

全国计算机技术与软件专业技术资格（水平）考试

2013 年下半年 软件设计师 上午试卷

（考试时间 9：00～11：30 共 150 分钟）

请按下述要求正确填写答题卡

1. 在答题卡的指定位置上正确写入你的姓名和准考证号，并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。
2. 本试卷的试题中共有 75 个空格，需要全部解答，每个空格 1 分，满分 75 分。
3. 每个空格对应一个序号，有 A、B、C、D 四个选项，请选择一个最恰当的选项作为解答，在答题卡相应序号下填涂该选项。
4. 解答前务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项，如需修改，请用橡皮擦干净，否则会导致不能正确评分。

例题

● 2013 年上半年全国计算机技术与软件专业技术资格（水平）考试日期是 (88) 月 (89) 日。

(88) A. 3 B. 4 C. 5 D. 6

(89) A. 20 B. 21 C. 22 D. 23

因为考试日期是“5 月 20 日”，故 (88) 选 C，(89) 选 A，应在答题卡序号 88 下对 C 填涂，在序号 89 下对 A 填涂（参看答题卡）。

●在程序执行过程中, Cache 与主存的地址映像由(1)。

- (1) A. 硬件自动完成 B. 程序员调度
C. 操作系统管理 D. 程序员与操作系统协同完成

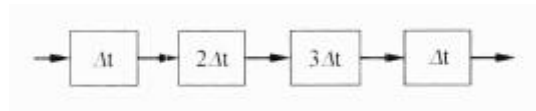
●指令寄存器的位数取决于(2)。

- (2) A. 存储器的容量 B. 指令字长 C. 数据总线的宽度 D. 地址总线的宽度

●若计算机存储数据采用的是双符号位(00 表示正号、11 表示负号), 两个符号相同的数相加时, 如果运算结果的两个符号位经(3)运算得 1, 则可断定这两个数相加的结果产生了溢出。

- (3) A. 逻辑与 B. 逻辑或 C. 逻辑同或 D. 逻辑异或

●某指令流水线由 4 段组成, 各段所需要的时间如下图所示。连续输入 8 条指令时的吞吐率(单位时间内流水线所完成的任务数或输出的结果数)为(4)。



- (4) A. $8/56\Delta t$ B. $8/32\Delta t$ C. $8/28\Delta t$ D. $8/24\Delta t$

●(5)不是 RISC 的特点。

- (5) A. 指令种类丰富 B. 高效的流水线操作 C. 寻址方式较少 D. 硬布线控制

●若某计算机字长为 32 位, 内存容量为 2GB, 按字编址, 则可寻址范围为(6)。

- (6) A. 1024M B. 1GB C. 512M D. 2GB

●下列网络攻击行为中, 属于 DoS 攻击的是(7)。

- (7) A. 特洛伊木马攻击 B. SYN Flooding 攻击 C. 端口欺骗攻击 D. IP 欺骗攻击

●PKI 体制中, 保证数字证书不被篡改的方法是(8)。

- (8) A. 用 CA 的私钥对数字证书签名 B. 用 CA 的公钥对数字证书签名
C. 用证书主人的私钥对数字证书签名 D. 用证书主人的公钥对数字证书签名

● 下列算法中，不属于公开密钥加密算法的是(9)。

- (9) A. ECC B. DSA C. RSA D. DES

● 矢量图是常用的图形图像表示形式，(10)是描述矢量图的基本组成单位。

- (10) A. 像素 B. 像素点 C. 图元 D. 二进制位

● 视频信息是连续的图像序列，(11)是构成视频信息的基本单元。

- (11) A. 帧 B. 场 C. 幅 D. 像素

● 以下多媒体素材编辑软件中，(12)主要用于动画编辑和处理。

- (12) A. WPS B. Xara3D C. PhotoShop D. Cool Edit Pro

● 为说明某一问题，在学术论文中需要引用某些资料。以下叙述中，(13)是不正确的。

- (13) A. 既可引用发表的作品，也可引用未发表的作品
B. 只能限于介绍、评论作品
C. 只要不构成自己作品的主要部分，可适当引用资料
D. 不必征得原作者的同意，不需要向他支付报酬

● 以下作品中，不适用或不受著作权法保护的是(14)。

- (14) A. 某教师在课堂上的讲课
B. 某作家的作品《红河谷》
C. 最高人民法院组织编写的《行政诉讼案例选编》
D. 国务院颁布的《计算机软件保护条例》

● 以下关于数据流图中基本加工的叙述，不正确的是(15)。

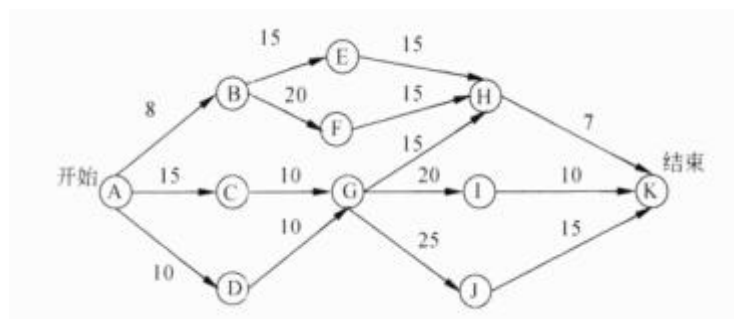
- (13) A. 对每一个基本加工，必须有一个加工规格说明

- B. 加工规格说明必须描述把输入数据流变换为输出数据流的加工规则
- C. 加工规格说明必须描述实现加工的具体流程
- D. 决策表可以用来表示加工规格说明

●在划分模块时，一个模块的作用范围应该在其控制范围之内。若发现其作用范围不在其控制范围内，则(16)不是适当的处理方法。

- (16)A. 将判定所在模块合并到父模块中，使判定处于较高层次
- B. 将受判定影响的模块下移到控制范围内
 - C. 将判定上移到层次较高的位置
 - D. 将父模块下移，使该判定处于较高层次

●下图是一个软件项目的活动图，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，则里程碑(17)在关键路径上。若在实际项目进展中，活动AD在活动AC开始3天后才开始，而完成活动DG过程中，由于有临时事件发生，实际需要15天才能完成，则完成该项目的最短时间比原计划多了(18)天。



- (17)A. B B. C C. D D. I
- (18)A. 8 B. 3 C. 5 D. 6

●针对“关键职员在项目未完成时就跳槽”的风险，最不合适的风险管理策略是(19)。

- (19)A. 对每一个关键性的技术人员，要培养后备人员
- B. 建立项目组，以使大家都了解有关开发活动的信息
 - C. 临时招聘具有相关能力的新职员
 - D. 对所有工作组织细致的评审

●程序运行过程中常使用参数在函数(过程)间传递信息, 引用调用传递的是实参的(20)。

- (20) A. 地址 B. 类型 C. 名称 D. 值

●已知文法 $G: S \rightarrow A0|B1, A \rightarrow S1|1, B \rightarrow S0|0$, 其中 S 是开始符号。从 S 出发可以推导出(21)。

- (21) A. 所有由 0 构成的字符串 B. 所有由 1 构成的字符串
C. 某些 0 和 1 个数相等的字符串 D. 所有 0 和 1 个数不同的字符串

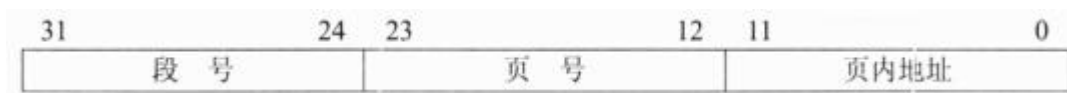
●算术表达式 $a+(b-c)*d$ 的后缀式是(22) ($-$ 、 $+$ 、 $*$ 表示算术的减、加、乘运算, 运算符的优先级和结合性遵循惯例)。

- (22) A. $b\ c\ -\ d\ * \ a\ +$ B. $a\ b\ c\ -\ d\ * \ +$ C. $a\ b\ +\ c\ -\ d\ *$ D. $a\ b\ c\ d\ -\ * \ +$

●假设系统采用 PV 操作实现进程同步与互斥, 若有 n 个进程共享一台扫描仪, 那么当信号量 S 的值为 -3 时, 表示系统中有(23)个进程等待使用扫描仪。

- (23) A. 0 B. $n-3$ C. 3 D. n

●假设段页式存储管理系统中的地址结构如下图所示, 则系统中(24)。



- (24) A. 页的大小为 4K, 每个段的大小均为 4096 个页, 最多可有 256 个段
B. 页的大小为 4K, 每个段最大允许有 4096 个页, 最多可有 256 个段
C. 页的大小为 8K, 每个段的大小均为 2048 个页, 最多可有 128 个段
D. 页的大小为 8K, 每个段最大允许有 2048 个页, 最多可有 128 个段

●某文件管理系统采用位示图(bitmap)记录磁盘的使用情况。如果系统的字长为 32 位, 磁盘物理块的大小为 4MB, 物理块依次编号为: 0、1、2、位示图字依次编号为: 0、1、2、那么 16385 号物理块的使用情况在位示图中的第(25)个字中描述; 如果磁盘的容量为 1000GB, 那么位示图需要(26)个字来表示。

- (25) A. 128 B. 256 C. 512 D. 1024
- (26) A. 1200 B. 3200 C. 6400 D. 8000

●假设系统中有三类互斥资源 R1、R2 和 R3, 可用资源数分别为 10、5 和 3。在 T0 时刻系统中有 P1、P2、P3、P4 和 P5 五个进程, 这些进程对资源的最大需求量和已分配资源数如下表所示, 此时系统剩余的可用资源数分别为(27)。如果进程按(28)序列执行, 那么系统状态是安全的。

资源 进程	最大需求量			已分配资源数		
	R1	R2	R3	R1	R2	R3
P1	5	3	1	1	1	1
P2	3	2	0	2	1	0
P3	6	1	1	3	1	0
P4	3	3	2	1	1	1
P5	2	1	1	1	1	0

- (27) A. 1、1 和 0 B. 1、1 和 1 C. 2、1 和 0 D. 2、0 和 1
- (28) A. P1→P2→P4→P5→P3 B. P5→P2→P4→P3→P1
- C. P4→P2→P1→P5→P3 D. P5→P1→P4→P2→P3

●(29)开发过程模型最不适用于开发初期对软件需求缺乏准确全面认识的情况。

- (29) A. 瀑布 B. 演化 C. 螺旋 D. 增量

●(30)不是增量式开发的优势。

- (30) A. 软件可以快速地交付
- B. 早期的增量作为原型, 从而可以加强对系统后续开发需求的理解
- C. 具有最高优先级的功能首先交付, 随着后续的增量不断加入, 这就使得更重要的功能得到更多的测试
- D. 很容易将客户需求划分为多个增量

●在对程序质量进行评审时, 模块结构是一个重要的评审项, 评审内容中不包括(31)。

- (31) A. 数据结构 B. 数据流结构
- C. 控制流结构 D. 模块结构与功能结构之间的对应关系

●SEI 能力成熟度模型 (SEICMM) 把软件开发企业分为 5 个成熟度级别, 其中(32)重点关注产品和过程质量。

(32)A. 级别 2: 重复级 B. 级别 3: 确定级 C. 级别 4: 管理级 D. 级别 5: 优化级

●系统可维护性的评价指标不包括(33)。

(33)A. 可理解性 B. 可测试性 C. 可移植性 D. 可修改性

●逆向工程从源代码或 U 标代码中提取设计信息, 通常在原软件生命周期的(34)阶段进行。

(34)A. 需求分析 B. 软件设计 C. 软件实现 D. 软件维护

●一个程序根据输入的年份和月份计算该年中该月的天数, 输入参数包括年份(正整数)、月份(用 1~12 表示)。若用等价类划分测试方法进行测试, 则(35)不是一个合适的测试用例(分号后表示测试的输出)。

(35)A. (2013, 1; 31) B. (0, 1; ‘错误’) C. (0, 13; ‘错误’) D. (2000, -1; ‘错误’)

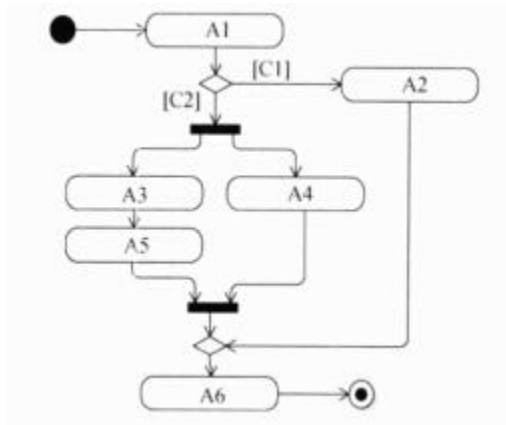
●(36)不是单元测试主要检查的内容。

(36)A. 模块接口 B. 局部数据结构 C. 全局数据结构 D. 重要的执行路径

●在领域类模型中不包含(37)。

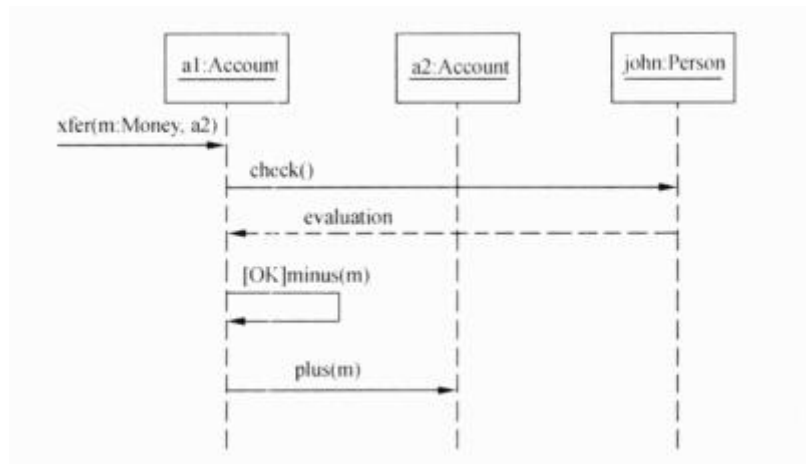
(37)A. 属性 B. 操作 C. 关联 D. 领域对象

●在执行如下所示的 UML 活动图时, 能同时运行的最大线程数为(38)。



- (38) A. 4 B. 3 C. 2 D. 1

● 下图所示的 UML 序列图中，(39)表示返回消息，Account 应该实现的方法有(40)。



- (39) A. xfer B. check C. evaluation D. minus
 (40) A. xfer() B. xfer()、plus() 和 minus()
 C. check()、plus() 和 minus() D. xfer()、evaluation()、plus() 和 minus()

● 在面向对象技术中，(41)定义了超类和子类之间的关系，子类中以更具体的方式实现从父类继承来的方法称为(42)，不同类的对象通过(43)相互通信。

- (41) A. 覆盖 B. 继承 C. 信息 D. 多态
 (42) A. 覆盖 B. 继承 C. 消息 D. 多态
 (43) A. 覆盖 B. 继承 C. 消息 D. 多态

● (44)设计模式定义一系列算法，把它们一个个封装起来，并且使它们可相互替换。

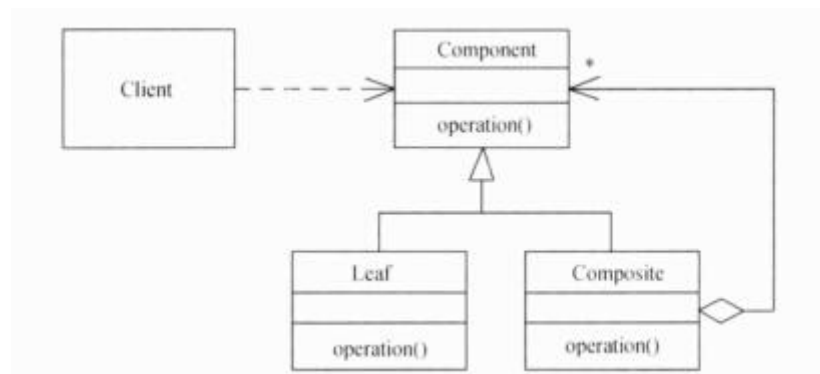
这一模式使得算法可独立于它的客户而变化。

- (44) A. 策略 (Strategy) B. 抽象工厂 (Abstract Factory)
C. 观察者 (Visitor) D. 状态 (State)

●在发布-订阅 (Publish-Subscribe) 消息模型中，订阅者订阅一个主题后，当该主题有最新消息到达时，所有订阅者都会收到通知。(45) 设计模式最适合这一模型。

- (45) A. 适配器 (Adapter) B. 通知 (Notifier)
C. 状态 (State) D. 观察者 (Observer)

●下图所示为(46)设计模式，适用于：(47)。



- (46) A. 组件 (Component) B. 适配器 (Adapter)
C. 组合 (Composite) D. 装饰器 (Decorator)

(47) A. 表示对象的部分-整体层次结构

- B. 不希望在抽象和它的实现部分之间有一个固定的绑定关系
C. 在不影响其他对象的情况下，以动态、透明的方式给单个对象添加职责
D. 使所有接口不兼容类可以一起工作

●将高级语言程序翻译为机器语言程序的过程中，常引入中间代码，其好处是(48)。

- (48) A. 有利于进行反编译处理 B. 有利于进行与机器无关的优化处理
C. 尽早发现语法错误 D. 可以简化语法和语义分析

●对高级语言源程序进行编译的过程中，有穷自动机 (NFA 或 DFA) 是进行(49)的适当工具。

(49) A. 词法分析 B. 语法分析 C. 语义分析 D. 出错处理

●弱类型语言(动态类型语言)是指不需要进行变量/对象类型声明的语言。(50)属于弱类型语言。

(50) A. Java B. C/C++ C. Python D. C#

●若有关系 $R(A, B, C, D, E)$ 和 $S(B, C, F, G)$, 则 R 与 S 自然联结运算后的属性列有(51)个, 与表达式 $\pi_{1, 3, 6, 7}(\sigma_{3 < 6}(RS))$ 等价的 SQL 语句如下:

SELECT (52) FROM (53) WHERE (54);

(51) A. 5 B. 6 C. 7 D. 9

(52) A. A, R, C, F, G B. A, C, S, B, S, F
C. A, C, S, B, S, C D. C, R, A, R, C, S, B, S, C

(53) A. R B. S C. RS D. R, S

(54) A. $R.B=S.B \wedge R.C=S.C \wedge R.D < S.D$ B. $R.B=S.B \wedge R.C=S.C \wedge R.E < S.E$
C. $R.B=S.B \wedge R.C=S.C \wedge R.D < S.D$ D. $R.B=S.B \wedge R.C=S.C \wedge R.E < S.E$

●在分布式数据库系统中, (55)是指用户无需知道数据存放的物理位置。

(55) A. 分片透明 B. 复制透明 C. 逻辑透明 D. 位置透明

●计算机系统的软硬件故障可能会造成数据库中的数据被破坏。为了防止这一问题, 通常需要(56), 以便发生故障时恢复数据库。

(56) A. 定期安装 DBMS 和应用程序

B. 定期安装应用程序, 并将数据库做镜像

C. 定期安装 DBMS, 并将数据库作备份

D. 定期将数据库作备份; 在进行事务处理时, 需要将数据更新写入日志文件

●以下关于线性表存储结构的叙述, 正确的是(57)。

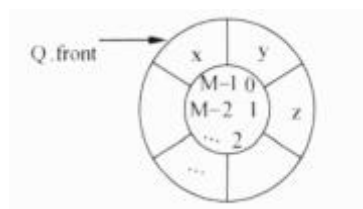
(57) A. 线性表采用顺序存储结构时, 访问表中任意一个指定序号元素的时间复杂度为常量级

B. 线性表采用顺序存储结构时,在表中任意位置插入新元素的运算时间复杂度为常量级

C. 线性表采用链式存储结构时,访问表中任意一个指定序号元素的时间复杂度为常量级

D. 线性表采用链式存储结构时,在表中任意位置插入新元素的运算时间复杂度为常量级

● 设循环队列 Q 的定义中有 front 和 size 两个域变量,其中 front 表示队头元素的指针, size 表示队列的长度,如下图所示(队列长度为 3,队头元素为 x、队尾元素为 z)。设队列的存储空间容量为 M,则队尾元素的指针为(58)。



- (58) A. $(Q.front + Q.size - 1)$ B. $(Q.front + Q.size - 1 + M) \% M$
C. $(Q.front - Q.size)$ D. $(Q.front - Q.size + M) \% M$

● 在一个有向图 G 的拓扑序列中,顶点 V_i 排列在 V_j 之前,说明图 G 中(59)。

- (59) A. 一定存在弧 (v_j, v_i)
B. 一定存在弧
C. 可能存在 v_i 到 v_j 的路径,而不可能存在 v_j 到 v_i 的路径
D. 可能存在 v_j 到 v_i 的路径,而不可能存在 v_i 到 v_j 的路径

● 以下关于哈夫曼树的叙述,正确的是(60)。

- (60) A. 哈夫曼树一定是满二叉树,其每层结点数都达到最大值
B. 哈夫曼树一定是平衡二叉树,其每个结点左右子树的高度差为-1、0 或 1
C. 哈夫曼树中左孩子结点的权值小于父结点、右孩子结点的权值大于父结点
D. 哈夫曼树中叶子结点的权值越小则距离树根越远、叶子结点的权值越大则距离树根越近

●某哈希表(散列表)的长度为 n ，设散列函数为 $H(\text{Key}) = \text{Key} \bmod p$ ，采用线性探测法解决冲突。以下关于 p 值的叙述中，正确的是(61)。

- (61) A. p 的值一般为不大于 n 且最接近 n 的质数 B. p 的值一般为大于 n 的任意整数
C. p 的值必须为小于 n 的合数 D. p 的值必须等于 n

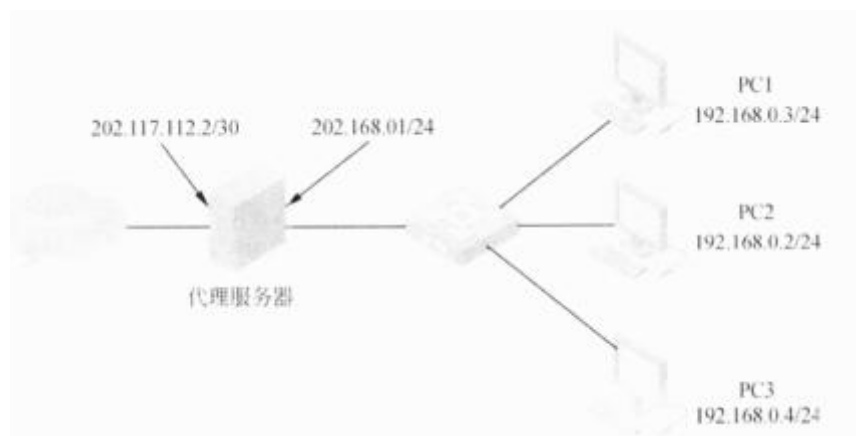
●对 n 个基本有序的整数进行排序，若采用插入排序算法，则时间和空间复杂度分别为(62)；若采用快速排序算法，则时间和空间复杂度分别为(63)。

- (62) A. $O(n^2)$ 和 $O(n)$ B. $O(n)$ 和 $O(n)$ C. $O(n^2)$ 和 $O(1)$ D. $O(n)$ 和 $O(1)$
(63) A. $O(n^2)$ 和 $O(n)$ B. $O(n \lg n)$ 和 $O(n)$ C. $O(n^2)$ 和 $O(1)$ D. $O(n \lg n)$ 和 $O(1)$

●在求解某问题时，经过分析发现该问题具有最优子结构性质，求解过程中子问题被重复求解，则采用(64)算法设计策略；若定义问题的解空间，以深度优先的方式搜索解空间，则采用(65)算法设计策略。

- (64) A. 分治 B. 动态规划 C. 贪心 D. 回溯
(65) A. 动态规划 B. 贪心 C. 回溯 D. 分支限界

●某单位的局域网配置如下图所示，PC2 发送到 Internet 上的报文的源 IP 地址为(66)。



- (66) A. 192.168.0.2 B. 192.168.0.1 C. 202.117.112.1 D. 202.117.112.2

●在 IPv4 向 IPv6 过渡期间，如果要使得两个 IPv6 结点可以通过现有的 IPv4 网络进行通信，则应该使用(67)；如果要使得纯 IPv6 结点可以与纯 IPv4 结点进行通信，则需要使用(68)。

- (67) A. 堆栈技术 B. 双协议栈技术 C. 隧道技术 D. 翻译技术
(68) A. 堆栈技术 B. 双协议栈技术 C. 隧道技术 D. 翻译技术

●POP3 协议采用(69)模式进行通信, 当客户机需要服务时, 客户端软件与 POP3 服务器建立(70)连接。

- (69) A. Browser/Server B. Client/Server C. PeertoPeer D. PeertoServer
(70) A. TCP B. UDP C. PHP D. IP

●There is nothing in this world constant but inconstancy. —SWIFT Project after project designs a set of algorithms and then plunges into construction of customer-deliverable software on a schedule that demands delivery of the first thing built.

In most projects, the first system built is (71) usable. It may be too slow, too big, awkward to use, or all three. There is no (72) but to start again, smarting but smarter, and build a redesigned version in which these problems are solved. The discard and (73) may be done in one lump, or it may be done piece-by-piece. But all large-system experience shows that it will be done. Where a new system concept or new technology is used, one has to build a system to throw away, for even the best planning is not so omniscient (全知的) as to get it right the first time.

The management question, therefore, is not whether to build a pilot system and throw it away. You will do that. The only question is whether to plan in advance to build a (74), or to promise to deliver the throwaway to customers. Seen this way, the answer is much clearer. Delivering that throwaway to customers buys time, but it does so only at the (75) of agony (极大痛苦) for the user, distraction for the builders while they do the redesign, and a bad reputation for the product that the best redesign will find hard to live down.

- (71) A. almost B. often C. usually D. barely
(72) A. alternative B. need C. possibility D. solution
(73) A. design B. redesign C. plan D. build

- (74) A. throwaway B. system C. software D. product
- (75) A. worth B. value C. cost D. invaluable