面向对象的软件构造导论

第一章

- 软件包含哪三个部分？
  - 程序+数据+文档
- 测试驱动的基本过程包含哪三个部分？
  - 红灯（失败）+绿灯（成功）+重构
- 面向过程与面向对象的出发点分别是什么？
  - 面向过程：怎么做；面向对象：是什么
- 面向对象三大特性？
  - 封装+继承+多态
- Java语言有什么特点？简单说几个。
  - 与平台无关（一次编译，到处运行）；面向对象；安全性（废除指针）；执行效率低（解释性语言）。

第一章

- 设计一个下五子棋的系统， 面向对象和结构化编程的思想分别是什么？
  - 结构化思想：开始游戏，黑子先走，绘制画面，判断输赢，白子走，绘制画面，判断输赢，循环。
  - 面向对象思想：黑白双方，棋盘系统（负责绘制画面），规则系统（负责判断输赢）。
- 如果增加悔棋功能，二者的优缺点是什么？
  - 结构化思想：从输入到显示到判断，整个步骤都要改动。
  - 面向对象思想：棋盘系统加一个回溯的功能，黑白双方和规则系统都不用改，体现低耦合特性，代码更容易被复用。

第二章

- Java 引用数据类型有哪些？
  - 类+接口+数组
- int型表数范围是多少？
  - -2^31-2^31-1(约21亿)
- 当低级别的值赋给高级别的变量时，一定不会发生精度丢失的情况，对吗？
  - 不对，有可能丢失。
- Java异常处理是否包含错误？
  - 包含。异常（Exception）和错误（Error）都是异常处理的重要子类。
- Java采用的是隐式分配器还是显式分配器的？
  - 隐式的

第三章

- 类的构造方法的返回值是void?
  - 类的构造方法没有任何返回值，包括void。
- 类的构造方法是否可以被static修饰?
  - 不可以，构造方法是用于创建对象的，static是跟类相关的。
- 一个构造方法下可以使用**多个**this()来调取类内其它构造方法，对吗?
  - 不对，这样相当于创建了多个对象。
- 当一个类中某个成员变量使用访问修饰符default时，该类的子类**不能**访问这个成员变量，对吗?
  - 不一定，如果子类跟这个类在一个包中，就可以访问。
- Java数组在声明时可以指定其长度?
  - 不可，只能在初始化时声明长度。

第四章

- 子类用super关键字调用父类的属性和方法时，需要子类有相同的属性和方法吗？
  - 不需要
- 子类是否可以直接继承父类的构造方法？
  - 不可以，只可以通过super关键词调用
- 被static修饰的方法有没有super关键字？
  - 没有
- 一个构造方法下可以同时有this()和super()来调取其它构造方法吗？
  - 不可以
- 抽象类一定包含抽象方法，对吗？
  - 不对，不一定
- 抽象类和接口的主要相同点是什么？
  - 都不能被实例化

第三章

- 局部变量，成员变量和静态变量的区别是什么？

```
public class Car {
        private String color; // 成员变量：定义在类中，方法体之外
           static String country = “CN”; // 静态变量：由static修饰的变量

        public Car(String color){
        this.color = color;
}
    public void run() {
        String name =“比亚迪” // 局部变量：定义在方法体，构造方法，语句块中的变量
              System.out.println(this.color +“比亚迪在马路上行驶着");
        }
}
```

第三章

- 局部变量能否和成员变量重名？
  - 可以，局部变量可以与成员变量重名，这时可用“this”来指向成员变量

```
public class Test {
    private String a = “成员变量”;
    public void useA() {
        String a = “局部变量”
        System.out.println(this.a);
        System.out.println(a);
    }
    public static void main(String[] args) {
        Test t = new Test();
        t.useA();
    }
}
// 输出：
// 成员变量
// 局部变量
```

第三章

- 局部变量能否和静态变量重名？
  - 可以，局部变量可以与静态变量重名，这时可用“类名”来指向静态变量

```
public class Test {
    private static String a = “静态变量”;
    public void useA() {
        String a = “局部变量”
        System.out.println(Test.a);
        System.out.println(a);
    }
    public static void main(String[] args) {
        Test t = new Test();
        t.useA();
    }
}
// 输出：
// 静态变量
// 局部变量
```

第五章

- 单例模式中访问修饰符的选择问题
  - 1.属性：private，因为属性需要封装；2.构造方法：private，因为不能让其它类调用构造方法创建对象；3.成员方法：public，因为需要其它类调用该方法来访问唯一实例。
- 单例模式中static的选择问题
  - 1.属性：static，因为需要全局唯一；2.构造方法：不能static，因为构造方法是用于创建对象的，static是跟类相关的；3.成员方法：static，因为需要其它类不创建对象就能调用该方法来访问唯一实例。

第五章

- 单一职责原则？开闭原则？依赖倒转原则？

一个类只负责一个职责；对扩展开放，对修改关闭；针对接口/抽象类编程，具体依赖抽象

- 单例模式：饿汉优缺点？懒汉优缺点？同步锁优缺点？

线程安全，浪费资源；延迟加载，线程不安全；线程安全，增加开销（锁只在初始化时有意义）

- 简单工厂模式和工厂模式分别满足单一职责和开闭原则吗？

单一职责都满足，简单工厂模式不满足开闭，工厂模式满足开闭

- 工厂模式有什么优缺点？

横向扩展（增加加盟店）方便，纵向扩展（在加盟店增加产品）困难，除非新开加盟店

- 单例模式和工厂模式属于行为型，创建型还是结构型模式？

创建型

- 抽象工厂模式有什么优缺点？

增加产品族容易（增加具体工厂和产品），增加产品类型困难（增加抽象工厂里的抽象方法）

第六章

- 条件组合覆盖，条件覆盖和判定/条件覆盖谁覆盖度更强，为什么？

条件组合覆盖，条件覆盖不一定满足判定覆盖，判定/条件覆盖没有考虑计算机对多个条件的执行。

- 白盒和黑盒测试又叫什么？

结构测试/逻辑驱动的测试；功能测试/数据驱动的测试。

- 黑盒测试中等价类划分在设计测试用例时为什么要一次性覆盖所有有效等价类？

如果测试通过，所有需求都满足，提高了测试效率。

- 黑盒测试中等价类划分在设计测试用例时为什么要一次性覆盖一个无效等价类？

如果覆盖多个，测试失败，无法确定是哪个无效等价类导致的失败，导致定位失败。

- 在黑盒测试的边界值分析方法中，多边界测试和单边界测试哪种能帮助定位错误？

单边界测试。

第七章

- 数组和集合的区别是什么？
  - 数组：长度不可变，可存放基本数据类型和引用数据类型；集合：长度可变，只可存放引用数据类型。
- 数组和集合是否都能装不同类型的数据？
  - 都可，只是数组需要声明Object，而集合无需声明。
- Hashset和HashMap的无序指的是什么？
  - 无序指的是输入输出是无序的，但是内部其实是有序的。
- 策略模式中环境类（上下文类）的作用是什么？
  - 屏蔽高层模块（客户端类）对策略（算法）的直接访问，封装可能存在的变化（减少耦合，封装变化）。
- 策略模式是创建型模式还是行为型模式，为什么？
  - 行为型，因为它会根据对应的对象来执行对应的方法，更加关注行为的选择。

第八章

- 输入和输出相对哪里定义的？
  - 内存（程序），流入内存的是输入，流出内存的是输出。
- FilterInputStream 是字符流还是字节流？
  - 字节流，只要是Stream结尾都是字节流。
- 为什么要将字节流转为字符流？
  - 1.字符人是可以理解的；
  - 2.有些编码是变长的，用字节流无法确认读几个字节，而字符流可以确认。
- 字节流和字符流是怎么转换的？
  - 在输入中，InputStreamReader负责将字节流转为字符流；
  - 在输出中，OutputStreamWriter负责将字符流转为字节流。
- 在序列化中，可以直接利用Serializable接口完成**序列化**，对吗？
  - 不对，是调用ObjectOutputStream对象的writeObject输出可序列化对象。Serializable接口仅仅只是做一个标记用它告诉代码只要是实现了Serializable接口的类都是可以被序列化的。

第八章

- 我们先定义一个类，定义一个静态变量 staticVar = 5,将它被序列化后，再把staticVar 设置成10，最后反序列化后得出的对象的staticVar 是5还是10呢？
  - 10，是因为在序列化的时候，并不保存静态变量，因为序列化保存的是对象状态，而static变量时属于类的状态，因此序列化并不保存静态变量。
- 父类和子类代码中静态代码块、初始化块和构造方法的执行顺序是什么？
  - 1.父类的静态代码块 2.子类的静态代码块 3.父类的初始化块 4.父类的构造方法 5.子类的初始化块 6.子类的构造方法
- 接口的属性为什么要设置为静态常量，以及可以有什么属性呢？
  - 接口定义了一种**统一的协议**，通常是**不同类通用**属性和行为更高层的抽象，**不通用**的放在自己的类中。
- 怎么调用父类的父类？
  - 要调用Java中父类的父类的方法，可以使用super关键字来直接调用父类的方法，或者通过间接调用覆盖方法的父类来调用超级父类的方法。

第九章

- 在MVC模式中，哪个部分包含DAO层？为什么？
  - 模型，因为模型部分要负责数据的操作，而DAO层正好封装了数据库的操作。
- MVC模式是否体现了单一职责原则，如果有是怎么体现的呢？
  - 体现了，MVC中将模型，视图和控制器封装在了不同的类中以减少耦合。
- Swing中是怎么利用MVC模式的？
  - Swing中监听器可以作为控制器，在监听到用户的操作后，选择调用模型中的方法或者视图中的方法来进行响应。
- 轻量级容器可以不包含在其它的容器中，对不对？
  - 不对，轻量级容器必须包含在重量级容器中。
- Swing不是线程安全的，对不对？
  - 对，所有Swing组件及其相关类都必须保证在同一个线程（事件分派线程）中进行访问。

第十章

- 一个应用只能有一个进程，对吗？
  - 不对。
- 实现Runnable接口可以实现资源共享，继承Thread类不可以实现资源共享，对吗？
  - 都可实现资源共享，只是通过继承Thread类创建多线程时，每个任务有成员变量时不共享，必须加static才能做到共享。
- 想用中断线程时可以直接调用interrupt方法实现中断。
  - 不行，直接调用interrupt方法只会让中断标注变为true。
- wait(time)方法可不可以被notify()/notifyAll()唤醒？
  - 可以，有两种唤醒方式。
- 生产者和消费者之间的同步和协作为什么不能通过sleep方法让它们停止生产和消耗？
  - sleep方法必须指定时间，但是无法通过时间来确定唤醒生产者和消费者。
  - sleep方法不会释放锁，则别的线程无法进当前同步方法。

第十章

- 为什么在多线程实现中实现Runnable接口比继承Thread类降低开销呢？
  - Runnable相当于一个任务，而Thread才是真正的处理线程，我们需要的只是定义这个任务，然后将任务交给线程去处理，这样就达到了松耦合，也符合面向对象的思想，另外也节省了开销，继承Thread的同时，不仅拥有了任务的方法run()，还继承了其它所有的方法。综合来看，用Runnable比Thread好的多。
- 在Java中，synchronized关键字只能对对象加锁吗？
  - 不是的，还可以对方法、类等加锁（课程没有涉及）。

第十一章

- 泛型方法必须定义在泛型类中，对吗？
  - 不对，二者没有依赖关系。
- 泛型类中含有类型参数的静态方法必须是泛型方法，对吗？
  - 对，泛型类中静态方法的泛型必须独立于泛型类的泛型。
- 泛型类中的静态方法必须是泛型方法，对吗？
  - 不对，泛型类中可以有普通（不带类型参数）的静态方法。
- 泛型通配符可以在非泛型类中，也可以在泛型类中定义，对吗？
  - 对，二者不冲突。
- 一个类可以有多个类型对象，对吗？
  - 不对，只能有一个。
- 是否可以通过Class类反射抽象类和接口信息？
  - 可以。

第十二章

- 在网络编程中，一个服务器端是否可以为多个客户提供服务，如果可以怎么实现？
  - 可以，使用多线程机制。
- 套接字（Socket）允许两台计算机之间通过字符流来通信，对吗？
  - 不对，是字节流。