

全国计算机技术与软件专业技术资格（水平）考试

2023 年上半年 软件设计师 上午试卷

（考试时间 09:00~11:30 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填写你的姓名和准考证号，并粘贴考生条形码。
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）

2023 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. 25 | B. 26 | C. 27 | D. 28 |

因为考试日期是“5 月 27 日”，故 (88) 选 C，(89) 选 C，应在答题卡序号 88 下对 C 填涂，在序号 89 下对 C 填涂（参看答题卡）

(1) A. 接口和外设 B. 运算器、控制器和寄存器
C. CPU、主存及外设部件 D. DMA 控制器和中断控制器

(2) A. 高速缓存地址 B. 主存物理地址
C. 硬盘的扇区地址 D. 虚拟地址

(3) A. 1 B. 2 C. 4 D. 8

(4) A. 中断源的设备地址 B. 中断服务程序的入口地址
C. 传递数据的起始地址 D. 主程序的断点地址

(5) A. 补码零的表示是唯一的 B. 可以将减法运算转化为加法运算
C. 符号位可以与数值位一起参加运算 D. 与真值的对应关系简单且直观

(6) A. 1004ns B. 1998ns C. 2003ns D. 2008ns

(7) A. 传输层 B. 会话层 C. 表示层 D. 应用层

在 PKI 体系中，由 SSL / TLS 实现 HTTPS 应用。浏览器和服务器之间用于加密 HTTP 消息的方式是（8），如果服务器证书被撤销那么所产生的后果是（9）。

- （8） A. 对方公钥 + 公钥加密 B. 本方公钥 + 公钥加密
C. 会话密钥 + 公钥加密 D. 会话密钥 + 对称加密
- （9） A. 服务器不能执行加解密 B. 服务器不能执行签名
C. 客户端无法再信任服务器 D. 客户端无法发送加密信息给服务器

以下关于入侵防御系统功能的描述中，不正确的是（10）。

- （10） A. 监测并分析用户和系统的网络活动
B. 匹配特征库识别已知的网络攻击行为
C. 联动入侵检测系统使其阻断网络攻击行为
D. 检测僵尸网络，木马控制等僵尸主机行为

Web 应用防火墙无法有效防护（11）。

- （11） A. 登录口令暴力破解 B. 恶意注册
C. 抢票机器人 D. 流氓软件

著作权中（12），的保护期不受限制。

- （12） A. 发表权 B. 发行权 C. 展览权 D. 署名权

国际上为保护计算机软件知识产权不受侵犯所采用的主要方式是实施（13）。

- （13） A. 合同法 B. 物权法 C. 版权法 D. 刑法

以下关于计算机软件著作权的叙述中，不正确的是（14）。

- （14） A. 软件著作权人可以许可他人行使其软件著作权，并有权获得报酬
B. 软件著作权人可以全部或者部分转让其软件著作权，并有权获得报酬
C. 软件著作权属于自然人的，该自然人死亡后，在软件著作权的保护期内，继承人能继承软件著作权的所有权利
D. 为了学习和研究软件内含的设计思想和原理，通过安装、显示、传输或者存储软件等使用软件的，可以不经软件著作权人许可，不向其支付报酬

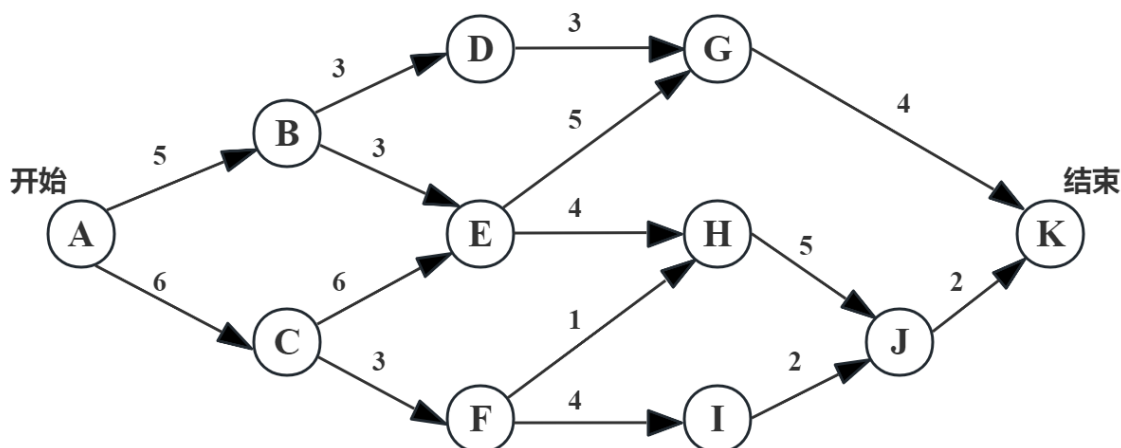
以下关于数据流图基本加工的叙述中，不正确的是 (15)。

- (15) 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. 加工规格说明需要给出实现加工的细节
D. 决策树、决策表可以用来表示加工规格说明

以下关于好的软件设计原则的叙述中，不正确的是 (16)。

- (16) A. 模块化
B. 提高模块独立性
C. 集中化
D. 提高抽象层次

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，关键路径长度为 (18)。



- (17) A. B
B. E
C. G
D. I
- (18) A. 15
B. 17
C. 19
D. 23

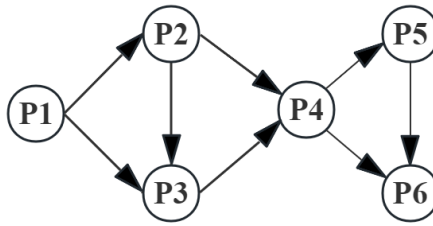
由 8 位成员组成的开发团队中，一共有 (19) 条沟通路径。

- (19) A. 64
B. 56
C. 32
D. 28

对布尔表达式 “a or ((b < c) and d)” 求值时，当 (20) 时可进行短路计算。

- (20) A. a 为 true
B. b 为 true
C. c 为 true
D. d 为 true

进程 P1、P2、P3、P4、P5 和 P6 的前趋图如下所示。



若用 PV 操作控制进程 P1、P2、P3、P4、P5 和 P6 并发执行的过程，需要设置信号量 S1、S2、S3、S4、S5、S6、S7 和 S8，且信号量 S1~S8 的初值都等于零。下面 P1~P6 的进程执行过程中，空①和空②处应分别为 (26) ，空③和空④处应分别为 (27) ，空⑤和空⑥处应分别为 (28) 。

```

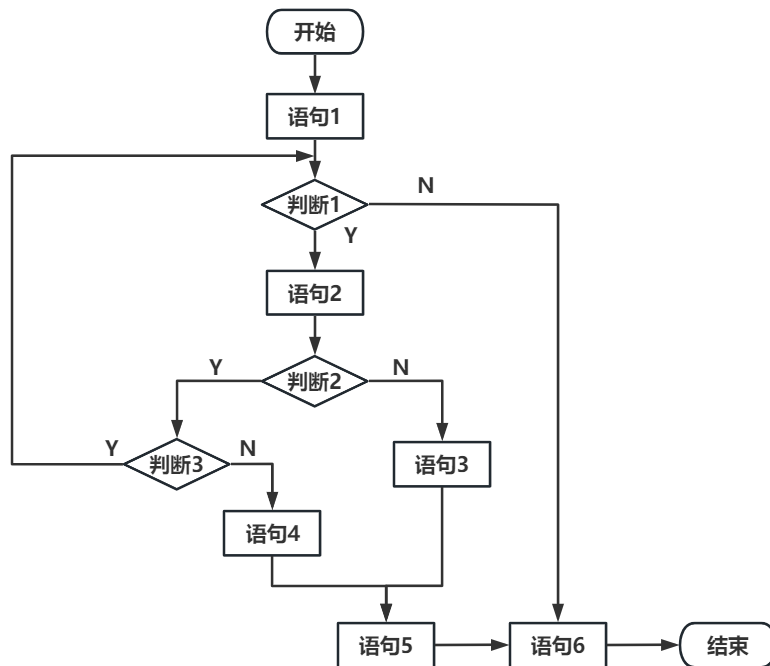
begin
  S1, S2, S3, S4, S5, S6, S7, S8: semaphore; // 定义信号量
  S1:=0; S2:=0; S3:=0; S4:=0; S5:=0; S6:=0; S7:=0; S8:=0;
  Cobegin
    process P1      process P2      process P3      process P4      process P5      process P6
    begin           begin           begin           begin           begin           begin
      P1 执行;      P(S1);          ②              ④              ⑤              P(S7)
      V(S1);        P2 执行;          P3执行;          P4执行;          P5执行;          P(S8)
      V(S2);        ①              ③              V(S6);          ⑥              P6执行;
    end;           end;           end;           end;           end;           end;
  Coend;
end;
    
```

- (26) A. P (S1) P (S2) 和 V (S3) V (S4) B. P (S1) P (S2) 和 V (S1) V (S2)
 C. V (S3) V (S4) 和 P (S1) P (S2) D. V (S3) V (S4) 和 P (S2) P (S3)
- (27) A. V (S5) 和 P (S4) P (S5) B. V (S3) 和 P (S4) V (S5)
 C. P (S5) 和 V (S4) V (S5) D. P (S3) 和 P (S4) P (S5)
- (28) A. V (S6) 和 V (S8) B. P (S6) 和 P (S7)
 C. P (S6) 和 V (S8) D. P (S8) 和 P (S8)

以下关于增量模型优点的叙述中，不正确的是 (29) 。

- (29) A. 能够在较短的时间提交一个可用的产品系统
 B. 可以尽早让用户熟悉系统
 C. 优先级高的功能首先交付，这些功能将接受更多的测试
 D. 系统的设计更加容易

以下流程图中，至少需要（34）个测试用例才能覆盖所有路径。采用 McCabe 方法计算程序复杂度为（35）。



- （34） A. 3 B. 4 C. 5 D. 6
- （35） A. 2 B. 3 C. 4 D. 5

在软件系统交付给用户使用后，为了使用户界面更友好，对系统的图形输出进行改进，该行为属于（36）维护。

- （36） A. 改正性 B. 适应性 C. 改善性 D. 预防性

采用面向对象方法开发学生成绩管理系统，学生的姓名、性别、出生日期、期末考试成绩、查看成绩操作均被（37）在学生对象中。系统中定义不同类，不同类的对象之间通过（38）进行通信。

- （37） A. 封装 B. 继承 C. 多态 D. 信息
- （38） A. 继承 B. 多态 C. 消息 D. 重载

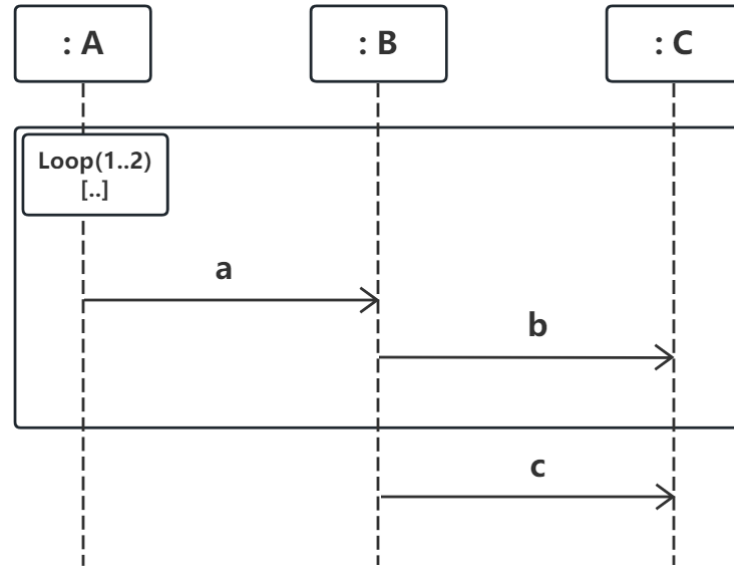
对采用面向对象方法开发的系统进行测试时，通常从不同层次进行测试。测试类中定义的每个方法属于（39）层。

- （39） A. 算法 B. 类 C. 模板 D. 系统

在面向对象系统设计中，如果重用了一个包中的某个类，那么就要重用该包中的所有类，这属于（40）原则。

- （40） A. 共同封闭 B. 共同重用 C. 开放—封闭 D. 接口分离

以下关于 UML 序列图的描述是（41），下图所示 UML 图中消息可能执行的顺序是（42）。



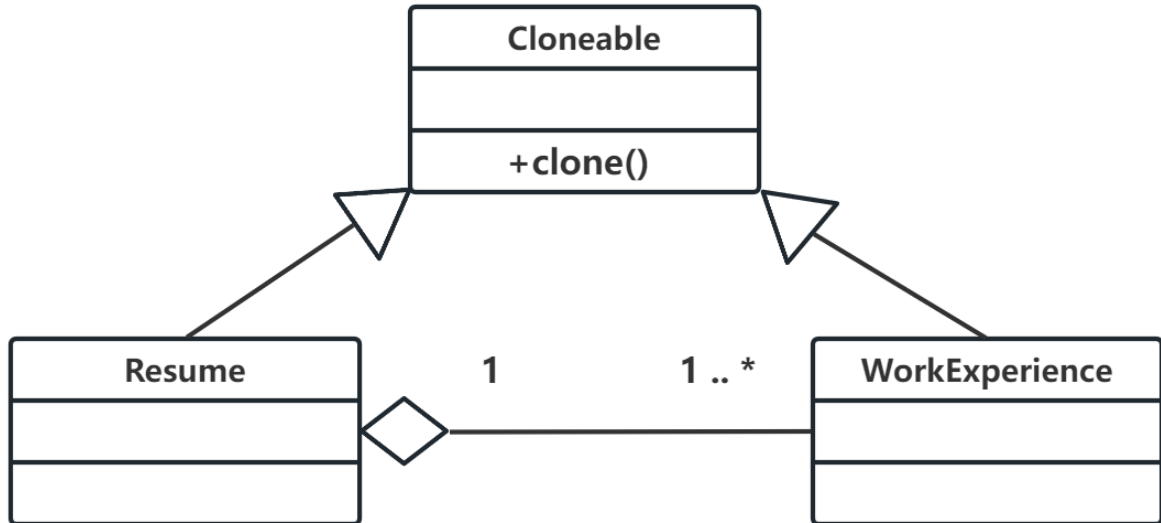
- （41） A. 系统在它的周边环境的语境中所提供的外部可见服务
B. 某一时刻一组对象以及它们之间的关系
C. 系统内从一个活动到另一个活动的流程
D. 以时间顺序组织的对象之间的交互活动

- （42） A. $a \rightarrow b \rightarrow c \rightarrow a \rightarrow b$ B. c
C. $a \rightarrow b \rightarrow a \rightarrow b \rightarrow c$ D. $a \rightarrow b \rightarrow c \rightarrow a \rightarrow b \rightarrow c$

UML 包图展现由模型本身分解而成的组织单元及其依赖关系，以下关于包图的叙述中，不正确的是（43）。

- （43） A. 可以拥有类、接口构件、结点 B. 一个元素可以被多个包拥有
C. 一个包可以嵌套其他包 D. 一个包内元素不能重名

在某招聘系统中，要求实现求职简历自动生成功能。简历的基本内容包括求职者的姓名、性别、年龄及工作经历等。希望每份简历中的工作经历有所不同，并尽量减少程序中的重复代码。针对此需求，设计如下所示的类图。该设计采用了__(44)__模式，由 Cloneable 示例指定创建对象的种类，声明一个复制自身的接口，并且通过复制这些 Resume, WorkExperience 的对象来创建新的对象。该模式属于__(45)__模式。



- (44) A. 单例 (Singleton) B. 抽象工厂 (Abstract Factory)
 C. 生成器 (Builder) D. 原型 (Prototype)
 (45) A. 混合型 B. 行为型
 C. 结构型 D. 创建型

某旅游公司欲开发一套软件系统，要求能根据季节，节假日等推出不同的旅行定价包，如淡季打折、一口价等。实现该要求适合采用__(46)__模式，该模式的主要意图是__(47)__。

- (46) A. 策略模式 (Strategy) B. 状态 (State)
 C. 观察者 (Observer) D. 命令 (Command)
 (47) A. 将一个请求封装为对象，从而可以用不同的请求对客户进行参数化
 B. 当一个对象的状态发生改变时，依赖于它的对象都得到通知并被自动更新
 C. 允许一个对象在其内部状态改变时改变它的行为
 D. 定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换

Python 中采用__(48)__方法来获取一个对象的类型。

- (48) A. str() B. type() C. id() D. object()

在 Python 语言中, 语句 $x = \underline{(49)}$ 不能定义一个元组。

- (49) A. (1, 2, 1, 2) B. 1, 2, 1, 2
C. tuple() D. (1)

关于 Python 语言的叙述中, 不正确的是 (50)。

- (50) A. for 语句可以用在序列 (如列表、元组和字符串) 上进行迭代访问
B. 循环结构如 for 和 while 后可以加 else 语句
C. 可以用 if...else 和 switch...case 语句表示选择结构
D. 支持嵌套循环

在数据库应用系统的开发过程中, 开发人员需要通过视图层、逻辑层次上的抽象来对用户屏蔽系统的复杂性, 简化用户与系统的交互过程。错误的是 (51)。

- (51) A. 视图层是最高层次的抽象
B. 逻辑层是比视图层更低一层的抽象
C. 物理层是最低层次的抽象
D. 物理层是比逻辑层更高一层的抽象

给定关系模式 $R(U, F)$, 其中 U 为属性集, F 是 U 上的一组函数, 属于自反律的是 (52)。

- (52) A. 若 $Y \in X \in U$, 则 $X \rightarrow Y$ 为 F 所蕴含
B. 若 $X \rightarrow Y, Y \rightarrow Z$, 则 $X \rightarrow Z$ 为 F 所蕴含
C. 若 $X \rightarrow Y, Z \in Y$, 则 $X \rightarrow Z$ 为 F 所蕴含
D. 若 $X \rightarrow Y, X \rightarrow Z$, 则 $X \rightarrow YZ$ 为 F 所蕴含

给定关系模式 $R(U, F)$, $U = \{A, B, C, D\}$, 函数依赖集 $F = \{AB \rightarrow C, CD \rightarrow B\}$ 。关系模式 R (53), 主属性和非主属性个数分别为 (54)。

- (53) A. 只有 1 个候选关键字 ACB B. 只有一个候选关键字 BCD
C. 有 2 两个候选关键字 ABD 和 ACD D. 有 2 两个候选关键字 ACB 和 BCD
(54) A. 4 和 0 B. 3 和 1 C. 2 和 2 D. 1 和 3

如果将 Students 表的插入权限赋予用户 User1, 并允许其将该权限授予他人, 那么正确的 SQL 语句如下: GRANT (55) TABLE Students TO User1 (56) 。

- (55) A. INSERT B. INSERT ON C. UPDATE D. UPDATE ON
(56) A. FOR ALL B. PUBLIC
C. WITH GRANT OPTION D. WITH CHECK OPTION

利用栈对算术表达式 $10 * (40 - 30 / 5) + 20$ 求值时, 存放操作数的栈 (初始为空) 的容量至少为 (57), 才能满足暂存该表达式中的运算数或运算结果的要求。

- (57) A. 2 B. 3 C. 4 D. 5

设有 5 个字符, 根据其使用频率为其构造哈夫曼编码。以下编码方案中 (58) 是不可能的。

- (58) A. { 111, 110, 101, 100, 0 } B. { 0000, 0001, 001, 01, 1 }
C. { 11, 10, 01, 001, 000 } D. { 11, 10, 011, 010, 000 }

设有向图 G 具有 n 个顶点、e 条弧, 采用邻接表存储, 则完成广度优先遍历的时间复杂度为 (59)。

- (59) A. $O(n + e)$ B. $O(n^2)$ C. $O(e^2)$ D. $O(n * e)$

对某有序顺序表进行折半查找 (二分查找) 时, 进行比较的关键字序列不可能是 (60)。

- (60) A. 42, 61, 90, 85, 77 B. 42, 90, 85, 61, 77
C. 90, 85, 61, 77, 42 D. 90, 85, 77, 61, 42

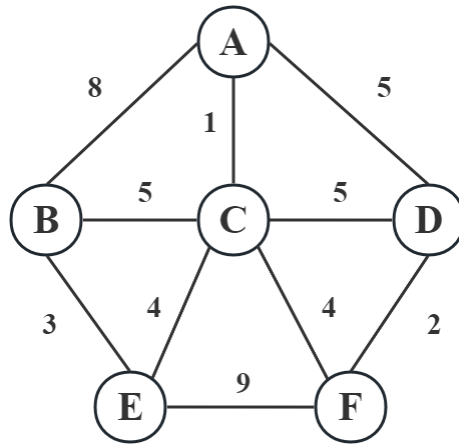
设由三棵树构成的森林中, 第一棵树、第二棵树和第三棵树的结点总数分别为 n_1 、 n_2 和 n_3 。将该森林转换为一颗二叉树, 那么该二叉树的右子树包含 (61) 个结点。

- (61) A. n_1 B. $n_1 + n_2$ C. n_3 D. $n_2 + n_3$

对一组数据进行排序，要求排序算法的时间复杂度为 $O(n \lg n)$ ，且要求排序是稳定的，则可采用 (62) 算法。若要求排序算法的时间复杂度为 $O(n \lg n)$ ，且在原数据上进行，即空间复杂度为 $O(1)$ ，则可以采用 (63) 算法。

- (62) A. 直接插入排序 B. 堆排序
C. 快速排序 D. 归并排序
- (63) A. 直接插入排序 B. 堆排序
C. 快速排序 D. 归并排序

采用 Kruskal 算法求解下图的最小生成树，采用的算法设计策略是 (64)。该最小生成树的权值是 (65)。



- (64) A. 分治法 B. 动态规划 C. 贪心法 D. 回溯法
- (65) A. 14 B. 16 C. 20 D. 32

www 的控制协议是 (66)。

- (66) A. FTP B. HTTP C. SSL D. DNS

在 Linux 操作系统中通常使用 (67)。作为 Web 服务器，其默认的 Web 目录为 (68)。

- (67) A. IIS B. Apache C. NFS D. MYSQL
- (68) A. /etc/httpd B. /var/log/httpd C. /etc/home D. /home/httpd

SNMP 的传输层协议是 (69)。

- (69) A. UDP B. TCP C. IP D. ICMP

某电脑无法打开任意网页，但是互联网即时聊天软件使用正常。造成该故障的原因可能是 (70)。

- (70) A. IP 地址配置错误 B. DNS 配置错误 C. 网卡故障 D. 链路故障

Low-code and no code software development solutions have emerged as viable and convenient alternatives to the traditional development process.

Low-code is a rapid application development (RAD) approach that enables automated code generation through (71) building blocks like drag-and-drop and pull-down menu interfaces. This (72) allows low-code users to focus on the differentiator rather than the common denominator of programming. Low-code is a balanced middle ground between manual coding and no-code as its users can still add code over auto-generated code.

No-code is also a RAD approach and is often treated as a subset of the modular plug-and-play, low-code development approach. While in low-code there is some handholding done by developers in the form of scripting or manual coding, no-code has a completely (73) approach, with 100% dependence on visual tools.

A low-code application platform (LCAP) — also called a low-code development platform (LCDP) — contains an integrated development environment (IDE) with (74) features like APIs, code templates, reusable plug-in modules and graphical connectors to automate a significant percentage of the application development process. LCAPs are typically available as cloud-based Platform-as-a-Service (PaaS) solutions.

A low-code platform works on the principle of lowering complexity by using visual tools and techniques like process modeling, where users employ visual tools to define workflows, business rules, user interfaces and the like. Behind the scenes, the complete workflow is automatically converted into code. LCAPs are used predominantly by professional developers to automate the generic aspects of coding to redirect effort on the last mile of (75).

- (71) A. visual B. component-based C. object-oriented D. structural
(72) A. block B. automation C. function D. method
(73) A. modern B. hands-off C. generic D. labor-free
(74) A. reusable B. built-in C. existed D. well-known
(75) A. delivery B. automation C. development D. success