

## 소프트웨어공학

1. 다음 중 응집도(cohesion)가 가장 높은 것은?

- ① 절차적(procedural) 응집도
- ② 시간적(temporal) 응집도
- ③ 기능적(functional) 응집도
- ④ 논리적(logical) 응집도

2. 다음은 웹사이트 회원가입 양식에 나이를 입력하는 텍스트 박스의 요구사항이다. 경계값 분석 기법을 이용해 테스트 케이스를 작성할 경우, 입력값의 집합으로 옳은 것은?

회원의 나이를 입력하는 텍스트 박스에는 0세 이상, 120세 이하의 값만 입력해야 한다.

- ① 0, 60, 120
- ② -10, 0, 120, 130
- ③ 0, 1, 2, 118, 119, 120
- ④ -1, 0, 1, 119, 120, 121

3. 다음 소프트웨어 개발 방법론과 모델링 도구를 바르게 연결한 것은?

(가) 정보공학 방법론  
(나) 구조적 방법론  
(다) 객체지향 방법론

A. 유스케이스 다이어그램(Usecase Diagram)  
B. E-R 다이어그램(E-R Diagram)  
C. 자료 흐름도(Data Flow Diagram)

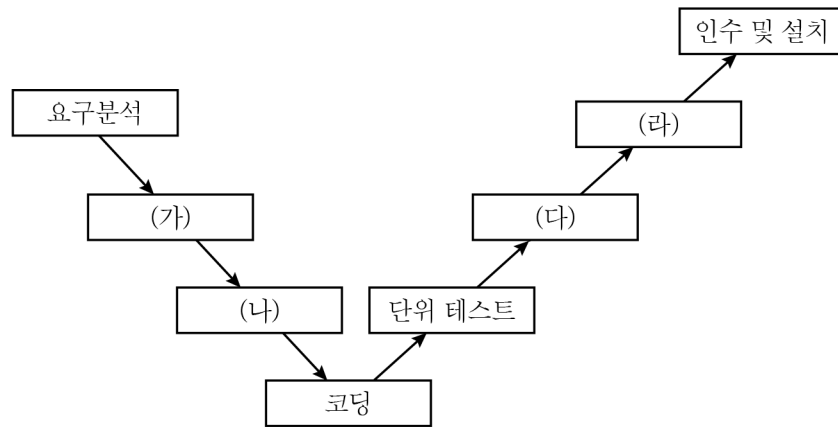
- |   | (가) | (나) | (다) |
|---|-----|-----|-----|
| ① | A   | B   | C   |
| ② | A   | C   | B   |
| ③ | B   | C   | A   |
| ④ | C   | B   | A   |

4. 다음 코드에 대해 조건 커버리지를 100 % 만족하는 테스트 케이스 집합은? (단, 단축연산(short-circuit evaluation)은 수행하지 않는다)

```
if(a > 5 && b == 0)
    printf("첫 번째 조건 충족\n");
if(a == 3 || b < 0)
    printf("두 번째 조건 충족\n");
```

- ① {(a = 2, b = 0), (a = 2, b = 1)}
- ② {(a = 3, b = 0), (a = 4, b = -2)}
- ③ {(a = 6, b = 0), (a = 2, b = 1)}
- ④ {(a = 6, b = 0), (a = 3, b = -1)}

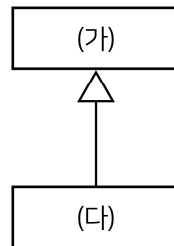
5. V 모델의 단계에서 (가) ~ (라)에 들어갈 내용을 바르게 연결한 것은?



A. 모듈 설계  
B. 시스템 테스트  
C. 아키텍처 설계  
D. 통합 테스트

- |   | (가) | (나) | (다) | (라) |
|---|-----|-----|-----|-----|
| ① | A   | B   | C   | D   |
| ② | A   | C   | B   | D   |
| ③ | C   | A   | B   | D   |
| ④ | C   | A   | D   | B   |

6. 다음은 클래스 다이어그램을 Java 코드로 구현한 것이다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?



```
class Lecture extends (나) {
    public Lecture() {}
}
class Study {
    public Study() {}
}
```

- |   | (가)     | (나)     | (다)     |
|---|---------|---------|---------|
| ① | Study   | Study   | Lecture |
| ② | Study   | Lecture | Lecture |
| ③ | Lecture | Study   | Study   |
| ④ | Lecture | Lecture | Study   |

7. 다음 파이썬 코드에서 문장 커버리지를 100% 만족하는 (가) ~ (다)에 들어갈 경로를 바르게 연결한 것은? (단, 단축연산은 수행하지 않는다)

```
(1) a = int(input('종료 숫자? '))
(2) sum = 0
(3) i = 1
(4) while i <= a:
(5)     if i % 2 == 0:
(6)         sum += i
(7)     i += 1
(8) print('{}까지 짝수합은 {}'.format(a, sum))
```

(1) - (2) - (3) - (4) - (5) - (가) - (나) - (다) - (8)

(가)
(나)
(다)

- |   |     |     |     |
|---|-----|-----|-----|
| ① | (6) | (7) | (5) |
| ② | (6) | (7) | (4) |
| ③ | (7) | (4) | (6) |
| ④ | (7) | (4) | (7) |

8. 객체 지향 설계 원칙에 대한 설명으로 옳지 않은 것은?

- ① 단일 책임(single-responsibility)의 원칙: 클래스를 변경해야 하는 이유는 오직 하나여야 한다.
- ② 개방 폐쇄(open-closed)의 원칙: 모듈은 수정에는 열려 있어야 하고 확장에는 닫혀 있어야 한다.
- ③ 리스코프 대체(Liskov substitution)의 원칙: 상위 클래스의 객체는 자신의 하위 클래스로 대체할 수 있어야 한다.
- ④ 의존 관계 역전(dependency inversion)의 원칙: 클라이언트는 구체적인 클래스가 아닌 추상 클래스에 의존해야 한다.

9. (가)에 해당하는 시암 치댐버(Shyam R. Chidamber)와 크리스 캐머러(Chris F. Kemerer)의 객체지향 설계 척도는?

$$\boxed{(\mathcal{I})} = C_{m1} + C_{m2} + \dots + C_{mn}$$

단,  $C_{mi}(i=1, 2, \dots, n)$ 는 클래스 C에 있는 메소드의 복잡도이다.

- ① NOC(Number Of Children)
- ② WMC(Weighted Methods per Class)
- ③ CBO(Coupling Between Object classes)
- ④ LCOM(Lack of Cohesion in Methods)

10. 시험 운영 중인 보안관제 시스템이 다음 표와 같이 운영되면서 정상 가동 중에 고장이 발생하였을 때, 시스템의 가용성(availability)은?

	가동1	고장1	가동2	고장2	가동3	고장3	가동4
시간(h)	12	4	9	2	10	3	5

- ① 0.2
- ② 0.25
- ③ 0.8
- ④ 0.9

11. 다음과 같이 Java 코드를 변경하였을 때 적용된 기법은?

```
void select(int num)
{
    System.out.println("1. 한국어, 2. 영어");
    if(num == 1)
        System.out.println("한국어");
    else
        System.out.println("영어");
}
```



```
void select(int num)
{
    System.out.println("1. 한국어, 2. 영어");
    lang(num);
}

void lang(int i)
{
    if(i == 1)
        System.out.println("한국어");
    else
        System.out.println("영어");
}
```

- ① 리팩토링(refactoring)
- ② 재사용(reuse)
- ③ 순공학(forward engineering)
- ④ 역공학(reverse engineering)

12. 다음과 같은 조건으로 소프트웨어 개발 프로젝트를 수행할 때, 계약 기간 내에 프로젝트를 성공적으로 수행하기 위해 필요한 최소 기간은?

프로젝트 명	재고관리 시스템 개발			
추정 소스 코드 라인 수	65,000 LOC			
개발자 1인당 월평균 생산성 (LOC/MM)	초급자	1,000	중급자	1,500
현재 투입 가능 개발자 수 (명)	초급자	4	중급자	2

- ① 7개월                      ② 8개월  
③ 9개월                    ④ 10개월

13. 다음 설명에 해당하는 소프트웨어 형상 관리 활동은?

- 형상 관리 계획서에 따라 변경 관리 프로세스가 진행되었는지 평가한다.
- 형상 항목의 변경이 요구사항에 맞게 이루어졌는지 확인한다.

- ① 형상 통제(configuration control)
- ② 형상 식별(configuration identification)
- ③ 형상 감사 및 검토(configuration audits and reviews)
- ④ 형상 상태 보고(configuration status accounting)

14. 프로토타입 모델에 대한 설명으로 옳지 않은 것은?

- ① 프로토타입을 통해 새로운 요구사항을 발견할 수 있다.
- ② 시스템 개발 최종 종료 단계에서만 오류를 탐지할 수 있다.
- ③ 사용자가 초기 프로토타입을 최종 완성된 소프트웨어로 오인할 수 있다.
- ④ 개발자 입장에서 프로토타이핑 과정을 관리·통제하기 어렵다.

15. SPICE(ISO/IEC 15504) 모델에서 정의한 프로세스 성숙도 수준에 속하지 않은 것은?

- ① 수행(performed) 단계
- ② 관리(managed) 단계
- ③ 정의(defined) 단계
- ④ 최적화(optimizing) 단계

16. 다음 Java 코드의 실행 결과는?

```
class Car {
    public void type() {
        System.out.println("Car");
    }
}

class Truck extends Car {
    public void type() {
        System.out.println("Truck");
    }
}

class Campingcar extends Truck {
    public void type() {
        super.type();
        System.out.println("Camping car");
    }
}

public class Main {
    public static void main(String[] args) {
        Car c = new Campingcar();
        c.type();
    }
}
```

- ① Car
- ② Camping car
- ③ Car  
Truck
- ④ Truck  
Camping car

17. 순환 복잡도(cyclomatic complexity)가 4인 Java 코드는?

- ① 

```
int max(int x, int y) {
    int m = 0;
    if(x > y)
        m = x;
    else
        m = y;
    return m;
}
```
- ② 

```
int bonus(int base) {
    int x = 0;
    if(base > 2000)
        x = 3000;
    else if(base > 1000)
        x = 2000;
    else
        x = 1000;
    return x;
}
```
- ③ 

```
int calculate(int x, int y) {
    while(x != y) {
        if(x > y)
            x -= y;
        else
            y -= x;
    }
    return x;
}
```
- ④ 

```
int penalty(int speed) {
    int money = 0;
    if(speed > 200) {
        if(speed < 150)
            money = 50;
        else {
            money = 30;
            if(speed <= 100)
                money = 0;
        }
    }
    return money;
}
```

18. 다음 파이썬 코드에 적용한 디자인 패턴은?

```
class Greeting(object):
    def say(self):
        kor = Korea()
        kor.kor_hi()

        jan = Japan()
        jan.jan_hi()

        chn = China()
        chn.ch_hi()

class Korea(object):
    def kor_hi(self):
        print("안녕하세요\n")

class Japan(object):
    def jan_hi(self):
        print("곤니찌와\n")

class China(object):
    def ch_hi(self):
        print("니하오마\n")

class Hello(object):
    def sayhello(self):
        sh = Greeting()
        sh.say()

insa = Hello()
insa.sayhello()
```

- ① 브릿지(bridge)
- ② 어댑터(adapter)
- ③ 퍼사드(facade)
- ④ 컴포지트(composite)

19. 다음 파이썬 코드에 해당하는 결합도(coupling)는?

```
a = 0
b = 0

def getinput():
    c = int(input("숫자 입력? "))
    return c

def getadd():
    c = a + b
    return c

def main():
    global a
    global b
    a = getinput()
    b = getinput()
    c = getadd()
    print("{} + {} = {}".format(a, b, c))

main()
```

- ① 스탬프(stamp) 결합도
- ② 제어(control) 결합도
- ③ 공통(common) 결합도
- ④ 내용(content) 결합도

20. 다음 세부 품질 요소를 모두 포함하는 ISO/IEC 9126의 소프트웨어 품질 특성은?

- ☐ 보안성
- ☐ 준수성
- ☐ 상호운용성
- ☐ 정확성
- ☐ 적합성

- ① 신뢰성(reliability)
- ② 기능성(functionality)
- ③ 사용성(usability)
- ④ 안정성(stability)

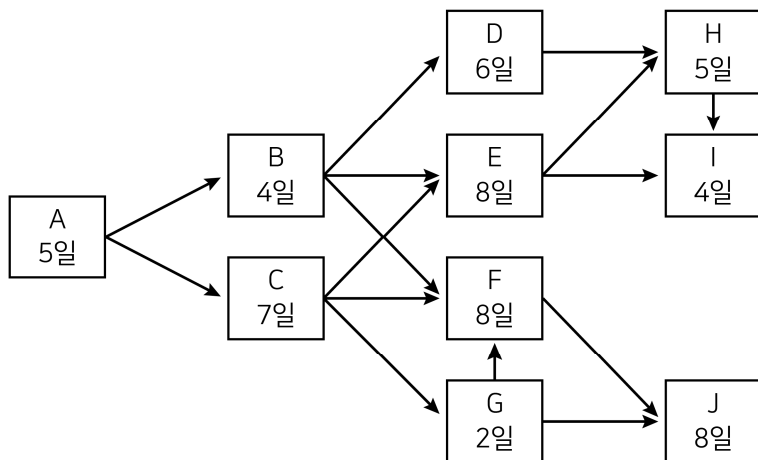
21. WBS(work breakdown structure)에 대한 설명으로 옳지 않은 것은?

- ① 프로젝트를 다수의 구성 요소로 분할하여 계층적으로 기술한 업무 구조이다.
- ② 프로젝트 수행 과정에서 요구되는 자원의 단위와 단가 정보를 작성한다.
- ③ 구성 요소를 산출물 중심으로 작성하거나 프로세스 중심으로 작성할 수 있다.
- ④ 프로젝트 전체 범위를 정의하고 프로젝트 계획과 관리를 위한 자료로 사용된다.

22. 기능점수(function point)에 대한 설명으로 옳지 않은 것은?

- ① 기능점수의 기준이 되는 소프트웨어 기능은 데이터 측면의 기능과 트랜잭션 측면의 기능으로 구분된다.
- ② 기능점수는 소프트웨어 개발 이전에 업무량을 측정할 수 있으며 소프트웨어 개발에 사용하는 언어와는 무관하다.
- ③ 기능점수 측정 대상 유형은 내부 논리파일, 내부 연계파일, 외부 입력, 외부 출력, 외부 조회이다.
- ④ 기능점수 방법은 소프트웨어 개발, 유지관리 및 재개발을 위한 비용을 산정하는 데 사용한다.

23. 다음 CPM 네트워크의 각 노드가 작업명과 작업에 소요되는 예상 시간을 나타낼 때, 프로젝트를 완성하는 데 예상되는 최단 기간은?



- ① 21일
- ② 24일
- ③ 28일
- ④ 30일

24. UML 설계의 4+1 View 모델에 대한 설명으로 옳은 것은?

- ① 논리적 관점은 시스템의 기능을 설명하기 위해 클래스 다이어그램이나 시퀀스 다이어그램을 사용한다.
- ② 개발 관점은 시스템이 사용자에게 제공하는 기능에 주목하며, 유스케이스 다이어그램이나 상태 다이어그램을 사용한다.
- ③ 물리적 관점은 시스템이 어떻게 개발되는지에 대한 관점이며, 클래스 다이어그램이나 유스케이스 다이어그램을 사용한다.
- ④ 프로세스 관점은 시스템에 필요한 하드웨어 환경을 나타내며, 배치 다이어그램이나 클래스 다이어그램을 사용한다.

25. 다음 코드에서 제시된 설계 원칙은?

```

class Gongmuwon {
    private String name;
    private String dept;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void setDept(String dept) {
        this.dept = dept;
    }
}
  
```

- ① 캡슐화
- ② 다형성
- ③ 상속
- ④ 추상화