

总结

题型

- 单项选择题、判断题 （Java语法）
- 简答题 （软件工程各种概念）
- 看程序写结果，程序完形填空 （结构化编程、递归、Lambda演算、类的初始化）
- 问答题 （面向对象编程）
- 编程题 （最后2章的内容GUI+网络）
- 附加题 （字节码）

判断题&软工问答题

- 判断题

- `main()`方法必须处理所有未被处理的异常。 ()

- 问答题

- 为什么“个人英雄主义”式编程无法解决大规模软件开发问题？请给出软件工程定义。（3分）

Lamdar演算

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

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 会报如下错误
Exception in thread "main"
java.lang.NullPointerException at...。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接口的定义和实现的伪代码。伪代码要包含对方的方法的调用。（5分）

编程题

- 生产者客户端和消费者客户端，他们都向服务器端的联名账户请求。生产者客户端负责向账户存钱，消费者客户端负责向账户取钱。客户端登陆之后进入主界面有一个JTextArea显示每次存（取）款的信息（存（取）了多少钱，余额多少），一个JTextField用来输入存（取）款的数量，一个JButton用来发送存（取）款请求。服务器接受请求更改联名账户的余额，并向客户端发送存（取）款消息进行同步。重点写出GUI对象的构建和初始化、Button的响应方法、Socket通讯、线程、异常的运行的相关片段代码。注意多线程同步问题。

字节码题

- 写出下列程序的输出结果。（5分）结合后面的字节码，说明Java中遵从什么样的分派原则，编译器在编译中如何编译方法的调用，JVM是如何决定运行哪个方法的。（5分）

考点

- 第一章
- 第二章
- 不考
- 第三章
- 第四章
- 第五章

- 函数副作用不考
- 第六章
- 第七章
- 数组、String、ArrayList、输入和输出
- 第八章
- 版本控制、调试

考点

- 第九章

- 顺序图不考

- 第十章

- 第十一章

- JUnit单元测试

- 集成

- 第十二章

- CRC卡不考

- 第十三章

- 第十四章

- 部署

知识点

- 顺序、选择、循环

- 文件读写

- String

- ArrayList

- 数据建模、算法建模

- 递归、高阶函数

- Lambda演算

- 程序正确性验证

- 软件开发生命周期

- 结构化方法和面向对象方法

- 整数、浮点数操作

- 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

- 接口

- 针对接口编程

- 可修改性

- 异常

- GUI控件、布局、事件响应

- 网络、线程

- Java 字节码 解析

重点

- 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）

问答

- lambda演算主要考推演过程
- 高阶函数主要考思想，不在乎具体语言
- 用例图、数据流图、结构图、类图会考，标准UML
- JUnit不考