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、python语法 、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）

判断题

- 判断题
  - main()方法应该处理所有未被处理的异常。 （）

Lamdar演算

- 给出完整的lamdar演算的具体推导步骤
- 1.
- ZERO = \f.\x.x
- SUCC = \n.\f.\x.f (n f x)
- 求　　SUCC (SUCC ZERO)

看程序写结果

- 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"));
- }
- }

面向对象题目

- 1.
- class Piece{
- }
- public class Board{//棋盘
- public Piece[] piecelist;
- public static void main(String[] args){
- Board board = new Board();
- board.piecelist.get(0);
- }
- }

- 1) 上面的代码在运行的时候会报错。试分析Piece和Board两个类的关系，找出错误的原因，画出两者的类图并进行改正。（5分）
- 2) 分析Piece和Board类的职责（数据职责和行为职责），请问吃子的职责应该是谁的职责，在上面的代码中加上相应的方法的定义和实现的伪代码。（5分）

字节码、JVM题

- Invokevirtual、invokespecial、invokeinterface的区别？（重要指令的含义）
- JVM内存区域划分？

问答、简答

- lambda演算主要考推演过程
- 用例图、数据流图、结构图、类图会考，标准UML

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）