# Review

## Java简介

1. Java是面向对象的、跨平台的程序设计语言；
2. Java程序是运行在Java虚拟机之上的；
3. 要下载安装JDK，才可以开发和运行Java程序；

## 语言基础

1. 变量的声明和赋值 类型 变量名 = 值;
2. 变量命名规范 字母开头，第二个单词首字母大写
3. 定义变量的两种方式 类型 变量名,变量2； 变量名= 值；

Java数据类型

引用类型

基本类型

数值型

布尔型(boolean)

整型(byte,short,int,long)

浮点型(float,double)

类(class)

接口类型(interface)

数组类型

枚举类型(Enum)

字符型(char)

注解类型(Annotation)

1. 赋值运算符：= += -=
2. 算数运算符：+,-,\*,/,%
3. 关系运算符：>,<,==,!=,>=,<=
4. 逻辑运算符：&&，||，!
5. 自动类型转换：目标类型大于源类型
6. 强制类型转换：（类型名）值
7. if、多重if

if(boolean条件){…….}

else if(){}

else{}

1. switch

switch(数字/字符串){

case 1:

……

break;

case 2:

default;

}

1. %与/区别 取余/除法
2. i++、++i、i--、--i

## 循环

1. for循环

标记：for(初始化;进入循环的条件;循环结束执行){

循环体

for(初始化;进入循环的条件;循环结束执行){

break 标记；

}

}

1. while循环

while(进入循环的条件){

循环体

}

1. do while循环

do{

循环体

}while(条件);

1. continue和break区别

continue：结束当次循环，进入下一次循环

break: 结束循环

## 数组

1. 数组的声明

String[] a = new String[9];

String[] b = {"a","b","c"};

String[] c = **new** String[]{"a","b","c"};

1. 数组的赋值

a[0] = "b";

1. 数组排序

Arrays.*sort*(c);

## 类和对象

1. 面向对象三大原则

封装、继承、多态

1. 对象的特征—属性
2. 对象的行为—方法
3. 自定义类

public class Student{

private String name;

private int age;

public void paly(){

}

}

1. 创建和使用对象

Student stu = new Student();

stu.paly();

1. 类和方法的命名规范

类：首字母大写 驼峰式命名

方法：首字母小写 驼峰式命名 动词

1. 方法返回值

return

哪里调用 返回到哪里 调用的地方 方法可以看成一个 值

1. 方法调用

stu.play();

1. 变量作用域
2. 面向对象的思想
3. 包，包的命名规范

com.javasm.部门名.项目名.模块功能.

com.baidu.

org.oracle

1. 注释：单行注释，块注释，文档注释

//

/\*

\*/

/\*\*

\*

\*

\*/

## 封装

1. private
2. setter/getter方法
3. 构造方法
4. 方法重载

参数类型或参数数量不同 方法名相同

1. this关键字

当前对象

1. static关键字

~~单例模式~~

## 继承

1. extends关键字 is – a
2. 访问修饰符

public,protected,缺省,private

1. 继承中的构造方法

构造方法第一行代码，必须调用父类的构造方法

1. super关键字

super(父类的构造方法)

super.name 指明调用父类的属性

1. 重写

参数类型一致

## 多态

1. 多态是具有表现多种形态的能力的特征
2. abstract抽象方法，抽象类

抽象方法：没有方法体

抽象类：不能实例化，子类必须重写父类的抽象方法

## 接口

1. 创建接口**interface**

接口里面的方法都是抽象方法

修饰符都是public

接口属性都是常量

1. 实现接口implements

实现多个接口，不同的接口之间用,分割

1. final关键字

final修饰的类：不能被继承

final修饰的变量：不能重新赋值 常量

final修饰的方法：不能被重写

final修饰的参数 ： 不能被重新赋值

1. 常量

|  |  |  |  |
| --- | --- | --- | --- |
| OO基本特征 | 定义 | 具体实现方式 | 优势 |
| 封装 | 隐藏实现细节，对外提供公共的访问接口 | 属性私有化、添加公有的setter、getter方法 | 增强代码的可维护性,数据的安全性 |
| 继承 | 从一个已有的类派生出新的类，子类具有父类的一般特性，以及自身特殊的特性 | 继承需要符合的关系：is-a，通过使用extends关键字实现继承，使用super调用父类的成员 | 1、实现抽象（抽出像的部分）  2、增强代码的可复用性 |
| 多态 | 同一个实现接口，使用不同的实例而执行不同操作 | 通过Java接口/继承来定义统一的实现接口；通过方法重写为不同的实现类/子类定义不同的操作 | 增强代码的可扩展性、可维护性 |

## 异常

1. 运用try块、catch块和finally块处理异常
2. 运用多重catch块处理异常
3. 运用关键字throw和throws处理异常
4. Exception是所有异常类的父类
5. final和finally区别、throw和throws区别

## 语言包

1. 包装类(常用)

valueOf和parseXxx

1. String类(重点) -->手写一个String -->char[]

* 获取String长度， .length()
* 截取字符串， substring()
* 字符串比较 equals ==
* 字符串搜索 indexOf() lastIndexOf()
* 字符串替换 replace() replaceAll()
* 去掉字符串首尾空格 trim()
* 数据格式化getBytes(),toCharArray()
* split() 分割字符串 以符号分割为String数组

1. StringBuffer和StringBuilder
   * 连接：append
   * 倒置：reverse
2. Math类
3. Object类
   * toString();
   * ~~clone();~~
4. Class类(了解)

Class.forName(“xxx.xxx.xxx”)

## 工具包

1. Date类(常用)

* new Date()
* getTime() 时间的毫秒数
* System.currentTimeMillis()

1. Calendar类

* Calendar getInstance()
* int get(int fields)
* void set(int fields, int value)
* void add(int fields, int amount)
* Date getTime()

1. DateFormat类(常用)

* SimpleDateFormat
* format
* Parse

1. Instant 类

* Now()
* Plus()
* Minus()

1. LocalDate/LocalTime/LocalDateTime类(常用)
2. DateTimeFormatter类(常用)
3. NumberFormat类

* DecimalFormat
* format

1. Random类(常用)

* Random rd = new Random();

rd.nextInt(30);0~30之间,包括0不包括30

* TreadLocalRandom

1. ArrayList类

* size() 获取List大小
* isEmpty() 判断List是否为空
* add() 添加元素
* get() 获取 get(4)
* set() set(2,”Hello”)
* clear() 清空
* remove() 移除某一个数据
* removeAll() 批量移除
* 数组转换为List：Arrays.asList(1,2,3);
* toArray(T[] t)

1. HashMap类

* put(key,value) 存值
* get(key) 获取值
* keySet() key集合(忽略)
* EntrySet()
* forEach()
* values() 值集合(忽略)

1. Set集合
2. 循环集合

* 普通for循环
* 增强型for循环/foreach
* 迭代器

1. ~~集合排序~~

* 基本数据类型
* Collections.sort(list);
* 自定义类型
  + 创建类，实现Comparator接口
  + 自定义类，实现Comparable接口

## I/O流

Read/write/readLine()

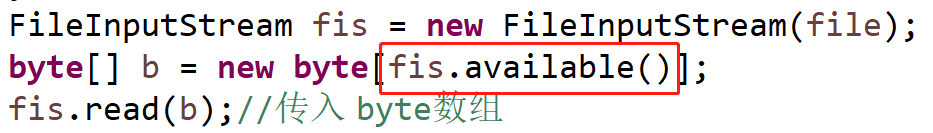
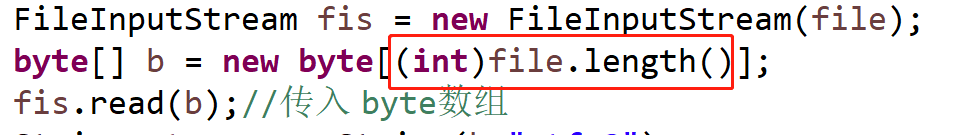
1. File类

exists()、createNewFile()、mkdirs()、length()

递归

File[] listFiles()

String[] list()

1. FileInputStream、FileOutputStream
2. 

可以修改成.available()方法获取流的大小

1. 复制文件
2. 序列化/反序列化
3. Properties类加载配置文件，并读取文件内容

## 多线程

1. 在JAVA中有3种实现线程的方式：继承Thread、实现Runnable，实现Callable+Futute
2. 线程的生命周期
3. ~~理解线程池~~
4. ~~synchronized关键字~~

~~同步方法~~

~~同步代码块 synchronized(){}~~

1. ~~Lock ReetrantLock lock()/unlock()~~
2. ~~生产者/消费者模式（线程通信）~~

~~Object: Wait()/notify()~~

~~Thread: sleep(time) vs wait(time)/wait()~~

~~Thread: Join()~~

## ~~网络~~

1. InetAddress ip
2. URL （调用接口/服务）--->HttpClient--->String 转换 JSON
3. ~~Socket 编写服务端(ServerSocket server = new ServerSocket(端口号);)~~
4. ~~Socket 编写客户端(Socket socket = new Socket(服务器IP地址 , 服务器端口号);)~~

## MYSQL

1. 安装数据库
2. SQL语句的分类：

DDL: create alter drop

DML: insert delete update

DQL: select

DCL: grant commit rollback

1. 基本增、删、改、查语句
2. 约束

非空约束 not null

唯一性约束 unique

默认约束 default

主键约束 primary key (一张表有且 只有一个主键列)

联合主键：（多个列充当一个主键列）

整型+自增(auto\_increment)

Varchar+uuid(当前此刻的时间充当id /UUID)

Tid sid

1 2

1 3

Create table a（

Tid

Sid

primary key（tid,sid）

）

外键约束：foreign key

外键列：从表 (参考主表的主键列数据)

T\_user 从表

Rid 外键列

T\_role 主表

Id 主键列

1. 定义字段别名 user\_name as name
2. Where条件 where ……
3. In in(1,2,3)
4. Like like(%a%) \_
5. 逻辑运算符(and;or;not;)
6. 排序order by 字段 asc/desc,;
7. 分组group by (Having)；
8. 常用函数(count(\*);)
9. 分页LIMIT ?/?,?
10. 多表查询 交叉查询 select \* from A,B where A.ID = B.ID
11. 外查询 left join on right join on inner join on
12. 子查询 select \* from (select \* from A) as tmp where tmp.id = 1
13. 自查询
14. distinct
15. 集合查询 union VS union all
16. from\_unixtime(unix\_timestamp,”%Y-%m-%d %H:%i:%S”)时间格式转换
17. Now()/last\_insert\_id() /uuid()
18. 设计数据库规则：数据库表每一列不可再分割

## JDBC

Driver 接口 java.sql.\*

JDBC连接数据库(背写)

Connection

PreparedStatement和Statement区别

ResultSet对象 select

DBUtils-->了解

注：~~划横斜部分~~表示可以忽略不记