# 考试：满分100分 + 20分附加题

基础题：5题（2分/题，共10分）

简单编程题：3题（5分/题，15分）

简单简答题：7题（5分/题，共35分）

较难编程题：2题（8分/题，共16分）

较难简答题：3题（8分/题，共24分）

附加题：2题（10分/题，共20分）

# 基础题（2分/题）

1. 两个对象a和b，请问a==b和a.equals(b)有什么区别？

|  |
| --- |
| 返回值：两者均返回boolean，true为相等，false为不等  如a和b为基础数据类型，则只能使用==来做比较。  如a和b为引用数据类型，则比较的都是地址值，但如果该类重写了equals方法，则equals方法比较的是两个对象的属性值。  通过查看源码发现，Java的核心类库中，String类型对equals方法进行了重写，比较的是对象的属性值。 |

1. 以下代码的执行结果是什么？

|  |
| --- |
| Public static void main(String[] args){  Int i=0;  change(i);  i= i++;  System.out.println(“i = ”+ i);  }  Public static void change(int i){  i++;  }  请写出答案：i=1 |

1. float f=3.4;是否正确，表达式15/2\*2的值是多少？

|  |
| --- |
| 错误  14 |

1. String str = new String(“hello”);创建了哪些对象？

|  |
| --- |
| 至少在内存中开辟出两块内存空间：   1. new String(“hello”)--堆中的对象 2. “hello”的字符数组--方法区有则使用，没有则创建 |

1. 以下代码的执行结果;

|  |
| --- |
| public static void main(String[] args) {  int a = 8, b = 3;  System.out.println(a>>>b);  System.out.println(a>>>b | 2);  }  请写出答案：  2  1 |

# 基础编程题（5分/题）

1. 请写一个冒泡排序，实现{5,7,3,9,2}从小到大排序

|  |
| --- |
| public class Test21 {  public static void main(String[] args) {  int[] nums = new int[]{5,7,3,9,2};  sort(nums);  for (int num : nums) {  System.out.print(num);  }  }  public static void sort(int[] ints){  if (ints.length<2){return;}  for (int i = 0; i < ints.length-1; i++) {  for (int j = 0; j < ints.length-i-1; j++) {  if (ints[j]>ints[j+1]){  int temp = ints[j];  ints[j] = ints[j+1];  ints[j+1] = temp;  }  }  }  }  } |
| 预期结果：  23579 |

1. 通项公式如下：f(n)=n + (n-1) + (n-2) + .... + 1，其中n是大于等于5并且小于10000的整数，例如：f(5) = 5 + 4 + 3 + 2 + 1，f(10) = 10 + 9 + 8 + 7+ 6 + 5 + 4 + 3 + 2 + 1，请用非递归的方式完成方法long f( int n)的方法体。

|  |
| --- |
| public static long f(int n){  return (1+n)\*n/2;  } |

1. 编写一个懒汉式单例设计模式

|  |
| --- |
| //线程安全的懒汉式  class Singleton{  private Singleton() { };  private static Singleton instance;  public static Singleton getInstance(){  if (instance==null){  synchronized (Singleton.class){  if (instance==null){  instance = new Singleton();  }  }  }  return instance;  }  } |

# 基础简答题（5分/题）

1. 访问修饰符的作用范围由大到小，及各自的范围是什么？可以修饰什么？

|  |
| --- |
| 访问修饰符的作用范围由大到小依次是：   1. public    1. 范围：类、包、子类、其他包    2. 可以修饰：类、接口、变量、方法 2. Protect    1. 范围：类、包、子类    2. 可以修饰：变量、方法 3. default（默认，什么也不写）    1. 范围：类、包    2. 可以修饰：类、接口、变量、方法 4. Private    1. 范围：类    2. 可以修饰：变量、方法   访问权限范围越小，安全性越高 |

1. 请解释Overload与Override的区别？

|  |
| --- |
| Overload方法的重载  作用：同一方法名下，可以有不同的功能  前提：使用在同一类中  实现：   1. 方法名相同 2. 参数列表不同（参数个数、参数类型）   注：与返回值类型无关  ————————————————————  Override方法的重写  作用：覆盖父类的方法  前提：子类继承父类后  实现：修饰符、返回值、方法名必须完全相同（一般情况下）   1. 创建子类，调用该方法，实际上运行的是子类的方法。如同将父类中的方法覆盖了一样 2. 修饰符可以不同的情况：子类重写的访问修饰符不能小于父类被重写方法的访问修饰符，否则编译报错 3. 返回值可以不同的情况：子类方法的返回值类型必须是父类被重写方法的返回值类型的子类 |

1. 请解释String、StringBuilder、StringBuffer的区别？

|  |
| --- |
| 按照运行速度由慢到快，依次为：   1. String：字符串常量，不可变的字符序列，一旦创建不可更改。适用于少量字符串操作。 2. StringBuffer : 字符串变量，可更改。是线程安全的，因此效率低。适用于多线程下大量操作。 3. StringBuilder : 字符串变量，可更改。是线程不安全的，因此效率高。适用于单线程下大量操作。 |

1. ArrayList和LinkedList的底层实现（存储结构、扩容机制）

|  |
| --- |
| ArrayList  存储结构：数组结构  适用场景：当查询操作多时选择  扩容机制：为原数组长度的1.5倍，向下取整  LinkedList  存储结构：双向链表结构  适用场景：当增删操作多时选择  扩容机制：链表实现，没有初始化大小，也就没有扩容机制  两者均为线程不同步，效率高于Vector |

1. Java中的IO流的四大基类是什么（2分）,请列出常用的IO流类型（至少5个）（3分）？

|  |
| --- |
| 四大基类：   1. 字节输入流：InputStream 2. 字节输出流：OutputStream 3. 字符输入流：Reader 4. 字符输出流：Writer   常用的IO流类型   1. 节点流：FileInputStream ,FileOutputStream ,FileReader ,FileWriter 2. 缓冲流：BufferedInputStream , BufferedOutputStream , BufferedReader , BufferedWriter 3. 打印流：PrintStream, PrintWriter 4. 控制台IO：System.in 5. 转换流：InputStreamReader ,OutputStreamWriter 6. 对象流：ObjectInputStream , ObjectOutputStram |

1. 请列举一些常见的异常或错误类型（至少5个）

|  |
| --- |
| 1. java.lang.NullPointerException 空指针异常，当试图在要求使用对象的地方使用了null时，会抛出该异常。如调用null对象的实例方法、属性等 2. java.lang.IOException IO流异常 3. java.lang.ArithmeticException 算术异常，如除数为0 4. java.lang.IndexOutOfBoundsException 索引越界异常，当访问某个序列的索引值小于0或大于等于序列大小时，会抛出该异常 5. java.lang.ClassCastException类型转换异常 当两个类不存在子父类关系时使用强制类型转换会抛出该异常 6. java.lang.ThreadDeath 线程结束 调用Thread类的stop方法是抛出该错误 7. java.lang.NoSuchMethodError 方法不存在错误 该类的定义中没有该方法的定义时抛出错误 |

1. 编写多线程程序有几种实现方式？

|  |
| --- |
| 1. 继承Thread类 2. 实现Runnable接口 3. 线程池 |

# 较难简答题（8分/题）

1. 运行时异常与编译时异常有何异同？请列举一些运行时异常和编译时异常的类型。

|  |
| --- |
| 相同点：都属于异常机制的一部分  不同点：   1. 编译时异常（受检异常）：必须处理（捕获或抛出）的异常，如果不处理，程序就不能编译通过 2. 运行时异常（不受检异常）：一般由程序逻辑错误引起。可以选择处理，也可以选择不处理。如不处理，编译会通过，但在运行时遇到问题会直接由JVM抛出并终止程序。   常见的运行时异常：   1. java.lang.NullPointerException 空指针异常，当试图在要求使用对象的地方使用了null时，会抛出该异常。如调用null对象的实例方法、属性等 2. java.lang.ArithmeticException 算术异常，如除数为0 3. java.lang.NullPointerException 空指针异常，当试图在要求使用对象的地方使用了null时，会抛出该异常。如调用null对象的实例方法、属性等   常见的编译时异常：   1. java.lang.IOException IO流异常 2. java.lang.FileNotFoundException 找不到该文件 3. java.lang.ClassNotFoundException 类无法找到 |

1. HashMap的底层实现及扩容机制？

|  |
| --- |
| 底层实现：  是由数组+链表实现的，其本质是一定长度的Entry类型的数组，数组中存放的是链表。  Entry中存放了key，value，hash值，指向下一个元素的引用  扩容机制：  负载因子（默认0.75，过大碰撞几率增加，过小资源浪费）  当占用空间达到当前空间的75%时，扩容为当前空间容量的两倍。 |

1. 如何实现序列化，有什么意义？

|  |
| --- |
| 通过对象流实现对象的序列化和反序列化。  序列化：ObjectOutputStream  反序列化：ObjectInputStream  实现步骤：   1. 创建FileOutputStream的实例，同时打开指定文件（一般为“xxx.dat”） 2. （可选）创建缓冲流，包装FileOutputStream，提高效率 3. 创建ObjectOutputStream的实例，包装缓冲流 4. 通过writeXxx方法，完成序列化（是什么类型就调用什么类型的方法，其中String类型为readUTF()） 5. 关闭流 6. 需要序列化对象的类实现java.io.Serializabel，其中被transient和static修饰的属性不能被序列化 7. （可选，建议提供）提供序列号 private static final long seiralVersionUID = 154676313L;   意义：将内存中的对象永久的以二进制形式保存在磁盘中 |

# 较难编程题（8分/题）

1. 第一个人10，第2个比第1个人大2岁，以此类推，请用递归方式计算出第8个人多大？

|  |
| --- |
| public class Test51 {  public static void main(String[] args) {  System.out.println(age(8));  }  public static int age(int num){  if (num==1){  return 10;  }else {  return age(num-1)+2;  }  }  } |
| 预期结果：  24 |

1. 有一个字符串，其中包含中文字符、英文字符和数字字符，请统计和打印出各个字符的字数。举例说明： String content = “中中国55kkfff”; 统计出：中：2, 国：1, 5：2, k：2, f：32

|  |
| --- |
| import java.util.HashMap;  import java.util.Map;  import java.util.Set;  public class Test52 {  public static void main(String[] args) {  String content = "中中国55kkfff";  HashMap<Character, Integer> hashMap = countNum(content);  Set<Map.Entry<Character, Integer>> entrySet = hashMap.entrySet();  for (Map.Entry<Character, Integer> entry : entrySet) {  System.out.println(entry.getKey() + ":" +entry.getValue());  }  }  public static HashMap<Character,Integer> countNum(String content){  char[] chars = content.toCharArray();  HashMap<Character,Integer> hashMap = new HashMap<>();  for (char c : chars) {  if (hashMap.get(c)==null){  hashMap.put(c,1);  }else {  hashMap.put(c, hashMap.get(c)+1);  }  }  return hashMap;  } |

# 附加题（10分/题）

1. 谈谈你对面向对象的理解

|  |
| --- |
| 面向对象是一种“万物皆对象”的编程思想，它的三大特性是封装、继承、多态。  比如把大象装进冰箱，  面向过程的做法是打开冰箱门，把大象装进去，把冰箱门关上，如有两个人要做，那就要写两次做法。  面向对象的做法是将冰箱作为对象，冰箱门的打开关闭添加内容作为方法，将其封装好，即屏蔽了差异，提高了代码的通用性。   1. 封装   将一类事物的属性和行为抽象成一个类，使其属性私有化，行为公开化。  提高①数据隐秘性②代码模块化③代码复用性   1. 继承   将一类事物共有的属性和行为抽象成一个父类，每一个子类是拥有父类的行为和属性，也有自己特有的行为和属性。既拓展了已存在的代码块，同时提高了代码的复用性   1. 多态   允许父类引用指向子类的对象 |

2.线程的创建方式有哪些，用哪种比较好？

|  |
| --- |
| 1. 继承Thread类  2. 实现Runnable接口  3. 线程池  Runnable和Thread相比，Runnable较好。原因：   1. 单继承多实现，继承了Thread无法再继承其他类 2. 多线程下可以共享一个target对象，更适合多线程处理同一份资源   使用线程池的优点：  频繁的创建关闭线程会大量消耗资源，使用线程池可以更好的管理线程。 |