

红字都是笔记！（自己写的）

紫色是书上第几页

黑色是重点内容！

前提：

一、编程基础

（1）基本数据结构和字长

P15 整型 浮点型 字符型... 占用字节 int 4 字节 char 2 字节 1 个英文字母占 2 个字节 1 个中文字符占 2 个字节 浮点数默认 double

（2）Java 程序结构（了解）**P12** 单选

1. java 是无格式的纯文本文件，可以用任何文本编译器编写
2. Java 文件名必须与类名相同，大小写也要一致
3. 关键字 class 定义新类，类名跟随其后
4. 一个 java 可以有多个类，每个类有多个方法，但是必须要有一个主类
5. Public static void main(String[] args) 是主方法 一个 java 只能有一个 main
6. 任何方法中可以有多条语句
7. 每个语句必须用;结尾
8. 为了增加可读性，可以加入一些注释行

（3）Java 编译和运行的过程(什么是 Java 虚拟机、什么是字节码、什么是平台无关性) **Java**

虚拟机功能：

1. 通过 **ClassLoader** 寻找和加载 class 文件
2. 解释字节码成为指令并执行，提供 class 文件的运行条件
3. 运行期间回收垃圾
4. 提供与硬件交互的平台

虚拟机是真实机器用软件模拟硬件的技术。代码储存在.class 文件中。每个文件最多有一个 **Public** 类。（什么是虚拟机）

字节码是用来在 java 环境下运行的高度优化的指令集。**JVM** 就是一个字节码解释器（什么是字节码）

因为有虚拟机的存在，**Java** 能够实现可移植性。虚拟机就好像是一个基本平台，java 文件都能够在虚拟机上运行。每个 java 解释器都能够执行 **JVM**（平台无关性）

P5 最上面一段，图上文字的编写

（4）掌握 OOP 编程的三大特点：封装、继承、多态 （面向对象编程）

- (5) 掌握 Java 的虚拟机（软件平台，满足一次编写处处运行）的概念，和平台无关性
- (6) Java 标志符合法性 单选

P17 6 点

1. 必须以字母、下划线或者美元符号开头
2. 不能是关键字，中间不能有空格

3. 不能使用操作符如+ -
4. 严格区分大小写
5. 长度没有限制，不能过长
6. 不能是 true false null

(7) 常用字符串处理函数

P52最常用的3个（length indexOf charAt）表 2-14 Public

int length()返回字符串长度

Public char charAt(int index) 返回在 index 位置的字符Public int

indexOf (String str) 返回 str 第一次出现的位置三、面对对象编

程（编程题、占比最大）作业、例题

(1) 掌握类的定义：类内包含的基本的例如 protect 成员变量会给的，成员变量

定义基本的成员变量和成员方法：如 get ()， set () 方法，显示，计算等基本处理构造方法：有参数的

创建对象：测试类（public、test），在测试类内有 main 函数用 new 关键字去创建类的对象

掌握构造函数的作用、特点（P69 1,2,3,4,5）

1. 构造方法的方法名与类名相同
2. 构造方法没有返回值，也不能写 void
3. 构造方法的主要作用是完成对类的对象的初始化
4. 构造方法一般不能够由编程人员调用，而是用 new 来调用
5. 创建一个类对象的同时，系统会自动调用类的构造方法为新对象的初始化

(2) 掌握子类的继承，子类的定义，子类中构造方法的继承（子类的构造方法如何调用父类的构造方法）

子类继承用 extends 关键字来实现。（仅支持单一继承）

新定义的类就是子类，可以从父类继承所有的非私有(private)的属性和方法

子类不能够直接访问父类的私有成员，可以通过非私有的方法访问到父类的私有成员，达到目的。也就是 get 和 set 函数的作用。

调用父类的构造方法使用 super 关键字。（1.调用父类中指定参数的构造方法。2.引用父类的成员）

(3) 理解方法的重载，重写的概念和应用

方法的重载（要求参数列表不同）和覆盖（方法相同，区别在于权限，权限可以扩大不能缩小）

P89

方法的重载指在一个类中定义了多个名字相同的方法，这些方法有不同的参数表，实现不同的功能。注意点：

1. 方法的返回值类型对方法的重载没有影响，返回值的类型不能够区分方法，因为方法可以没有返回值。

2. 重载方法之间是以所带参数的个数以及对应的参数类型来区分的。

方法的覆盖指的是子类中重新定义一个与父类一样的成员方法。与父类有相同的格式（名称， 类型， 参数个数和顺序， 返回值）

（4） 理解接口的定义（`interface`），类实现接口（`class implemence`）和定义（实现抽象方法，写出方法体），接口的多态性（运用），（都是抽象方法所以方法体要写出来）

接口的定义是用 `interface` 实现的：

```

Interface Person{
    String name ="sb"; Int
    num =10;
    Public abstract tell();
}

用类实现接口： implements : Class

Student implements Person{
    Public String
        tell(){ Return xxx;
    }
}

```

P88 例 3.19(接口作为数据类型实现！)

(5) 理解静态变量，静态方法的定义和用于 Static 属于整个类的，可以直接用类名调用
(不可以调用其他普通成员的方法，只能用自己的)

P73

Static String country = "china"; 在其他的 共享 country 是 china

2. I/O 流操作（编程题）作业、例题

(1) 从键盘输入一些数据（整型 Scanner、字符串） 对数据的合法性做出判断

```

if (scanner.hasNextInt()) {
    Num= scanner.nextInt();break;
}
Else{ xxxx}
Next XXX    获取下一个输入项。

```

(2) 重点：使用字符流（区分什么叫字节流：处理一个个字节）（1 个字符=2 个字节）
对文件基本操作（fileReader buffireader :.....)

字节流（InputStream 和 OutputStream）字符

流（Reader 和 Writer）

FileReader 是从一个文件中读取字符流主要的类. 读

取一个文本文件内容的操作

```
Char c[]= new char[n]
```

```
FileReader fr=new FileReader ("xx.text") Int num
```

```
= fr.read(c);
```

```
String str = new String (c,0,num)
```

```
FileWriter fw= new FileWriter("readme.text",ture);
```

```
Fw.writer("hello");
```

String str;
buf=new BufferedReader (new InputStreamReader (System.in));
str=buf.readLine(); //将输入的文字指定给字符串变量 str 存放
BufferedReader 和 BufferedWriter 类实现了 Reader 类和 Writer 类的常用的读取数据和写入数据的方法,但是又有特殊的方法:

public String readLine(): BufferedReader 的方法, 读取一行字符串。
public void newLine(): BufferedWriter 的方法, 写入回车换行符。

【例 7.6】 利用缓冲输出流向文件中写入数据内容,再通过缓冲输入流将文件的信息分行显示。

```
//BufferRW.java
import java.io.*;

public class BufferRW {
    public static void main(String args[]) throws IOException
    {
        String content[]={"你好","好久不见","常联系"};
        try
        {
            BufferedWriter bout=new BufferedWriter(new FileWriter("wordtest.txt"));
            BufferedReader bin=new BufferedReader(new FileReader("wordtest.txt"));
            for(int i=0; i<content.length; i++){
                bout.write(content[i]); //将字符串数组内容写入到缓冲输出流中
                bout.newLine(); //将回车换行写入到缓冲输出流中
            }
            bout.close();
            String str= null;
            while ((str=bin.readLine())!=null)
            {
                System.out.println(str); //在显示器上输出
            }
            bin.close();
        }
        catch (IOException ioe)
        {
            System.out.println("错误!" + ioe);
        }
    }
}
```

本例中,分别创建了缓冲输入流对象 bin 和缓冲输出流对象 bout,在使用 write()方法写入字符数据时,没有写入回车换行符号,后面使用 newLine()方法写入回车换行符号。在

- (3) 异常处理机制 (try 代码段、catch 捕捉) IOException
- (4) 区分字节字符流

3. 集合框架 (编程题) 作业、例题

Set List Map 特点

Set 继承 Collection 不允许出现重复元素

List 继承 Collection 允许有重复, 引入位置索引

Map 与 Collection 接口无关, 是键-值对, 而不是单个的元素数据的增删查改和遍历

- (1) 创建基本 List 集合的类的对象 (arraylist 最基本)
- (2) 写一个循环, 添加集合的元素 (add)
- (3) 通过迭代器遍历并输出 (next)

```
Public Static void main(String[] args){  
    ArrayList<String> list = new ArrayList<String>(); List .add(1);  
    List.add(2);  
    Iterator <String> iterators = list.iterator();  
    While(iterators.hasNext()){  
        String result = iterators.next(); System.out.println(result);  
    }  
}
```

4. 数据库

- (1) 理解使用数据库的步骤 (4 个);

P279 加载驱动 创建连接对象 执行 sql 语句 形成结果

1. 加载驱动: class.forName()

MySQL, SqlServer, Oracle

2. 创建连接对象

出现 commit connection

3. 执行 sql 语句

executeQuery()

execute()

executeUpdate()

4. resultset

- (2) 掌握 getConnection () 中各个参数的含义

P280 url 协议、主机名、端口号、访问的数据库名 user 用户名 psw 密码

getConnection(url, user, password)

如:


```
String url="jdbc:microsoft ....."
```

```
String user="sa"
```

String password= "sa" 5.多

线程

两种常用实现方法

P318/P321

看书!

6. 网络通信

(1) 掌握网络通信的几种方式

两大类 (4 小种)

使用 url (直接访问) / 使用 url connection 使

用 socket: 流逝、数据

第十章小标题

使用 url 访问网络资源, 使用 sockert 实现网络通行

(3) 理解 TCP socket 实现 C/S 通信-处理流程 (客户端服务器端是对称的) -非常简单

```

import ? java.net.*;
import ? java.io.*;
class Homme21 {
    public Homme21() {
        try{
            ServerSocket ss = new ServerSocket(8629);
            System.out.println("服务器端已运行，等待客户端连接: ");
            Socket socket = ss.accept();
            BufferedReader in = new BufferedReader(new InputStreamReader
(socket.getInputStream()));
            PrintWriter out = new PrintWriter(socket.getOutputStream(),true);
            BufferedReader br = new BufferedReader(new InputStreamReader(?
System.in));
            String line;
            while (true) {
                line = (in.readLine().trim());
                if(line.equals("BYE")){
                    out.println("YE");
                    out.close();
                    in.close();
                    socket.close();
                    br.close();
                    ss.close();
                    break;
                }
                System.out.println("客户输入为:" + line);
                System.out.println("请输入回答:");
                line = br.readLine();
                out.println(line);
            }
        }
        catch(IOException e) {}
    }
    public static void main(String[] args) throws Exception {
        new Homme21();
    }
}

```

7. GUI（概念理解）

（1）理解 GUI 编程的步骤（章节标题）

1.创建窗体， 2.添加常用组件， 布局管理器 3.事件处理

（2）事件处理的要素（事件源、事件对象、事件监听器、） P189 事

件处理机制 P190 6.5.1

1. 导入 java 包

2. 创建事件源

3. 给事件源注册事件监听器

4. 根据受到的事件类型，实现事件监听器的对应方法

对程序的阅读和理解 4*5=20（例：解释某一句话的作用） 编

程题 50 分

单选*10