软件工程笔记

第一章 软件工程

1. 软件工程定义
    - 系统的、规范的、可量化的 开发、运行和维护
2. 科学和工程的区别：P6
    - 科学重在把握规律性
    - 成熟的工程学需要科学知识的指导
3. 软件开发活动 P11
    - 需求开发
    - 软件设计
    - 软件构造
    - 软件测试
    - 软件交付与维护

第二章 发展

1. 发展：
    - 50年代： 科学计算，以机器为中心进行编程；像生产硬件一样生产软件
    - 60年代：业务应用（批量数据处理和事务计算）；软件不同于硬件；用软件工艺方式生产软件。

- 70年代：结构化方法；瀑布模型；强调规则和纪律。
- 80年代：追求生产力最大化；现代结构化方法/面向对象编程广泛应用；重视过程的作用。
- 90年代：企业为中心的大规模软件系统开发；追求快速开发、可变更性和用户价值；Web应用出现
- 00年代：大规模Web应用;大量面向大众的Web穿品；追求快速开发、可变更性、用户价值和创新。

第四章 项目管理基础

1. 项目的特征：
    - 明确的目标
    - 限定的开始和结束日期
    - 成本限制
    - 消耗人力和非人力资源
    - 多工种合作
2. 项目管理的目标
    - 时间
    - 成本
    - 质量水平
    - 资源
    - 客户认可
3. 软件项目管理
    1. 项目启动
    2. 项目计划
    3. 项目执行
    4. 项目跟踪与控制
    5. 项目收尾
4. 项目管理具体活动：
    - 计划制定
    - 团队管理
    - 成本控制
    - 质量保障
    - 度量
    - 过程控制
    - 进度跟踪与控制
    - 风险管理
    - 配置管理
5. 团队：
    - 共同的目标
    - 共担责任
    - 技能互补
    - 小规模团体
6. 团队结构：
    - 主程序员团队

    - 民主团队
    - 开放团队
7. 团队建设措施
    - 建立团队章程
    - 持续成功
    - 和谐沟通
    - 避免团队杀手
8. 质量属性：选用系统的某些**质量要素**进行量化处理，建立**质量特征**，这些特征被称为质量属性
9. 质量模型：人们从很多质量属性的定义中选择一些能够相互配合、相互联系的**特征集**，它们被称为质量模型。
10. 质量验证的方法：评审、测试、质量度量
11. 主要质量保障活动

里程碑	质量保障活动
需求开发	需求评审、需求度量
体系结构	体系结构评审、体系结构度量、集成测试（持续集成）
详细设计	详细设计评审、设计度量、集成测试
实现（构造）	代码评审、代码度量、集成测试
测试	测试、测试度量

12. 配置管理：用技术和管理的指导和监督方法，来记录和说明配置项的功能和物理特征，控制这些特征的变更，记录和报告变更处理及其实现状态，并验证与需求规格的一致性。
13. 配置项：置于软件配置管理之下的软件配置的各种有关项目，包括各类管理文档、评审记录和文档、软件文档、源码及其可执行码、运行所需的系统软件和支持软件以及相关数据等。
14. 基线：已经通过正式评审的规格说明或制品，可以作为下一步开发的基础，并且只有通过正式的变更控制过程才能变更。
15. 软件配置管理的活动：
    - 标识配置项
    - 版本管理
    - 变更控制
    - 配置审计
    - 状态报告
    - 软件发布管理
16. 配置管理工具

    SVN、CVS、VSS、ClearCase

第五章 软件需求基础

1. 需求工程的任务：
    - 说明软件系统将被应用的环境极其目标

    - 将目标和功能反映到软件那系统当中，映射为可行的软件行为，并对软件行为进行准确的规格说明
    - 妥善处理目标和功能随时间演化的变动情况
2. 需求工程活动
    1. 需求开发
        - 需求获取
        - 需求分析
        - 需求规格说明
        - 需求验证
    2. 需求管理
3. 需求获取
    1. 目标分析
        1. 根据问题确定目标
        2. 通过分析利害关系人确定目标
    2. 用户需求获取
        - 面谈
        - 集体获取方法
        - 头脑风暴
        - 原型
4. 需求分析中，需求工程师的人物
    1. 边界分析：系统用例图 & 上下文图
    2. 需求建模: 数据流图 & 实体关系图& 状态转换图& 类图
5. 需求验证的 标准
    - 正确、准确地反映用户意图
    - 需求集在整体上具有完整性和一致性
    - 可读性和可修改性
6. 需求验证方法：
    - 统计评审
    - 原型
    - 模拟
7. 需求：需求就是用户的一种期望，软件系统通过满足用户的期望来解决用户的问题。
8. 需求的层次性：
    - 业务需求：组织为什么要开发系统
    - 用户需求：帮助用户做什么
    - 系统级需求：用户对系统行为的期望
9. 需求、问题域和规格说明
    - 需求是一种期望，源于现实、高于现实
    - 问题域是对现实世界运行规律的一种反映，是需求的产生地，也是需求的解决地
    - 规格说明是软件产品的方案描述。

    忽视需求：软件系统单纯模拟现实而不是改变现实，丢失了软件产品的价值

    忽视问题域：脱离现实构建软件系统，使产品无法投入使用
10. 软件需求的分类：

    - 功能
    - 性能：速度、容量、负载、吞吐量、实时性
    - 质量属性：可靠性、可用性、安全性、可维护性、可移植性、易用性
    - 对外接口
    - 约束
11. 功能需求和非功能需求：
    - 功能需求是和系统主要工作相关的需求，即在不考虑物理约束的情况下，用户希望系统能够执行的活动，这些活动可以帮助用户完成任务
    - 非功能需求：除功能需求（和数据需求）之外的其他四种需求

第六章 需求分析方法

1. 需求分析的任务：
    1. **建立需求模型**，达成开发者和用户对需求信息的共同理解
    2. 依据共同理解，发挥创造性，**创建软件系统解决方案**、
2. 需求分析方法：
    - 结构化方法
        - 数据流图
        - 实体关系图
    - 面向对象方法
        - 用例图
        - 类图
        - 交互图
        - 状态图
3. 数据流图：外部实体、过程、数据流、数据存储
4. 用例图：用例、参与者、关系、系统边界‘
5. 用例描述
    - 参与者
    - 触发条件
    - 前置条件
    - 后置条件
    - 正常流程
    - 扩展流程
    - 特殊需求
6. 概念类图：
    - 对象
        - 标识符
        - 状态
        - 行为
    - 类
    - 链接
    - 关联(聚合)

    - 继承
7. 交互图
    - 顺序图
    - 系统顺序图
8. 消息：同步、异步、返回消息
9. 状态图

PPT预习题

1. 典型团队结构有哪些?
2. 什么是“配置项”?
3. 项目管理的目标是什么?
4. 变更控制的过程是什么?

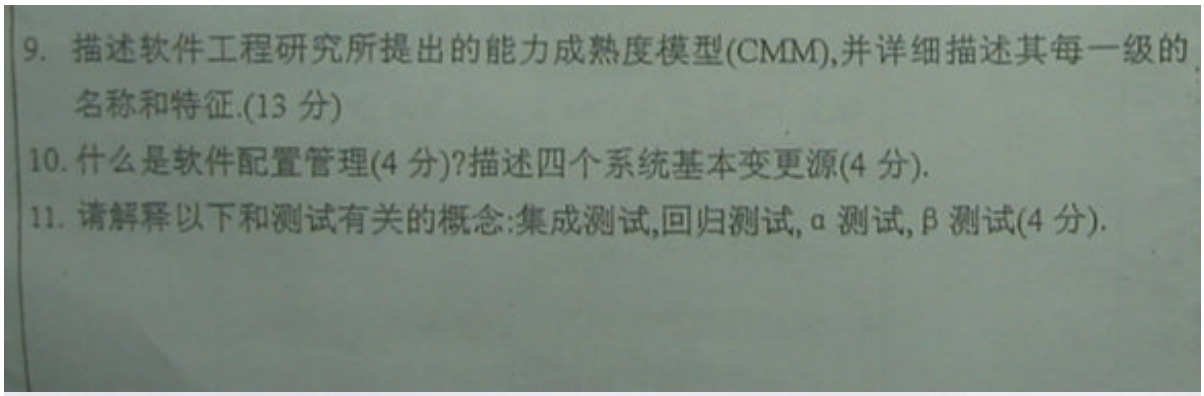
真题

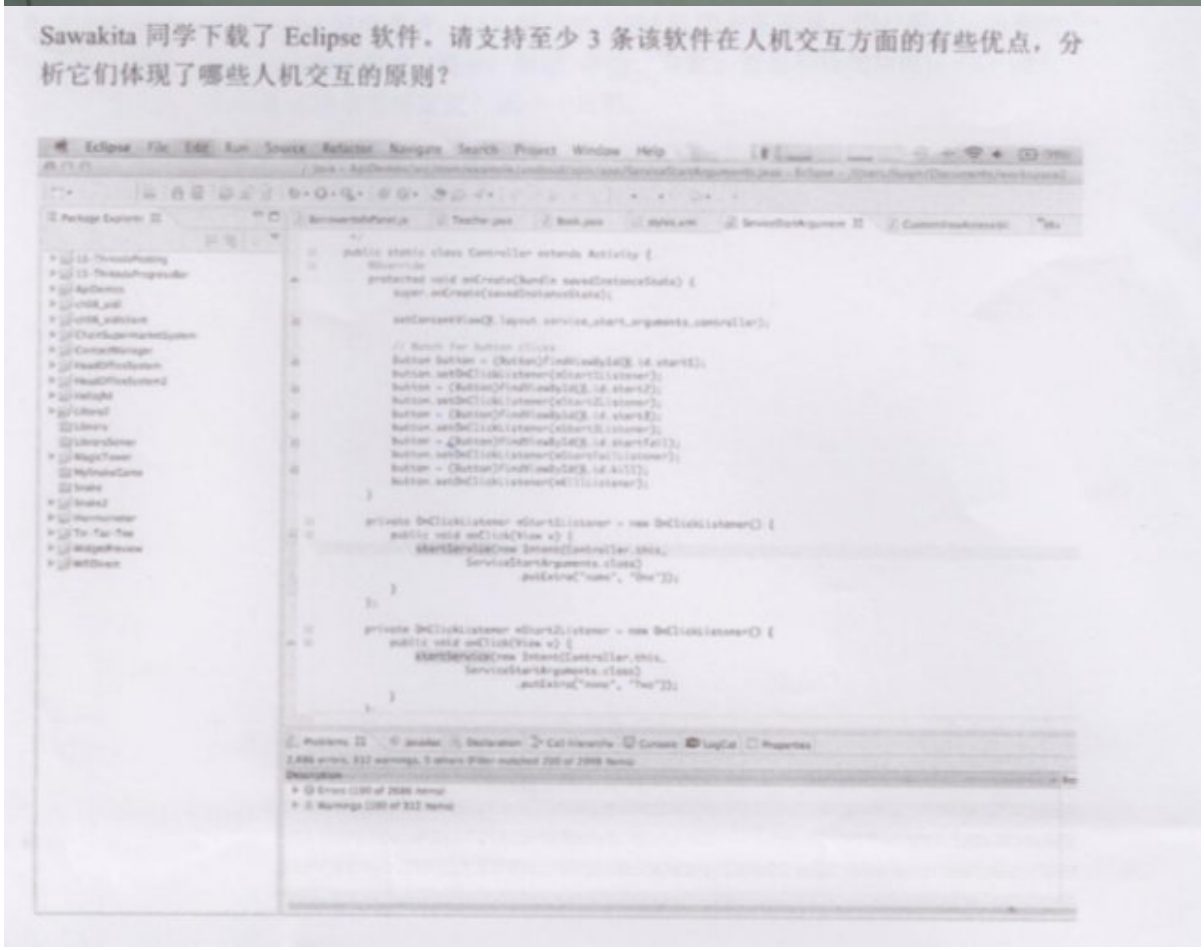
1. 名词解释

1. 软件工程
2. 软件需求
3. 持续集成
4. 软件演化生命周期模型
5. 螺旋模型+优缺点+图示
6. 软件设计

2. 简答题

1. 解释面向对象体系结构的风格及其优缺点
2. 软件质量保障常用的三种手段
3. 给5个需求描述，说出他们分别是哪种类型的需求
4. 需求分为哪几个层次?
5. 根据图书馆管理系统各举一个每个需求的例子
6. 什么是黑盒测试?
7. 有哪些黑盒测试的方法?
8. 描述软件体系结构的分层风格
9. 人机交互的原则

10. 

11. 

12. 结合实验，说明一个项目的质量保障包括哪些活动

3. 代码改进

1. 代码质量低的地方

```
class A{
    FinancialReport fr;
    WeatherData wd;
    Count totalCount;
    init();
}
init(){
    //对fr的3个初始操作
    //对wd的3个初始操作
    //对totalCount的1个初始操作
}//把几个字段的init都放到同一个方法里面
```

2. 下面的代码违反了哪个面向对象原则，有什么后果，应如何改进

```
public class Employee{
    private string EmployeeName {get ; set ;}
    private int EmployeeNo {get;set;}
    public Employee insert(){
        //datebase logic code
    }
    public Employee FindByID(){
        //database logic code;
    }
    public void GengerateReport(){
        //set reportFormation
    }
}
```

3. 分析下面这个类的设计是否合理，如果合理，解释原因，不合理分析原因并作出修改

```
public class Person{
    String name;
    public GetAge(){};
}
```

4. 下列是计算雇员所得税代码，请从交互和写作的角度分析代码是否合理

```
//题目的代码
public class Employee{
    Double income;
    Double getTax(){
        return income*tax.getTaxRate();
    }
}
public class Tax{
    Double taxrate;
    Double getTaxRate(){
```

```
        return taxRate;
    }
}
```

这个代码的问题应该是：计算所得税的代码不应该暴露在Employee中

```
// 答案的代码
public class Employee{
    double income;
    double getIncome(){return income};
    double getTax{
        Tax tax = new Tax();
        return tax.getTax();
    }
}
public class Tax{
    double taxRate;
    double getTaxRate(){
        return taxRate;
    }
    double getTax{
        return employee.getIncome()* tax.getTaxRate()
    }
}
```

不考虑语法错误，感觉这个代码也是毫无逻辑。

```
// 下面是我写的
public class Employee{
    Double income;
    Double getIncome(){return income;}

    double getTax(){
            Tax tax = new Tax();
            return tax.getTax(income);
    }
}
public class Tax{
    Double taxRate;
    Double getTaxRate(){
        return taxRate;
    }
    double getTax(double income){
            return taxRate * income;
    }
}
```

5. 下列是网络选课系统的部分代码，请从面向对象角度使用多态对以下代码进行合理修改

```
processCmd (int cmdID){
    switch (cmdID){
      case 1:addCourse();break;
      case 2:removeCourse();break;
      ...
    }
}//有大量 switch-case 语句
```

6. 消除以下代码的重复

```
private getTotalSum{
    //...
}
```

7. 找出代码质量不高的地方，详细说明其问题，并进行改进

8.

Sakuragi 开发了一个手机应用，准备投放到 Apple Appstroe 和 Google Play 市场中去，下面是他应用的部分关于应用描述的代码，请分析其设计是否合理，是否违反某些设计原则，是否能够应用某种设计模式来重构。

1）指出违反的原则，请解释该原则，并给出修改后的代码

2）解释该设计模式，写出应用该设计模式后的代码

```
class Application {
    prative String applicatioName;
    prative float avarageRate;
    prative ArrayList<NewFeature> newFeatureItems = new ArrayList<NewFeature>();

    String getDescriptionForiOS(){
        StringBuffer result = new StringBuffer();

        result.append("This is "+ applicatioName + " for iOS platform\n");

        for(int i =0; i< newFeatureItems.size(); i++){
            result.append(newFeatureItems.get(i).getDescription());
        }

        result.append("Avarage Rate from App Store\n");
        result.append(String.valueOf(avarageRate));
        return result.toString();
    }
    String getDescriptionForAndroid(){
        StringBuffer result = new StringBuffer();

        result.append("This is "+ applicatioName + " for Android platform\n");

        for(int i =0; i< newFeatureItems.size(); i++){
            result.append(newFeatureItems.get(i).getDescription());
        }

        result.append("Avarage Rate from Google Play\n");
        result.append(String.valueOf(avarageRate));
        return result.toString();
    }
}
```

9.

Rukawa 同学开发了一个影片出租店用的程序，其中需要计算客户的积分。如果电影是新发布的电影并且租用的时间超过 1 天，则可以得到 2 点积分，否则是 1 点积分。

1）请画出下列代码设计的顺序图。

2）指出其是否违反某些设计原则，解释这些原则

3）对其代码进行修改，写出修改之后的代码并画出修改之后的顺序图。

```
public class Customer {
    Rental rental;
    int getNewRentPoint(){
        Movie m = rental.getMovieRented();
        if((m.getPriceCode()== Movie.NEW_RELEASE)&& rental.getDaysRented()>1){
            return  2;
        }
        else return 1;
    }
}
public class Rental {
    private int daysRented;
    private Movie movieRented;

    public int getDaysRented(){
        return daysRented;
    }
    public Movie getMovieRented(){
        return movieRented;
    }
}
public class Movie {
    private int priceCode;
    public static final int CHILDRENS = 2;
    public static final int REGULAR = 0;
    public static final int NEW_RELEASE = 1;

    public int getPriceCode(){
        return priceCode;
    }
}
```

10.

11.

数据结构栈有四个功能：压栈、弹栈、得到栈的大小、得到栈是否为空。Akagi 同学使用继承如下设计了栈。

```
public class MyStack extends Vector {
    public   void push(Object element){
        insertElementAt(element,0);
    }
    public Object pop(){
        Object result = firstElement();
        removeElementAt(0);
        return result;
    }
}
```

Kogure 同学在设计雇员类的时候，如下设计：

```
public Person{
    private string name;
    public string getName(){
        return name;
    }
}
public class Employee extends Person {
}
```

1）指出两个关于继承的设计是否合理？是否违反设计原则？

2）对两段代码，如果合理，请解释其合理性。如果违反，请解释该原则，并修改

12.

Miyagi 写出如下代码

```
void validate_request(input_form i){
    if(!valid_string(i.name)){
        error_message("Invalid name");
    }
    if(!valid_month(i.date)){
        error_message("Invalid month");
    }
}
int valid_month(date d){
    return d.month >=1 && d.month<=12;
}
```

1) validate_request 方法和 valid_month 方法之间是哪种类型的耦合，如何修改？

Mitsui 随后对 Miyagi 做了下列修改

```
void validate_request(input_form i){
    if(!valid (i.name, STRING)){
        error_message("Invalid name");
    }
    if(!valid(i.date, DATE)){
        error_message("Invalid month");
    }
}
int valid(String s, int type){
    switch(type){
        case STRING:
            return strlen(s)<MAX_STRING_SIZE;
        case DATE:
            date d = parse_date(s)
            return d.month >=1 && d.month<=12;
}
```

2) validate_request 方法和 valid 方法之间是哪种类型的耦合，如何修改？

Fujima 开发一个个人所得税系统，代码如下，其中缴税的规则是

- For the first $10,000 of income, the tax is 10%
- For the next $10,000 of income above $10,000, the tax is 12 percent
- For the next $10,000 of income above $20,000, the tax is 15 percent
- For the next $10,000 of income above $30,000, the tax is 18 percent
- For any income above $40,000, the tax is 20 percent

```
tax = 0.
if (taxable_income == 0) goto EXIT;
if (taxable_income > 10000) tax = tax + 1000;
else{              tax = tax + .10*taxable_income;
     goto EXIT;
}
if (taxable_income > 20000) tax = tax + 1200;
else{      tax = tax + .12*(taxable_income-10000);
     goto EXIT;
}
if (taxable_income > 30000) tax = tax + 1500;
else{      tax = tax + .15*(taxable_income-20000);
     goto EXIT;
}
if (taxable_income < 40000){
     tax = tax + .18*(taxable_income-30000);
     goto EXIT;
}
else
     tax = tax + 1800. + .20*(taxable_income-40000);
EXIT;
```

1） 改进这个设计，给出改进的代码。

13.

4. 画图

1. 分析ATM的取款操作，画出用例图，并编写**用例描述**
2. 根据用例描述画出类图

3.

6. 体系结构题

1. 某一系统能实现如下功能，将一组字符串交替执行大小写转换。例如 I love this game 转化成 I LoVe ThIs GaMe 根据某种体系结构，给出系统物理设计模块依赖图，并解释 相应模块的职责。这题给了一个图例，split 指向 lower、upper，然后 upper 指向 merge
2.

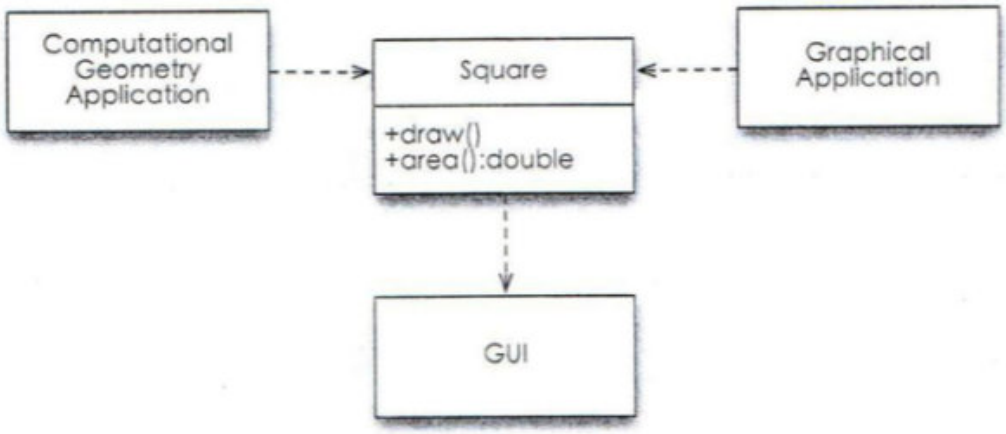
44.（15 分）在某数学软件中，用以下类表示一个长方形。

```
public class Rectangle{
    public double length;
    public double width;

    public double getLength(){
    }
    public double getWidth(){
    }
    public void setLenth(double l){
    }
    public void setWidth(double w){
    }
    public double getArea(){
        return length* width;
    }
}
```

（1）现在要设计一个正方形类，可以继承自 Rectangle 类么？请说明理由，并给出实现正方形类的代码。

（2）现在由于需要画出这个正方形，所以需添加 draw()方法。该 Square 类分别被几何计算应用类和图形应用类所使用。Square 的 draw()方法依赖于 GUI 类来实现。为此有人作出如下设计。请问这样的设计符合面向对象的原则么？请给出理由，如果不符合请给出修改方案，包括画出 UML 设计图，及写出相应的代码。



3.

（10 分）数学上，有理数（Rational number）是一个整数 a 和一个非零整数 b 的比，通常写作 a/b，故又称作分数。a 是被除数（Dividend），b 是除数（Divisor）。有理数集对加、减、乘、除四则运算是封闭的。

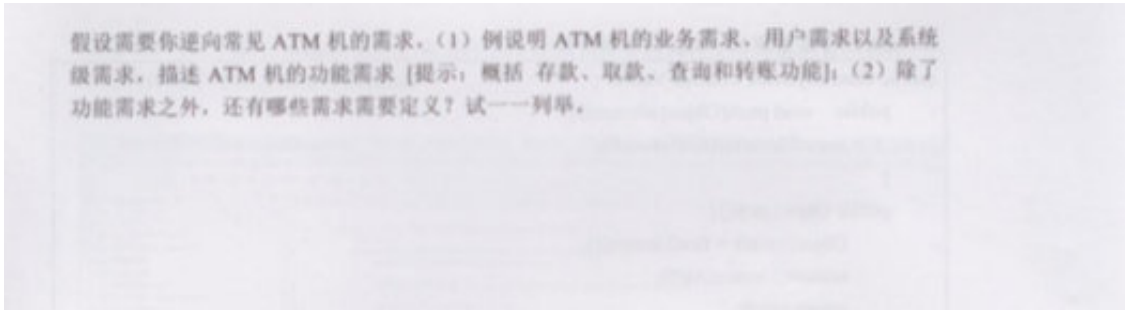
（1）根据面向对象封装的思想设计一个有理数类，用以进行有理数的四则运算。写出这个类的完整代码实现。（实现时整数用 int 类型表示，不考虑无穷大的整数和计算超出 int 类型范围的情况）

（2）写出对该类的除法进行单元测试的用例的代码实现。

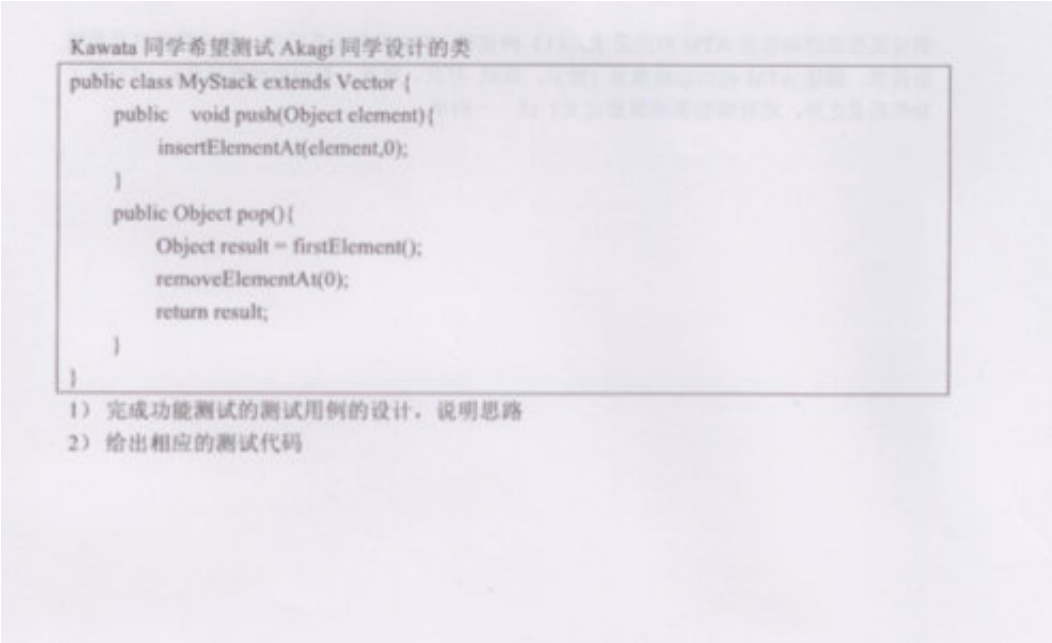
4. 一道关于契约式设计和防御式编程的代码修改题。

   这道题题目很长，具体我不太记得了，大概就是讲取款的操作，有几个条件:1取款金额必须是 100 的整数倍 2 每次取款金额不能超过 3000 3每天 的取款金额不能超过两万

   (具体代码太长我忘了，这部分我没复习到，瞎写的，题目估计没多大 帮助，教训就是一定要全面复习，感觉重点的类图，设计测试用例，都没考

5. 

6.



7.

7. 选择题 （谁知道考不考呢）

1. 有一个方法 int fun(A&a,int i) 的代码完全是顺序语句，那么最适合他的软件测试技术是： a 边界值分析 b 等价类划分 c 随机测试 d 语句覆盖
2. 软件程序设计时，最为重要的代码质量是： a时间性能和空间性能 b 可靠性 c 易读性 d 安全性
3. 下面的类是哪种内聚 a 过程内聚 b 功能 c 时序 d 逻辑

```
class Output{
    public :
            //outputs a  financial report
            void outputreport(financedata);
            //outputs the current weather
            void outputweather (weatherdata);
            //output a number in a nice formatted way
            void outputint(int );
}
```

4.

13. 在面向对象需求分析中可以用来建立用例的有效方法是：（）
A. 事件和事物
B. 涉及的用户角色及其任务
C. 功能分解
D. 面向对象设计原则

14. 下面哪一个不是软件体系结构的逻辑元素？（）
A. 部件　B. 模块　C. 连接件　D. 配置

15. 下面哪一个视图是软件详细设计文档中不需要描述的？（）
A. 用例图　B. 顺序图　C. 类图　D. 包图

16. 下面的类 Output 是哪种类型的内聚？（）

```
class Output {
    public:
      // outputs a financial report
      void outputreport(financedata);
     // outputs the current weather
     void outputweather(weatherdata);
     // output a number in a nice formatted way
     void outputint(int);
};
```

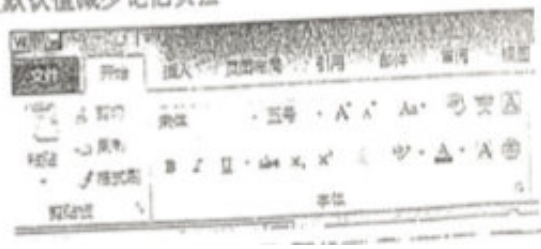
A. 过程内聚　B. 功能内聚　C. 时序内聚　D. 逻辑内聚

17. 下面哪一条面向对象设计原则的描述是错误的？（）
A. LSP 要求继承关系必须实现多态
B. DIP 会使得软件设计中增加很多抽象接口，
C. 一个类只有一个功能即为 SRP
D. 将一个通用的接口分割为多个具体的接口即为 ISP

18. 软件程序设计时，最为重要的代码质量是：（）
A. 时间性能和空间性能　B. 可靠性　C. 易读性　D. 安全性

19. 有一个方法 int fun(A & a, int i)的代码完全是顺序语句，没有任何分支结构，那么最适合它的软件测试技术是：（）
A. 边界值分析　B. 等价类划分　C. 随机测试　D. 语句覆盖

20. 下图的界面没有体现哪条人机交互设计原则？（）
A. 快速反馈
B. 通过直观识别减少记忆负担
C. 通过逐层展开的方式减少记忆负担
D. 通过设置默认值减少记忆负担



5.

11. 下面哪一个软件开发过程模型是文档驱动的？（）
A. 敏捷过程　B. 瀑布模型　C. 演化模型　D. 螺旋模型

12. 下列需求书写正确的是：（）
A. 系统应该容易使用
B. 操作员应该在 2 个小时内完成车辆加油
C. 操作员完成加油后，系统自动进行费用计算：费用=单价×升
D. 增值税的计算要符合国家相关法律

6.

11. “收银员输入购买的商品时，系统要显示该商品的描述、单价、数量和总价。”属于（）层次的需求。

A. 业务需求　　B. 非功能性需求　　C. 用户需求　　D. 系统需求

12. 对需求工程的下列说明，哪个是对的（）：

A. 当需求获取时，需求工程师和用户对于系统应该具有的功能意见不一致时，应当听从用户的，因为用户是上帝。

B. 涉众（Stakeholder），客户（Customer）和用户（User）是同一个概念。

C. 对于需求规格说明评审会议，用户不一定需要参与。

D. 需求的变更需要得到需求变更控制委员会的同意。

13. 在某大学学籍管理信息系统中，假设学生年龄的输入范围为16—40，则根据黑盒测试中的等价类划分技术，下面划分正确的是（）：

A. 可划分为2个有效等价类，2个无效等价类

B. 可划分为1个有效等价类，2个无效等价类

C. 可划分为2个有效等价类，1个无效等价类

D. 可划分为1个有效等价类，1个无效等价类

14. 关于软件构造的下列描述哪个是对的（）：

A. 结对编程中执行者（Driver）和观察者（Observer）两个角色是不能互换的。

B. 重构并非重头开始编写，并不包括对系统的所有修改。

C. 在开发前先写测试用例就是测试驱动开发。

D. 评审代码的时候尽量保持一个较高的评审速度，这样能够评审更多的代码。

15. 有一个类XmlEditor，现在要引入它的父类，以下哪一种命名方式比较好（）：

A. Editor　　B. AEditor　　C. IEditor　　D. XmlEditorSuperClass

16. 软件测试的目的是（）：

A. 发现软件开发中错误的存在

B. 避免软件开发中出现的错误

C. 尽可能定位并排除软件中潜藏的错误，提高软件的可靠性

D. 修改软件中出现的错误

17. 在系统集成测试中，使用（）来替换某些模块。它一般和所替代的模块有相同的接口，并且模拟实现了模块的行为。由于是模拟实现，所以相对于真实的实现要简单很多。

A. 桩　　B. 驱动　　C. Mock Object　　D. 客户端代码

18. 下列不属于软件项目管理活动的是（）：

A. 计划制定　　B. 质量保障　　C. 度量　　D. 项目启动

7.

19. 软件生存周期过程中，修改错误代价最大的阶段是（）：

A. 需求阶段　　B. 设计阶段　　C. 编程阶段　D. 发布运行阶段

20. 关于人机交互描述不正确的是（）：

A. 如果一个系统的大多数用户都是新手用户，整个系统的人机交互设计要侧重于易学性。

B. 人机交互的目标是让计算机控制人，而不是让人控制计算机。

C. 常见的界面类型包括批处理、命令行、全屏、图形化、多维交互等。

D. 精神模型就是使用用户进行人机交互时头脑中的任务模型。人机交互设计需要依据精神模型进行隐喻设计