day01

1. DOS命令:
   1. dir: 列出当前目录下的所有文件与文件夹名称
   2. cd: 进入指定的文件夹
   3. cd.. : 返回上一层目录
   4. cd\ : 返回到根目录(返回到盘符)
   5. 切换盘符: d:
   6. cls: 清屏
2. Java语言的特点:
   1. 面向对象的程序开发
   2. 开源
   3. 跨平台
   4. 安全
3. JDK与JRE是什么?
   1. JVM: Java虚拟机,用来加载.class文件到内存中运行,可以实现Java源文件在不同的操作系统上运行, JVM不是跨平台的
   2. JRE: 运行环境, 包含了JVM
   3. JDK: 开发环境, 包含了JRE
4. path环境变量:
   1. 配置.exe或者.bat可执行文件的路径,可实现可执行文件在任意的目录中运行
5. 关键字:  
   在java中具备特殊含义的英文单词,都是由小写字母组成
6. 标识符:
   1. 就是命名(俗称起名字)
7. 什么是常量, 可用来做什么?
   1. 常量是在程序运行中,不可改变的值
   2. 作用:
      1. 给变量赋值
      2. 可以参与运算
      3. 用来记录一个固定不变的值
8. 什么是变量, 可用来做什么?
   1. 变量是在程序执行过程中,其值可在指定范围内发生改变的值
   2. 作用:
      1. 变量就相当与超市的储藏柜(小格)
      2. 可以存数据, 可以取数据
9. 变量的作用域(%%%%%%)
   1. 什么叫作用域, 就是变量能够使用的范围
   2. 变量定义在代码块中,那么该变量的使用范围在代码块对应的一对大括号内有效
   3. 变量定义在方法中,那么该变量的使用范围在该该方法的范围内有效
   4. 变量定义在类中方法外,那么该变量的使用范围在类中的任意位置都可以访问
10. 常见的数据类型%%%%%%)
    1. 基本数据类型
       1. byte 一个字节
       2. short 二个字节
       3. int 四个字节
       4. long 八个字节
       5. float 四个字节
       6. double 八个字节
       7. char 字符 2个字节, char类型存储一个汉字默认情况下两个字节
       8. boolean 一个字节
    2. 引用数据类型
       1. 类 class
       2. 接口 interface
       3. 数组 Object[]

day02:

1. 数据类型转换
   1. 强制转换
      1. 容量大的数据类型,转换成容量小的数据类型
      2. 格式: 转换后的数据类型 变量名 = (转换后的数据类型) 转换前的数值;
   2. 自动类型转换
      1. 容量小的数据类型会 自动提升为 容量大的数据类型
      2. byte\short\char类型,参与运算的时候, 会自动提升为int类型
      3. 容量从小到大排序: byte < char\short < int < long < float < double
2. 运算符
   1. 算术运算符:
      1. % 取余(取模)
         1. 运算的结果,取决于左边数值的符号
         2. ++, --
            1. ++前置 ++a , 操作数a 先自增1, 然后再使用增1后的结果参与运算
            2. ++后置 a++, 操作数a, 先参与运算, 然后操作数a再自增1
   2. 赋值运算符:
      1. =与==的区别?
         1. = 代表 赋值
         2. == 代表 比较
      2. +=, -=, \*=, /= 等等..
         1. 作用是将左右两边的数值运算后,把结果赋值给左边
         2. 隐含了一个强制转换的操作
   3. 关系运算符
   4. 逻辑运算符
      1. & : 与运算符: 有假则假
      2. | : 或运算符: 有真则真
      3. !: 非运算符: 假变真, 真变假
      4. ^: 异或运算符: 相同为假,不同为真
      5. && || : 左边条件成立, 右边判断不执行
   5. 三元运算符:
      1. 注意: 用来给变量赋值, 但是不能打印数据
      2. ( 3 > 2) ? SYSO(2) : SYSO(3); 错误的
      3. 肯定会有一个返回结果数据
3. Scanner类
   1. 作用: 获取键盘输入的数据
   2. 格式:
      1. new Scanner(System.in).nextInt() 获取一个int数据
      2. new Scanner(System.in).nextLine() 获取一行字符串
      3. new Scanner(System.in).next() 可以获取键盘输入的数据包含字符串, 获取到空格等符号的时候结束获取
4. String类:
   1. 代表的是一个字符串对象,”abc”是一个字符串对象,字符串是一个常量,值不能改变, 如果要给字符串变量重新赋值, 那么会在内存中开辟一个新的空间
   2. 方法:
      1. equals()方法: 比较两个字符串对象内容是否相同
      2. isEmpty()方法: 判断该字符串对象内容是否是空字符串 , “”
      3. length()方法: 获取字符串对象中字符的个数
      4. substring()方法: 截取字符串内容,返回一个新的字符串
      5. getBytes()\ toCharArray() : 把字符串转换成数组

day03

1. Random 随机数类
   1. nextInt( int n) 方法: 返回0-n指定范围内的随机整数 包含0,不包含n
2. Math 数学类
   1. max() 方法 最大值
   2. min() 方法 最小值
   3. floor() 方法 返回比参数小的最大整数 4.3 🡪 4
   4. ceil() 方法 返回比参数大的最小整数 4.3 🡪 5
   5. round() 方法 四舍五入方法
   6. abs()方法 绝对值 不管是负数还是正数, 结果都是正数 -5 🡪 5
3. if选择结构

格式1:

if (条件) {

条件成立,执行if大括号内的代码

}

格式2:

if (条件) {

条件成立,执行if大括号内的代码

} else {

条件不成立,执行else大括号内的代码

}

格式3:

if的嵌套

if() {

if(){

} else {

}

} else {

if(){

} else {

}

}

1. switch选择结构
   1. 格式: 省略
   2. 注意事项:
      1. switch(参数) 可以接受的数据类型为: byte\short\int\char\String\枚举\
      2. case穿透
         1. 当case语句后面没有break的时候,会执行后面的case语句
      3. 什么时候算结束了switch语句
         1. 遇到break;
         2. 遇到了switch语句的右大括号
2. for 循环
   1. 格式: 省略
   2. 执行流程:
      1. for ( 初始化表达式 ; 循环条件表达式 ; 步进表达式 ) { code… }

先初始化表达式,再循环条件表达式, 然后code… , 最后步进表达式, 再再循环条件表达式, 然后code… , 当遇到循环条件表达式不成立时,跳出循环

1. while 循环
   1. 格式:省略
   2. while与for的区别

当不确定循环次数时,使用while

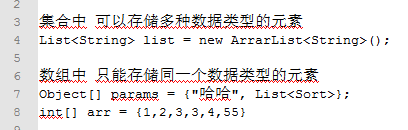
确定循环次数时,使用 for

day04:

1. 为什么要学习方法,使用方法后有什么好处?
   1. 提高了代码的复用性, 重复的代码写一次即可
   2. 使程序的结构可读性增强
2. 方法重载是什么?
   1. 在同一个类中,多个方法的名称相同,参数列表不同,这就是方法重载
   2. 参数列表不同
      1. 个数不同
      2. 参数类型不同
      3. 参数类型顺序不一样
   3. 方法重载的出现,方便了相同功能方法的命名
3. 数组的特点?
   1. 数组的理解: 它相当于包含了多个变量的容器, 相当于超市中大的存储柜(有很多个小格,小格的总个数是固定的)\
   2. 数组中是可以存储多个数据的容器,它的容量固定的
   3. 可以向数组内部,存储与取出数据

day05

1. 集合与数组的区别是什么?
   1. 集合长度可变, 存储的元素都是引用数据类型
   2. 数组长度固定, 存储的元素可以是引用数据类型和基本数据类型



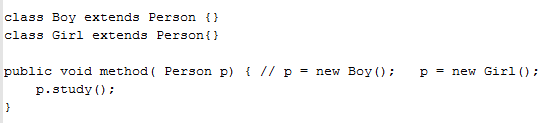
1. ArrayList集合的增删改查方法
   1. add()
   2. remove()
   3. set()
   4. get()

day07-09 面向对象

1. 面向对象的三大特性

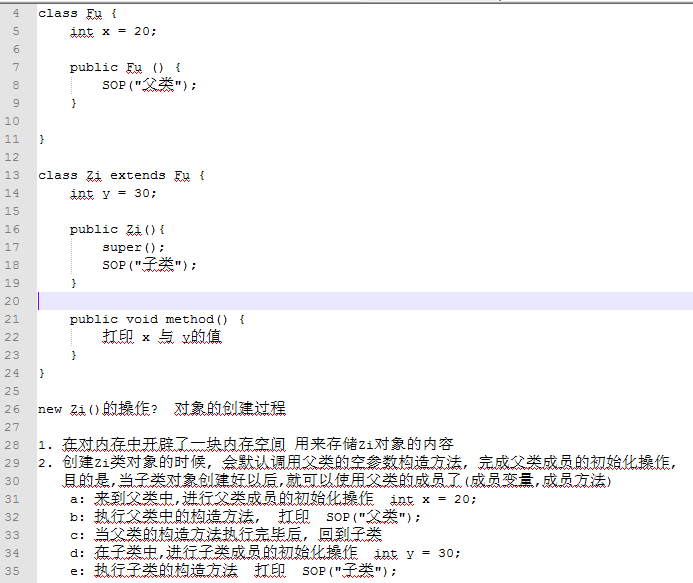
封装\继承\多态

1. 类与对象的概念进行解释
   1. 类:用来描述显示事物,以抽象的形式体现 ,例如 大众汽车图纸
   2. 对象: 事物的具体化, 例如 大众汽车辉腾
2. 抽象类\接口是什么?
3. 继承的作用是什么?
   1. 子类可以使用父类非私有的成员(成员变量与成员方法)
   2. 减少代码量
4. 多态的作用是什么?
   1. 提高了程序的扩展性



* 1. 类型转换
     1. 向上转换
        1. Person p = new Boy();
     2. 向下转换
        1. Boy b = (Boy) p;

1. final是什么? 有什么作用?
   1. final是最终的意思, 是一个关键字
   2. 使用final修饰的变量, 其值 不能改变
   3. 使用final修饰的方法, 该方法子类不能重写, 可以继承过来使用
   4. 使用final修饰的类, 该类不能被继承
2. static \ extends \ implememts是什么? 有什么作用?
3. 请列出抽象类与接口的不同指出
   1. 接口(只能有抽象方法与静态常量, 所有的方法与常量都是public 修饰)
   2. 抽象类(可以有任意的方法,可以有常量与变量, 所有的方法与常量\变量可以使用任意修饰符)
4. this\ super 是什么? 有什么作用?
   1. this 代表当前类对象,该对象可以使用当前类对象中的成员,也可以使用继承过来的成员 this.parentBox.getSelectedItem()
   2. this 来区别成员变量与局部变量
   3. super 代表当前类的父类对象, 例如 super(参数), 代表调用父类的有参数构造方法
5. Person类使用继承后,创建Person对象对应的实例化过程



1. 四种权限修饰符的访问范围

当前类中 同包下 不同包的子类中 不同包的其他类中

public Y Y Y Y

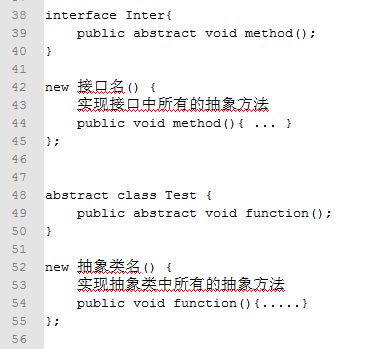
protected Y Y Y

默认的 Y Y

private Y

通常情况下, 变量 private , 方法 public

1. 内部类
   1. 格式: 省略
   2. 特点: 内部类可以访问外部类的成员(包含私有的成员)
   3. 匿名内部类: 就是没有名称的内部类



1. 方法重写 与 方法重载的区别?

override 与 overload 的区别?

方法重写: 在子父类情况中, 子类重新实现父类的同名方法

方法重载: 在同一个类中,多个方法的名称相同, 参数列表不同

day12-13

String:

StringBuilder\ StringBuffer:

append(): 在原有对象内容基础上,在末尾追加新数据

toString(): 把字符串缓冲区对象内容 转换成字符串

Date : 日期

getTime() 得到日期对应的毫秒值s

DateFormat: SimpleDateFormat: 日期格式化类

format: 按照指定的格式,把日期 转换成字符串

parse: 按照指定的格式,把字符串 转换成 日期

Calendar: 日历类

get()方法,可以获取到指定字段的 日期值

add()方法, 修改日历时间的方法  
 set()方法, 设置日历时间的方法

`正则表达式

String类match()方法, 验证匹配数据结果方法

基本数据类型包装类

byte Byte

short Short

long Long

float Float

double Double

boolean Boolean

char Character 字符类型

int Integer整数类型

自动装箱: Integer in = 3; 基本数据类型 🡪 包装类对象

自动拆箱: int num = in; 包装类对象 🡪 基本数据类型

1.String 与 StringBuffer的区别?

String:字符串:

当为字符串变量,重新赋值的时候, 会在内存中开辟一个新的内容,存储新的数据,把新的内存空间地址 赋值为字符串变量

StringBuffer 字符串缓冲区:

当为字符串缓冲区变量,重新赋值的时候,会在原有的内存空间内,进行新数据的存储,返回给字符串缓冲区变量的地址 是原有的地址

day14-15

Collection: 单列集合

|- List

|- ArrayList

|- LinkedList

|- Set

|-HashSet

|- LinkedHashSet

Map: 双列集合

|- HashMap

1.ArrayList 与 LinikedList的区别?

ArrayList 底层数据存储结构采用 数组结构方式, 特点 查询快,增删慢

LinkedList 底层数据存储结构采用 链表结构方式, 特点, 增删快, 查询慢

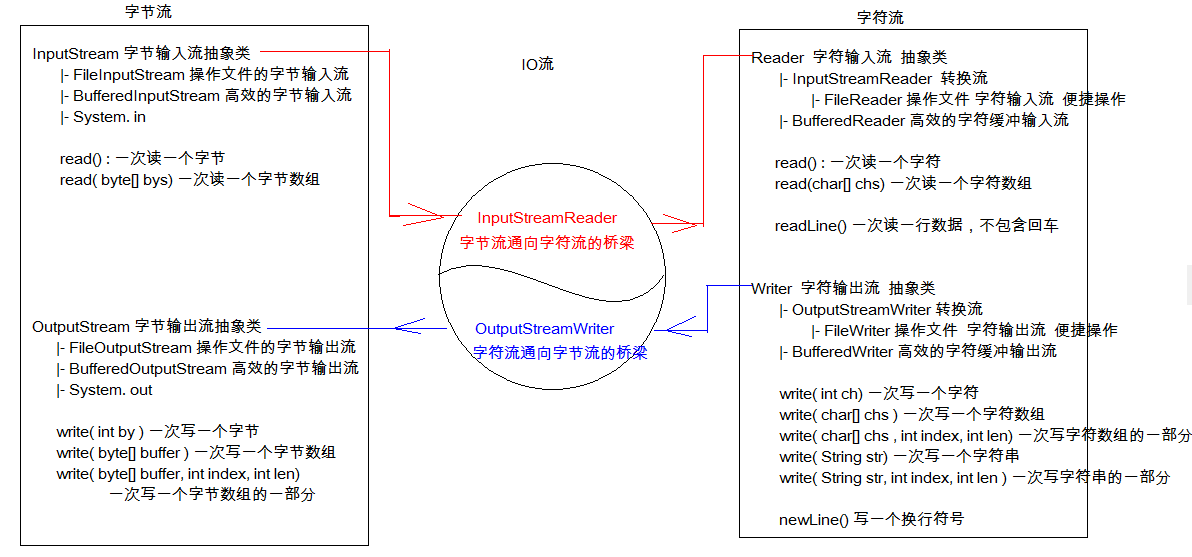
1. ArrayList集合中的元素是否是唯一的? 是否可以存储重复的数据?
   1. 不是唯一的
   2. 可以存储重复的元素
2. HashSet集合中的元素是否是唯一的? 如何保证元素的唯一?
   1. 元素唯一
   2. 重写元素数据的equals方法与hashcode方法,
3. HashMap集合中的元素是否是唯一的? 谁唯一? 如何保证唯一的?
   1. 键key是唯一的,值value是可以重复的
   2. 如何保证键唯一? 重写键 元素数据的equals方法与hashcode方法
4. 使用泛型的好处?
   1. 可以自动强制类型转换
   2. 避免了类型转换异常 ClassCastException
   3. 将运行期的错误 提前到了 编译期进行处理
   4. 明确了 集合中存储的元素类型
5. 可变参数 是如何理解的? 它相当于是什么?
   1. 类型相同,个数不确定的一个参数 int…
   2. 相当于是一个指定数据类型的数组 int[]
6. 数据结构
   1. 栈与队列
      1. 栈的特点
         1. 先进后出
      2. 队列的特点
         1. 先进先出
7. 异常
   1. try: 可能出现异常代码的部分
   2. catch: 捕捉异常,进行异常的处理
   3. finally: 不管是否有没有异常产生,都必须要执行的操作
   4. throws 抛出异常,交给调用者处理, 必须使用在方法的声明上,告诉调用者该方法可能会产生一个或多个异常
      1. publc void method() throws Exception
   5. throw: 抛出异常, 需要进行处理, 必须使用在方法内部, 一定会抛出一个异常对象,
8. return 与 throw的区别?
   1. return结束方法,进行返回, 可以带回一个数据(基本数据类型\引用数据类型)
   2. throw 结束方法,进行返回, 带回一个异常对象
9. 异常的种类有哪两种?
   1. 编辑期异常 Exception
      1. 异常必须处理,否则无法正常编辑通过运行
   2. 运行期异常 RuntimeException
      1. 运行期异常可以不理睬,程序可以正常运行

day16-17

1.文件的复制

普通字节流的方式

高效的缓冲字节流的方法



2.常见编码表

ASCII :

GBK: 简体中文

GB2312简体中文, 对GBK的升级

GB18030简体中文, 对GB2312的升级

UniCode编码: 统一码

UTF-8: 统一码

day18

1.线程对象创建的方式

a. 继承Thread类

b. 实现Runnable接口的类, 把类对象 作为 Thread对象的参数使用

c. 实现Callable接口的类

2. run()方法与start()方法:

start: 启动线程

run: 指定线程要执行的操作

new Thread(). start() 启动一个新线程

new Thread().run() 没有启动新线程

同步机制Synchronized :

作用: 保证多个线程对象操作共享数据时, 保证共享数据的正确性

同步机制的方式:

同步代码块

同步方法

静态同步方法

线程的生命周期图解(文字描述)

day19

使用TCP协议,完成客户端与服务器段的数据传送

Socket与ServerSocket完成一个字符串的发送与响应

day21-22

给一个数据库表

使用JDBC的方式,完成数据表内容的增删改查

* + - 1. 加载驱动, 注册驱动
      2. 获取连接对象
      3. 获取发送SQL语句的对象 Statement PreparedStatement
      4. 发送SQL语句,得到结果集ResultSet
      5. 遍历结果集 Result, 通过next()方法
      6. 获取当前表记录的数据,
         1. getString(“sname”) 获取分类名称
         2. getDouble(“money”) 获取金额
         3. getObject(..) 获取指定数据类型数据…
      7. 释放资源

使用三种以上的DBUtils的ResultsetHandler

压题:

面向对象一道:

封装\继承\抽象类\接口

集合:

使用Collection(HashSet)集合存储自定义对象并遍历

使用Map 集合存储自定义对象并遍历(两种遍历方式)

IO:

复制文件(注意异常的处理)

普通字节流的方式

高效的缓冲字节流的方法

文件夹遍历(递归) (注意异常的处理)

网络:

使用Socket与ServerSocket完成一个字符串的发送与响应, 客户端获取反馈信息

数据库:

给一个数据库表

使用JDBC的方式,完成数据表内容的增删改查

* + - 1. 加载驱动, 注册驱动
      2. 获取连接对象
      3. 获取发送SQL语句的对象 Statement PreparedStatement
      4. 发送SQL语句,得到结果集ResultSet
      5. 遍历结果集 Result, 通过next()方法
      6. 获取当前表记录的数据,
         1. getString(“sname”) 获取分类名称
         2. getDouble(“money”) 获取金额
         3. getObject(..) 获取指定数据类型数据…
      7. 释放资源