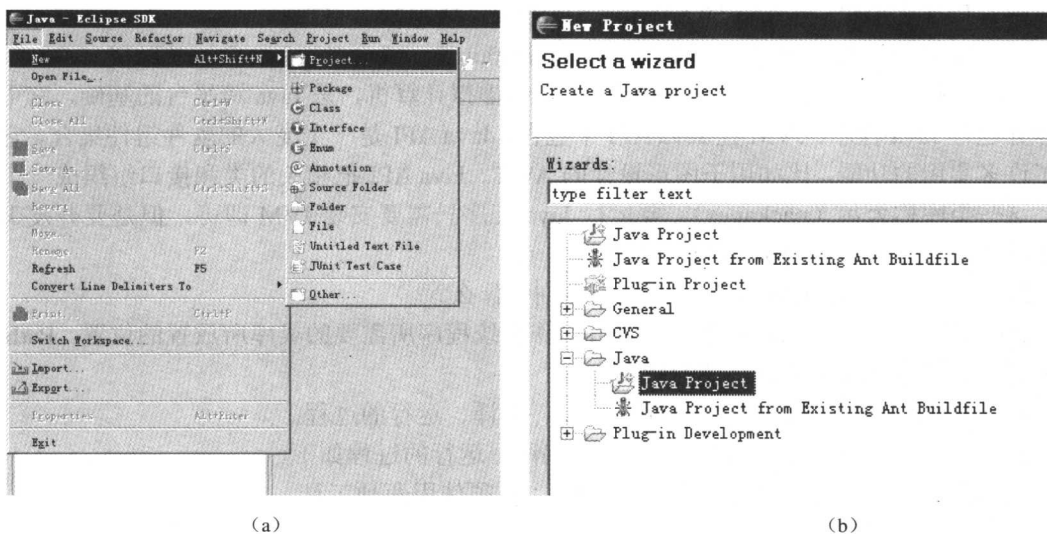


图 1-1 创建 Java 项目

单击 “Next” 按钮，在弹出的对话框中，在【Project Name】文本框中输入项目名称，比如 `MyProject`，在项目布局窗口【Project Layout】中选择 `use project folder as a root for sources and class files`，然后单击 `Finish` 按钮。

(2) 新建一个 Java 类。

在导航视图中选择要添加程序的 Java 项目，右击后在弹出的快捷菜单中选择【New】|【Class】命令，进入建立 Java 类向导。

在如图 1-2 所示的窗口中输入类名 `FirstClass`，并选中 “`public static void main (String [] args)`” 来自动创建 `main ()` 方法”。

单击 Finish 按钮，在编辑器中输入一行代码：“System.out.println (“Hellow World!”)”。

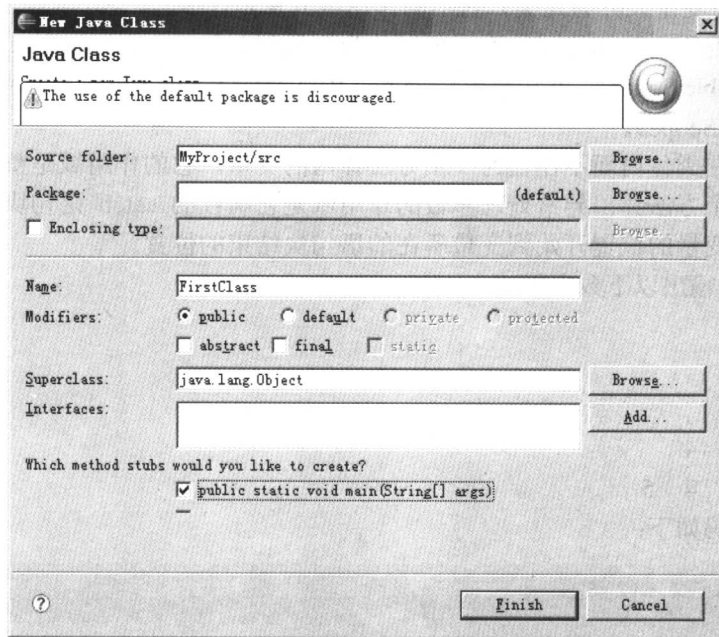


图 1-2 设置类名

(3) 运行项目。单击 Run 按钮旁的小三角形，选择【Run as】|【java Applicatin】来运行该程序。

1.2 Java 语言基础

1. 简述 Java 程序的基本结构组成。

答：一个基本的 Java 应用程序的基本构成包括：

(1) 类代码块。也可称为类体，所有的程序都需要一个类作为程序的主要结构。在类代码块中声明各个变量和方法。

(2) 变量。是程序执行所需要的数据，需要在类体或者方法体内声明和进行相应的设置。

(3) 方法代码块。是程序的操作结构。

(4) 注释。程序的解释部分，JVM 将忽略这些代码。

(5) 语句块。在 Java 程序中使用“{”和“}”来定义代码块。

(6) 语句。使用分号作结尾。

2. 组成 Java 语句的五种主要元素是什么？

答：语句主要由五种元素组成：标识符、关键字、数据类型、运算符和分隔符。这 5 种元素有着不同的语法含义和组成规则，它们互相配合，共同完成 Java 语言的语意表达。

3. 下列哪些标识符是不合法的？

Abc1	123ab	\$abc	a@*\$	_abc	123_true
t	this	Cl	priva	catch	native

答：a@*\$、123_、true、this、catch 是非法的标识符。

4. 在 Java 中变量共分为几类？它们的作用域分别是什么？

答：变量根据所声明的位置的不同可以分为：局部变量（Local Variable Scope）、成员变量（Member Variable Scope）、方法参数（Method Parameter Scope）和异常处理参数（Exception Handler Parameter Scope）。

成员变量是在类体内而不在任何一个方法体内的变量，它的作用域是整个类体；方法参数的作用域在整个方法体；异常处理参数的作用域是它所在的 catch 语句块；局部变量的作用域是从声明该变量的位置开始到变量所在的语句块结束的位置。

5. 编写程序输出以下数字形式。

(1)

```
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

答：参考代码如下：

```
public class Xiti2_6_1
{
    public static void main(String args[])
    {
        final int size=5;
        int i,j,n,t,m=0;
        n=size;
        int a;
        for(i=0;i<n;i++)
        {
            for(j=0;j<=i;j++)//输出各行的值
            {
                a=j+1;
                System.out.print(a+" ");
            }
            System.out.println();
        }
    }
}
```

(2)

```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
```

答：参考代码如下：

```
public class Xiti 2_6_2
{
    public static void main(String args[])
    {
```

```

    {
        final int size=4;
        int i,j,n,t,m=0;
        n=size;
        int a;
        for(i=0;i<n;i++)
        {
            m=2*i+1;//确定一行中值的个数
            for(t=0;t<n-i;t++)//输出一行中前面所对应的空格
            {
                System.out.print(" "+" ");
            }

            for(j=0;j<m;j++)//输出各行的值
            {
                if(i>=j) a=j+1;
                else a=m-j;
                System.out.print(a+" ");
            }
            System.out.println();
        }
    }
}

```

6. 编写程序，求 1~100 的和。

答：参考代码如下：

```

public class Xiti2_7
{
    public static void main(String args[])
    {
        int sum=0;
        for(i=1;i<=100;i++)
        {
            sum=sum+i;
        }
        System.out.println("1~100 的和是: "+sum);
    }
}

```

7. 把例 2-16 转换为 while 循环和 do-while 循环。

答：转换为 while 循环时，参考代码如下：

```

public class Xiti2_8_1
{
    public static void main(String args[])
    {
        int i=1,j,n=10,s=0;
        while(i<=n)
        {

```

```

        int k=1;j=1;
        while(j<=i)//求出 i 的 i 次方的值
        {
            k=k*i;
            j=j+1;
        }
        i=i+1;
        s=s+k;
    }
    System.out.println("s="+s);
}

```

转换为 do-while 循环时, 参考代码如下:

```

public class Xiti2_8_1
{
    public static void main(String args[])
    {
        int i=1,j,n=10,s=0;
        do
        {
            int k=1;j=1;
            do //求出i的i次方的值
            {
                k=k*i;
                j=j+1;
            } while(j<=i)
            i=i+1;
            s=s+k;
        } while(i<=n)
        System.out.println("s="+s);
    }
}

```

8. 简述 for 循环、while 循环、do-while 循环的区别。

答: 三种循环语句的比较:

(1) for 语句和 while 语句先判断条件, 后执行语句, do-while 语句则是先执行语句再判断条件, 因此 do-while 语句的循环体至少执行一次。

(2) 必须在 while 语句和 do-while 语句之前对循环体变量赋初值, 而 for 语句可在表达式中对循环变量赋初值。

(3) 在循环次数已经确定的情况下, 通常用 for 语句; 而对于循环次数不确定只给出循环结束条件的问题, 通常用 while 语句解决。

9. 如果 A 的因子和是 B, 而 B 的因子和是 A, 则称 A 和 B 是一组亲密数对。A 的因子包括 1, 但不包括 A 本身。编程求 1000 以内的亲密数对。

答: 参考代码如下:

```

public class Xiti2_9{
    static int sushu(int n) {
        //判断是否为素数
        int i,j=0;
        for(i=2;i<=n;i++)
            if(n%i==0) j=j+1;
        if(j==1) return 1;
        else return 0;
    }
    public static int yinzihe(int n) {
        //求一个数的因子和
        int i;
        int a=1;
        if(sushu(n)==0)
        {
            for(i=2;i<n;i++)
                if(n%i==0) a=a+i;
        }
        return a;
    }
    public static void main(String args[])
    {
        int i;
        int a,b,c;
        System.out.println("1000以内的亲密数对有: ");
        for(i=2;i<1000;i++)
        {
            a=i;
            b=yinzihe(a);
            c=yinzihe(b);
            if(c==a) System.out.print("(" + a + ", " + b + ") " + " ");
        }
    }
}

```

10. 设 int y, m, d 分别表示一个日期中的年月日, 试编写程序求:

- (1) 对于任意 3 个整数组成的日期, 判断是否是一个合法的日期值。
- (2) 给定一个日期值, 计算若干天后的日期数。
- (3) 如果有两个日期, 计算它们相隔的天数。

答: (1) 对于任意 3 个整数组成的日期, 判断是否是一个合法的日期值。答: 参考代码如下:

```

public class Xiti2_10_1 {

    public static void main(String args[]) {
        int y=2010,m=2,d=29;
        if((d>0)&(d<=31)) {
            if((m<=0)|(m>12))
                System.out.print(y+"-"+m+"-"+d+"不是一个合法日期值");
            else if((m==4)|(m==6)|(m==9)|(m==11)) {

```

```

        if(d==31)
            System.out.print(y+"-"+m+"-"+d+"不是一个合法日期值");
        else System.out.print(y+"-"+m+"-"+d+"是一个合法日期值");
    }
    else if(m==2) {
        if((y%400==0)|((y%100!=0)&(y%4==0))) {
            if(d<=29)
                System.out.print(y+"-"+m+"-"+d+"是一个合法日期值");
            else
                System.out.print(y+"-"+m+"-"+d+"不是一个合法日期值");
        }
        else {
            if(d<=28)
                System.out.print(y+"-"+m+"-"+d+"是一个合法日期值");
            else
                System.out.print(y+"-"+m+"-"+d+"不是一个合法日期值");
        }
    }
    else
        System.out.print(y+"-"+m+"-"+d+"是一个合法日期值");
}
else
    System.out.print(y+"-"+m+"-"+d+"不是一个合法日期值");
}
}

```

(2) 给定一个日期值, 计算若干天后的日期数。其参考代码如下:

```

public class xiti2_6_2{
    public static void main(String args[]){
        int y=2006,m=1,d=20,n=100;
        int y1,m1,d1;
        int i=0,j=0,t=d;
        y1=y;
        m1=m;
        d1=d;
        for(i=1;i<=n;i++){
            d1=d1+1;
            if((m1==4)|(m1==6)|(m1==9)|(m1==11)) {
                if(d1==31) {
                    m1=m1+1;
                    d1=1;
                }
            }
            else if(m1==2) {
                if((y1%400==0)|((y1%100!=0)&(y1%4==0))) {
                    if(d1==30) {
                        m1=m1+1;
                        d1=1;
                    }
                }
            }
        }
    }
}

```

```

        else {
            if(d1==29) {
                m1=m1+1;
                d1=1;
            }
        }
    }
    else if(m1==12) {
        if(d1==32) {
            y1=y1+1;
            m1=1;
            d1=1;
        }
    }
    else{
        if(d1==32) {
            m1=m1+1;
            d1=1;
        }
    }
}

System.out.print("与日期"+y+"-"+m+"-"+d+"相隔"+n+"天的日期是:
                  "+y1+"-"+m1+"-"+d1);
}
}

```

(3) 如果有两个日期, 计算它们相隔的天数。其参考代码如下:

```

public class xiti2_6_3{
    static int t(int y,int m,int d){
        int i,t=d;
        for(i=1;i<m;i++){
            if((i==4)|(i==6)|(i==9)|(i==11)) t=t+30;
            else if(i==2){
                if((y%400==0)|((y%100!=0)&(y%4==0)))
                    t=t+29;
                else t=t+28;
            }
            else t=t+31;
        }
        return t;
    }
    static int t3(int y,int y1) {
        int i,t3=0;
        for(i=y;i<y1;i++){
            if((i%400==0)|((i%100!=0)&(i%4==0)))
                t3=t3+366;
            else t3=t3+365;
        }
        return t3;
    }
}

```

```

    }
    public static void main(String args[]) {
        int y=2006,m=4,d=12;
        int y1=2009,m1=4,d1=12;
        int t=0;
        int i,t1=0,t2=0,t3=0;
        t1=t(y,m,d);
        t2=t(y1,m1,d1);
        t3=t3(y,y1);
        t=t2+t3-t1;
        System.out.print(y+"-"+m+"-"+d+"与日期"+y1+"-"+m1+"-"+d1+"相隔天数
                           是: "+t);
    }
}

```

1.3 面向对象程序设计

1. 简述面向对象的主要特征。

答：面向对象的主要特征是封装性、继承性和多态性，衡量某一种程序设计语言是否是面向对象的程序设计语言，主要看它是否具有以上三种特性。

2. 一个消息由哪三部分组成，举例说明。

答：一个消息由三部分组成：接收消息的对象，接收对象要采取的方法，方法需要的参数。消息模式定义了一个对象和其他对象的交互能力和处理能力，摆脱了传统的控制结构。发送消息的对象不需要关心接收对象的内部数据结构，只要知道它的消息模式。

软件对象与其他对象进行交互与通信是通过发送给其他对象消息来实现的。当对象 A 想让对象 B 来执行一个 B 中的方法，对象 A 就会发消息给对象 B。消息模式：B.method(parameter)。

3. 简述类和对象之间的联系和区别。

答：类是对象的模板。即类是对一组有相同数据和相同操作的对象的定义，一个类所包含的方法和数据描述一组对象的共同属性和行为。类是在对象之上的抽象，对象则是类的具体化，是类的实例。

4. 在对象的声明周期内对象在内存中有哪些变化？特别是在创建阶段对象的声明和实例化的区别。什么是引用类型？

答：对象的声明并没有定义对象本身，而仅仅是用一个变量指代对象，声明并不为对象分配内存空间，而只是分配一个引用空间；对象的引用类似于指针，是 32 位的地址空间，它的值指向一个中间的数据结构，它存储有关数据类型的信息以及当前对象所在的堆的地址，而对于对象所在的实际的内存地址是不可操作的，这就保证了安全性。

对象的实例化，为对象分配内存空间。一个类的不同对象分别占据不同的内存空间。new 标识符为对象动态分配（即在程序运行时分配）内存，并为其返回一个引用。该引用是由 new 分配给对象的内存的地址。然后该引用被保存在变量中。

5. 对象的状态和行为是对象的主要属性，前者对应类的什么？行为又称为对象的操作，对应着类的什么？类的定义包括的声明是什么？

答：类描述了一组有相同特性（属性）和相同行为（方法）的对象。对象的状态和行为是对象的主要属性，前者对应类的属性，后者对应着类的方法。

类的声明包括类名、供其他对象使用的访问类型以及它从中继承状态和行为的父类（或接口）。类的声明要指明类的访问权限、类名、父类和要执行的接口等，它的一般格式为：

[类的修饰符] class 类名[extends 父类名称] [implements 接口名称列表]

6. 为什么说构造函数是一种特殊的方法？特殊在哪里？构造函数什么时候执行？被谁调用？

答：构造方法是一种特殊的方法。Java 中的每一个类都有构造方法，用来初始化该类的一个新的对象。构造方法具有和类名相同的名称，而且不返回任何数据类型，甚至也不返回 void 数据类型。这是因为类的构造方法隐含的返回类型是类本身。

构造一个类的实例对象，通过调用构造方法来实现。构造方法只有在实例化对象的时候放在标识符 new 后边被调用，其他时候是不允许被调用的。

7. 创建一个类，在其中定义一个无参数的构造方法，输出一些信息。再创建一个该类的对象，观察程序运行结果。

答：参考代码如下：

```
public class InitDemo {
    private int i;
    private String str;
    public InitDemo(){}
    public static void main(String[] args) {
        InitDemo demo=new InitDemo();
        System.out.println("int i init:"+demo.i);
        System.out.println("String str init:"+demo.str);
    }
}
```

8. 创建一个类，该类包含 int、double、char 和 String 类型的成员变量且未被初始化，编写一个 main() 方法，输出这些成员变量的值，观察 Java 是否执行了默认的初始化操作。

答：参考代码如下：

```
public class InitDemo {
    private int i;
    private double m;
    private char a;
    private String str;
    public static void main(String[] args) {
        InitDemo demo=new InitDemo();
        System.out.println("int i init:"+demo.i);
        System.out.println("double m init:"+demo.m);
        System.out.println("char a init:"+demo.a);
        System.out.println("String str init:"+demo.str);
    }
}
```