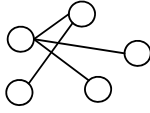
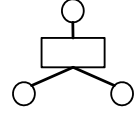
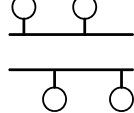
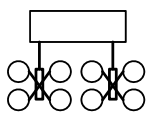


[정보처리기사 실기 - 2020년 3회]

* 2020년 3회 정보처리기사 실기 기출문제 (20문항, 문항당 5점)

한 문제에서 소문자로 파생되는 문제나, 가지수를 요구하는 문제는 대부분의 경우 부분채점을 적용합니다. 그러나 소문자로 파생되는 문제 내에서의 부분 배점은 적용하지 않습니다.

1. EAI 구축유형 중 Message Bus와 Hybrid를 제외한 빈칸에 알맞은 유형을 작성하시오.

유형	개념도	설명
(①)		두 대의 컴퓨터가 직렬 인터페이스를 이용하여 통신을 할 때 사용하는 프로토콜 - 가장 기본적인 애플리케이션 통합 방식 - 변경 및 재사용이 어렵다.
(②)		: 단일 접점인 허브 시스템을 통해 데이터를 전송하는 중앙 집중형 방식 - 확장 및 유지 보수가 용이 - 허브 장애 시 시스템 전체에 영향을 미친다.
Message Bus		: 애플리케이션 사이에 미들웨어를 두어 처리하는 방식 - 확장성, 대용량 처리 우수 (ESB 방식)
Hybrid		: Hub & Spoke와 Message Bus의 혼합 - 그룹 내에서는 Hub & Spoke 방식으로 연결하고, 그룹 간에는 Message Bus 방식으로 연결 - 데이터 병목 현상을 최소화

- 답안: ① PPP ② Hub & Spoke
- 출제: 통합구현 3강
- * PPP = Point to Point Protocol

[정보처리기사 실기 - 2020년 3회 기출]

2. 대표적인 내부 라우팅 프로토콜로, 링크 상태 라우팅 프로토콜로도 불리는 라우팅 프로토콜이다. RIP의 단점을 해결한 프로토콜이며 대규모 네트워크에 적합하다.

- 답안: OSPF(Open Shortest Path First, 최단 경로 우선 프로토콜)
- 출제: 응용 SW 기초 기술 활용 2강

3. 심리학자 톰 마릴은 컴퓨터가 메시지를 전달하고 메시지가 제대로 도착했는지 확인하며, 도착하지 않았을 경우 메시지를 재전송하는 일련의 방법을 가리켜 '기술적 은어'라는 뜻으로 () (이)라 불렀다.

- 답안: 프로토콜(Protocol)

* 프로토콜(Protocol)
: 통신 시스템이 데이터를 교환하기 위해 사용하는 통신 규칙

4. 헝가리안 표기법 개념 설명(약술형)

- 답안: 컴퓨터 프로그래밍에서 변수나 함수의 이름 앞에 데이터 타입을 명시하는 코딩 규칙

* 변수 명명법(Casing)

: 변수 이름의 일관성을 유지하여 코드의 가독성을 높여 이해하기 쉽고, 유지보수를 용이하기 위해 생긴 규칙
- 어떤 방법이 가장 좋은 것이 아닌 프로젝트 상황에 가장 적합한 명명법을 팀에서 결정하여 모든 개발자들이 규칙을 따라 코드를 작성한다.

1. Camel Casing(카멜 케이싱)

- 소문자로 시작
- 단어 사이의 단락을 대문자로 표기

```
String camelCasing;
int size;
```

2. Pascal Casing(파스칼 케이싱)

- 대문자로 시작
- 단어 사이의 단락을 대문자로 표기

```
String PascalCasing;
int Size;
```

[정보처리기사 실기 - 2020년 3회 기출]

3. Snake Casing(스네이크 케이스)

- 소문자로 시작
- 단어 사이의 단락을 _ (언더스코어, 밑줄 문자)로 표기

```
String snake_casing;
int size;
```

5. GNU Naming Convention

- 모든 문자 소문자
- 단어 사이 단락을 _ (언더스코어)로 표기

```
String gnu_naming_convention
int size;
```

4. Hungarian Notation(헝가리안 표기법)

- 이름 앞에 데이터 타입을 명시
- 데이터 타입은 정해진 약어로 표시

```
String strHungarianCasing;
int nSize;
```

6. 상수 표기법

- 대부분의 명명법에서 상수를 표기하는 방법은 거의 동일하다.
- 모든 문자 대문자
- 단어 사이의 단락을 _ (언더스코어)로 표기

```
String MACRO_CASING;
int SIZE;
```

5. 데이터베이스의 스키마(Schema)에 대해 간략히 서술하시오.

- 답안: 데이터베이스의 구조와 제약조건에 대한 명세를 기술한 것
- 출제: SQL 응용 1강

[정보처리기사 실기 - 2020년 3회 기출]

6. 다음 C 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수 하시오.)

```
#include <stdio.h>
```

```
int main() {
    int i, c = 0;
    while(i < 10) {
        i++;
        c *= i;
    }
    printf("%d", c);
}
```

i	c	i < 10
0	0	T
1	0×1=0	T
2	0×2=0	T
:	:	:
9	0×9=0	T
10	0×10=0	F

→ i가 10일 때 while문 종료

- 답안: 0
- 출제: 프로그래밍 언어 활용 1강

7. UI 설계 원칙 중 직관성 개념 설명(약술형)

- 답안: 누구나 쉽게 이해하고 사용할 수 있어야 한다.
- 출제: 화면 설계 1강

8. C++: 생성자 개념 설명(약술형)

- 답안: 객체 생성시 초기화 작업을 위한 함수로써, 객체를 생성할 때 자동으로 호출되고 제일 먼저 실행된다.
new 연산자를 통해서 객체를 생성할 때 반드시 호출되고 제일 먼저 실행되는 일종의 메소드
- 출제: 프로그래밍 언어 활용 4강, 20년 2회 기출풀이 강의

* JAVA, C++, Python 언어에서 생성자의 개념은 동일하다.

[정보처리기사 실기 - 2020년 3회 기출]

9. 형상 통제 개념 설명(약술형)

- 답안: 산출물의 변경사항을 버전별로 관리하여 목표 시스템의 품질 향상을 지원하는 활동
식별된 형상 항목에 대한 변경 요구를 검토하여 현재의 기준선이 잘 반영될 수 있도록 조정하는 작업
- 출제: 서버프로그램 구현 1강

* 형상 관리

: 소프트웨어의 개발 과정에서 발생하는 산출물의 변경 사항을 버전 관리하기 위한 일련의 활동

- 절차: 형상 식별 → 변경 제어 → 형상 상태 보고 → 형상 감사

* 변경 제어 (=형상 통제)

: 식별된 형상항목의 변경 요구를 검토, 승인하여 적절히 통제함으로써 현재의 베이스라인에 잘 반영될 수 있도록 조정하는 작업

- 적절한 형상 통제가 이루어지기 위해서는 형상통제위원회 승인을 통한 통제가 이루어질 수 있어야 한다.

10. TCP/IP 기반의 인터넷 통신 서비스에서 인터넷 프로토콜(IP)과 조합하여 통신 중에 발생하는 오류의 처리와 전송 경로의 변경 등을 위한 제어 메시지를 취급하는 무연결 전송(Connectionless Transmission)용의 프로토콜로 OSI 기본 참조 모델의 네트워크층에 해당한다. (Full name 또는 약어)

- 답안: ICMP(Internet Control Message Protocol)

- 출제: 소프트웨어 개발 보안 구축 1강

11. 다음 <속성 정의서>를 참고하여 <학생> 테이블에 대해 20자의 가변 길이를 가진 '주소' 속성을 추가하는 <SQL 문>을 완성하십시오.

<속성 정의서>

속성명	데이터타입	제약조건	테이블명
학번	CHAR(10)	UNIQUE	학생
이름	VARCAHR(8)	NOT NULL	학생

[정보처리기사 실기 - 2020년 3회 기출]

<SQL문> (①) TABLE 학생 (②) 주소 VARCHAR(20);

- 답안: ① ALTER ② ADD

- 출제: SQL 응용 1강

12. 소프트웨어 공학에서 리팩토링(Refactoring)을 하는 목적에 대해 간략히 서술하십시오.

- 답안: 코드의 외부 행위(동작, 기능)은 바꾸지 않고 내부 구조를 개선시켜 소프트웨어를 보다 이해하기 쉽고, 수정하기 쉽도록 만드는 것이다.

- 출제: 필기 요약-애플리케이션 테스트 관리 1강

* 리팩토링(Refactoring)

: 코드의 외부 행위는 바꾸지 않고 내부 구조를 개선시켜 소프트웨어 시스템을 변경하는 프로세스

- 이미 존재하는 코드의 설계를 안전하게 향상시키는 기술로, 단순히 코딩 스타일만을 개선하는 것이 아니라 성능과 코드의 구조, 즉 좋은 설계가 되도록 개선시키는 과정

- 리팩토링 대상: 중복된 코드, 거대한 클래스, 긴 메소드

* 리팩토링을 하는 목적

- 리팩토링은 소프트웨어의 디자인을 개선시킨다.

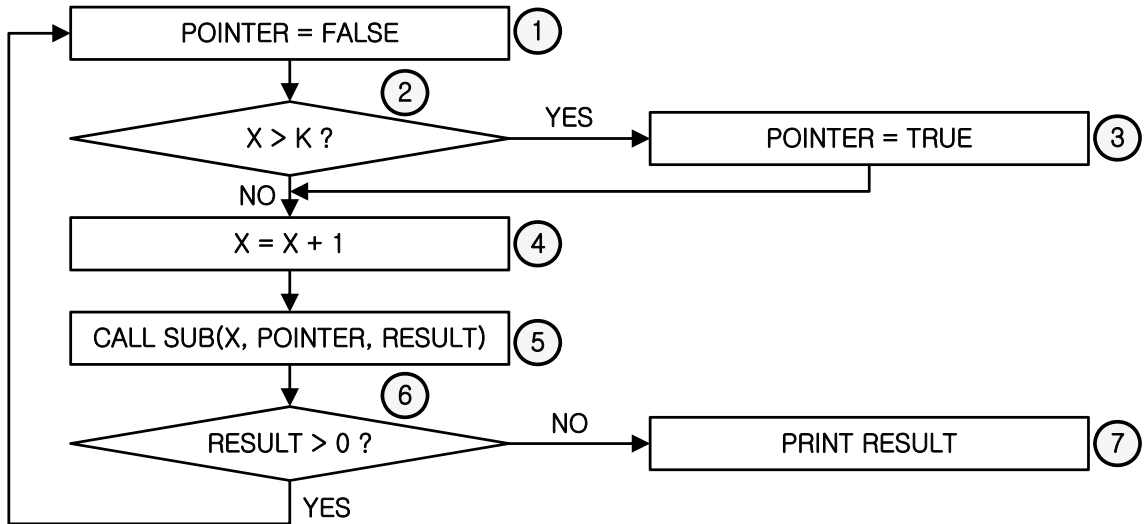
- 리팩토링은 소프트웨어를 이해하기 쉽게 만들어 준다.

- 버그를 빨리 찾을 수 있도록 도움을 준다.

- 리팩토링은 프로그램을 빨리 작성할 수 있도록 도와준다.

[정보처리기사 실기 - 2020년 3회 기출]

13. 분기 커버리지 과정 순서 배열



() → () → () → () → () → () → ()
 () → () → () → () → () → () → ()

- 답안: 1234567, 124561 또는 1234561, 124567
- 출제: 애플리케이션 테스트 관리 1강

[정보처리기사 실기 - 2020년 3회 기출]

* 분기 커버리지(=결정 커버리지)

: 결정 포인트 내의 모든 분기문(조건문)에 대해 수행

→ 조건문 내에 존재하는 조건들의 참(True)과 거짓(False)이 적어도 한 번 이상 실행된다.

테스트 케이스	X > K	RESULT > 0	경로
1	T	T	1 → 2 → 3 → 4 → 5 → 6 → 1
	F	F	1 → 2 → 4 → 5 → 6 → 7

또는

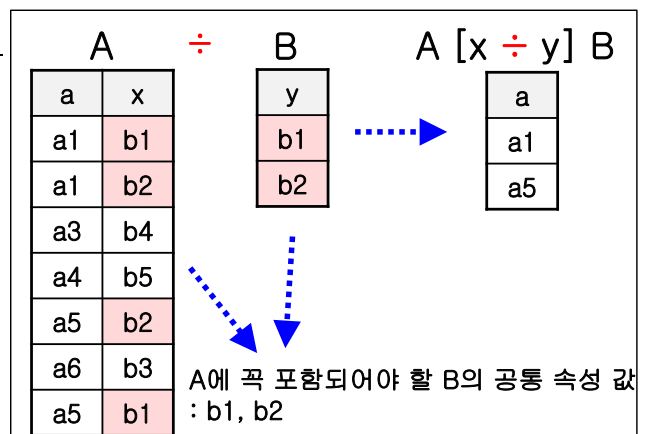
테스트 케이스	X > K	RESULT > 0	경로
2	T	F	1 → 2 → 3 → 4 → 5 → 6 → 7
	F	T	1 → 2 → 4 → 5 → 6 → 1

14. 관계대수: 두 개의 릴레이션 A와 B가 있을 때 릴레이션 B의 모든 조건을 만족하는 경우의 튜플들만 릴레이션 A에서 분리하여 프로젝션을 하는 연산자 기호

- 답안: ÷
- 출제: 응용 SW 기초 기술 활용 4강

* 프로젝션(Projection, π)

: 선택한 릴레이션에서 원하는 속성을 떼어내어 새로운 릴레이션으로 만든다.



[정보처리기사 실기 - 2020년 3회 기출]

15. SQL 작성: [AS](#), 집계함수(MIN, MAX, AVG), GROUP BY, HAVING

<성적> 테이블에서 과목별 점수의 평균이 90점 이상인 '과목이름', '최소점수', '최대점수' 검색

<처리조건>

- WHERE문은 사용하지 않는다.
- 집계함수(Aggregation Function)를 사용하여 구성한다.
- '최소점수', '최대점수'는 별칭(Alias)을 위한 AS문을 사용한다.
- 대/소문자를 구분하지 않고, 세미콜론(;)은 생략 가능하다.

<성적>

학번	과목이름	점수	학점
0001	정보	88	B
0002	컴활	65	C
0003	정보	90	A
0004	정보	97	A
0005	컴활	80	B

<결과>

과목이름	최소점수	최대점수
정보	88	97

- 답안: select 과목이름, min(점수) as 최소점수, max(점수) as 최대점수 from 성적 group by 과목이름 having avg(점수) >= 90;
- 출제: SQL 응용 2강

[정보처리기사 실기 - 2020년 3회 기출]

16. 내부 구조를 보지 않고 주로 구현된 기능을 테스트하는 테스트 기법으로, 동치 분할 테스트, 경계값 분석 테스트 등을 이용하는 테스트 기법을 쓰시오.

- 답안: [블랙박스 테스트\(Black Box Test\)](#)
- 출제: 애플리케이션 테스트 관리 1강

17. 다음 C언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오. (단, 출력문의 출력 서식을 준수하시오.)

```
#include <stdio.h>

int r1() {
    return 4;
}
int r10() {
    return (30+r1());
}
int r100() {
    return (200+r10());
}
int main() {
    printf("%d", r100());
    return 0;
}
```

실행 결과 추적:

- r1() returns 4.
- r10() calculates 30 + 4 = 34.
- r100() calculates 200 + 34 = 234.
- main() prints 234.

- 답안: 234
- 출제: 프로그래밍 언어 활용 3강

[정보처리기사 실기 - 2020년 3회 기출]

18. JAVA: 짝수 합계

```
public class Gisafirst {
    public static void main(String[] args) {
        int i = 0;
        int sum = 0;
        while (i < 10) {
            i++;
            if(i % 2 == 1)
                continue;
            sum += i;
        }
        System.out.print(sum);
    }
}
```

i	sum	i < 10	i % 2 == 1
0	0	T	
1		T	1%2==1 → T
2	2		2%2==1 → F
:	:	:	:

→ i가 10일 때 while문 종료

- 답안: 30
- 출제: 프로그래밍 언어 활용 1강

19. <학생> 테이블에서 '이름'이 "민수"인 튜플을 삭제하고자 한다. 다음 <처리조건>을 참고하여 SQL문을 작성하시오.

<처리조건>

- 명령문 마지막의 세미콜론(;)은 생략이 가능하다.
- 인용 부호가 필요한 경우 작은 따옴표(' ')를 사용한다.

- 답안: DELETE FROM 학생 WHERE 이름 = '민수' ;
- 출제: SQL 응용 2강

[정보처리기사 실기 - 2020년 3회 기출]

20. JAVA: 추상 클래스

- 코드: 다음 페이지
- 답안: Vehicle name:Spark

* 추상 클래스(Abstract Class) → 미완성 설계도

: 클래스들의 공통되는 필드와 메소드를 정의한 구체적이지 않은 클래스.

- 상위 클래스에는 메소드의 시그니처만 정의해 놓고 그 메소드의 실제 동작 방법은 이 메소드를 상속 받은 하위 클래스의 책임으로 위임한다.

```
[형식]  abstract class 클래스명 {
        ...
    }
```

- 특징
 - 1) 추상 클래스 객체는 생성할 수 없다. → 실체성이 없고 구체적이지 않으므로
 - 2) 추상 클래스와 실제 클래스는 상속 관계(extends)를 가진다.

추상

```

abstract class Vehicle {
    String name;
    abstract public String getName(String val);
    ⑤ public String getName() {
        ⑥ return "Vehicle name:" + name;
    }
}

class Car extends Vehicle {
    String name;
    ② public Car(String val) {
        ③ name = super.name = val;
    }
    public String getName(String val) {
        return "Car name:" + val;
    }
    public String getName(byte val[]) {
        return "Car name:" + val;
    }
}

public class Gisafirst {
    public static void main(String args[]) {
        ① Vehicle obj = new Car("Spark");
        ④ System.out.print(obj.getName());
    }
}

```

하위 클래스
에서 재정의

[실행 결과] Vehicle name:Spark

[실행 순서]

① Vehicle obj = new Car("Spark");

: 하위 클래스 생성자로 객체 변수를 생성하고 자료형은 상위 클래스로 지정 → 오버라이딩(재정의) 하기 위해

② public Car(String val)

: new 연산자를 통해서 객체 생성 시 초기화 작업을 위한 생성자 함수 실행

③ name = super.name = val;



④ System.out.print(obj.getName())

: obj 객체의 getName() 함수 호출 → 오버로딩: 같은 이름의 함수를 여러 개 정의하고, 매개변수의 유형과 개수를 다르게 함

⑤ public String getName()

: 호출한 함수에 매개변수가 없으므로, 매개변수가 없는 Vehicle 의 getName() 함수 호출

⑥ return "Vehicle name:" + name;

: Vehicle name:Spark 반환 후 출력

[정보처리기사 실기 - 2020년 3회 기출]

* 기출 분석 및 학습 방법

1. 학술형 문제가 다수 출제됨 (헝가리안 표기법, 스키마, 직관성, 생성자, 형상 통제, 리팩토링)

- 학술형에 약할 경우 SQL, 프로그래밍 언어 활용 파트에서 점수를 획득하여야 합니다.

- 아는 것과 펜으로 생각을 적는 것은 다릅니다. 용어집에서 밑줄 표시해 드린 키워드 중심으로 작성하면 됩니다.

2. 필기 내용 출제 (리팩토링, 블랙박스 테스트, 관계 대수 기호 등 용어 문제)

- 필기 강의 복습

3. 용어(단답형) 문제 출제

- 퀴즐렛, 용어집 활용

4. 프로그래밍 언어, SQL 작성 연습 → 만점

- 간단한 문장, 복잡한 문장 모두 작성

5. 이해 중심의 학습 (분기 커버리지)

- 정의 및 개념이 응용되어 출제 → 이해 중심 학습

6. 실전 모의고사, 기출/예상 문제집 자료 참고