

# 总结

# 题型

- 单项选择题、判断题（Java语法）
- 编程题  
看程序写结果、改写代码（递归、类的初始化、Lambda演算）
- 问答、简答题（结构化编程、面向对象编程（封装、继承、多态）、字节码基础）

# 判断题

- 判断题
  - 类的静态成员在类加载时就会被初始化。()

# 选择题

执行如下程序代码后，C的值是？

```
a=0;c=0;  
do{  
  --c;  
  a=a-1;  
}while(a>0);
```

- A、 0
- B、 1
- C、 -1
- D、 死循环

# 选择题

关于Java中的异常处理，下列说法正确的是？

- A. finally 块中的代码一定会执行，无论是否抛出了异常。
- B. 自定义异常类必须直接继承 Exception 类。
- C. try 块中只能有一个 catch 块来捕获异常。
- D. RuntimeException 是受检异常，必须在方法签名中声明或在方法中捕获。

# 编程题

写出下述Java程序输出结果（2分），并解释abc函数的功能（3分）

```
public class Test {
    public static int abc(int[] nums) {
        if (nums == null || nums.length < 2) {
            throw new IllegalArgumentException("Array must have at least two elements.");
        }

        int a = Integer.MIN_VALUE;
        int b = Integer.MIN_VALUE;

        for (int num : nums) {
            if (num > a) {
                b = a;
                a = num;
            } else if (num > b && num < a) {
                b = num;
            }
        }

        return b;
    }

    public static void main(String[] args) {
        int[] x = new int[] {1, 2, 3, 4, 5};
        System.out.println(Test.abc(x));
    }
}
```

# 编程题

编写一个Java程序，要求实现以下功能：

- 定义一个抽象类Shape，包含抽象方法area()用于计算形状的面积。
- 创建两个子类Circle和Rectangle，分别实现Shape类。

# 简答题

解释==和equals()的区别，并写出以下代码输出结果。

```
String str1 = new String("Hello");  
String str2 = "Hello";
```

```
System.out.println(str1 == str2);  
System.out.println(str1.equals(str2));
```



# Lamdar演算

- 求  $(\lambda x. (\lambda y. (xy)) (\lambda z. zz)) (\lambda w. w)$
- 求  $(\lambda x. x y) z$
- 对每个表达式，进行逐步的 $\beta$ -归约（Beta Reduction），并给出每一步的推导过程。

# 编程能力

- 看程序写结果，请说明这段代码实现什么功能。

- public class RS {

- public static String func(String str) {

- if (str == null || str.length() <= 1) {

- return str;

- }

- char[] a = str.toCharArray();

- char[] ra = new char[a.length];

- for (int i = 0; i < a.length; i++) {

- ra[a.length - 1 - i] = a[i];

- }

- return new String(ra);

- }

- public static void main(String[] args){

- System.out.print(func("Hello World"));

- }

- }

# 结构化编程

- 数据流图、结构图、流程图
- 数据流图向结构图的转化
- lambda演算主要考推演过程

# 面向对象题目

- 设计学生类，画出类图，并写出伪代码
- 画出选课系统的用例图，并解释用例图四要素
- 软件工程建模思想，需求、用例图、类图、标准UML图等。

# 面向对象题目

- 类的初始化顺序（静态变量、变量、构造器执行顺序）
- 类的继承
- 类的封装和协作（向系统添加新的功能如何设计？）
- 类之间的关系，图书馆和图书，图书和书页
  - 依赖、普通关联、聚合、组合、继承、实现
- 其它

# 字节码、JVM题

- Invokevirtual、invokespecial、invokeinterface的区别？（基础字节码指令的含义，iconst、istore）
- 分派原则（动静态绑定）
- JVM内存区域划分？堆区、栈区、方法区
- Overloading和Overriding

# 知识点

- 顺序、选择、循环
- String
- 数据建模、算法建模
- 递归
- Lambda演算
- 软件开发生命周期
- 结构化编程和面向对象编程
- Overriding vs Overloading
- 封装
- 职责、协作
- 类之间的关系
- 继承
- 多态
- 继承vs组合
- 类的初始化
- 动静态绑定
- 接口
- 针对接口编程
- 可修改性
- 异常
- 字节码、JVM

# 重点（结构化、面向对象）

- 结构化编程
  - 自顶向下逐步求精
  - 树状结构
  - 数据流图
  - 结构图
- 单个类封装
  - 数据和行为的在一起
  - 单一职责
- 多个类协作
  - 委托
  - 职责的分配
- 可修改性
  - 实现的修改（封装）
  - 扩展（继承，多态）
  - 灵活性（组合+接口）