

<정답> 30, 55, 15, 32, 44, 60, null

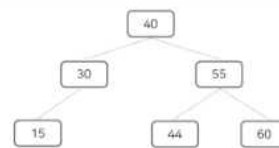
A1

이진 탐색 트리(binary search tree)는 이진 트리 기반의 탐색을 위한 자료 구조

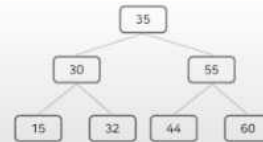
이진 탐색 트리의 조건

- 1) 모든 노드의 키는 유일하다
- 2) 왼쪽 서브 트리의 키들은 루트의 키보다 작다
- 3) 오른쪽 서브 트리의 키들은 루트의 키보다 크다.
- 4) 왼쪽과 오른쪽 서브 트리도 이진 탐색 트리이다.

1. 35값의 원소를 제거
35보다 큰 다음 값에 해당하는 값 40을 35의 위치에
대체함으로써 이진트리의 속성을 유지함



2. 32값을 추가



A2

다음 클래스 다이어그램은 Book에 대한 추상클래스이다. 다음을 참고하여 아래의 물음에 답하시오.

- 1) [보기]에서 setTitle() 메소드의 접근제한 속성에 대하여 설명하시오(10점).
- 2) [보기]의 클래스 다이어그램을 JAVA 언어 코드로 작성하시오(30점).

[정답]

1) setTitle 메소드의 "+" 표시는 public 접근제한 속성으로 어떤 클래스의 객체에서든 접근 가능하다.

2)

```

public abstract class Book{
    private String title;
    private String author;

    public abstract void setTitle( String title);
    public abstract void setAuthor( String author);
}
  
```

A2

다음 클래스 다이어그램은 Book에 대한 추상클래스이다. 다음을 참고하여 아래의 물음에 답하시오.

- 1) [보기]에서 setTitle() 메소드의 접근제한 속성에 대하여 설명하시오(10점).
- 2) [보기]의 클래스 다이어그램을 JAVA 언어 코드로 작성하시오(30점).

[정답]

1) UML 표기법의 접근제한자(Access Modifier)

접근제한자	표시	설명
public	+	어떤 클래스의 객체에서든 접근이 가능하다.
private	-	해당 클래스에서 생성된 객체들만 접근 가능하다.
protected	#	해당 클래스와 동일 패키지에 있거나 상속관계에 있는 하위 클래스의 객체들만 접근 가능하다.

2) java코드

- 추상클래스 이므로 **abstract 키워드를 사용하여** 클래스를 정의하여야 한다.
- {abstract} 프로퍼티가 있으므로 메소드에도 abstract 키워드를 사용하여야 한다.
- "-와 "+"에 따른 접근 제한자(Access Modifier) 사용을 주의하여야 한다.

3. 소프트웨어 개발 서술형 해설

Q3

<정답>

- 1) DFD (Data Flow Diagram)
- 2) Process(원형), DataFlow(화살표), Data Store(데이터 저장소), External Entity(DFD 외부범위)

A3

1. DFD 개념

- 데이터의 프로세스 흐름 표현
- 소프트웨어의 설계 및 분석에서 유용하게 사용

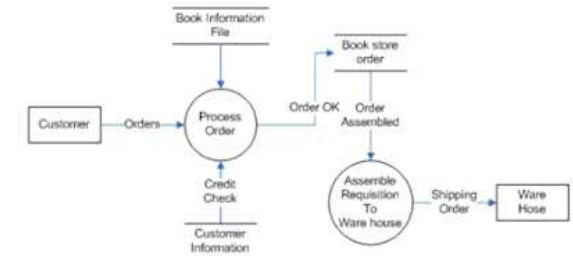
2. 구성요소

1) Process

- 입력되는 데이터를 원하는 데이터로 변환하는 과정
- 스스로 데이터를 만들어 낼 수는 없고 반드시 입력은 데이터가 있어야 함.
- 프로세스는 항상 새로운 가치를 부가해야 함, 원형으로 표시

3) Data Store

- 데이터가 저장된 장소, 평행선 두개로 표시



2) Data Flow

- DFD에서 구성요소간 인터페이스를 표현
- 대부분 프로세스들을 연결하지만 Data Store에서부터의 데이터 흐름을 나타내기도 함.
- 명칭과 함께 화살표로 표시

4) External Entity

- 프로세스 처리 과정의 시작과 끝을 표현
- DFD 범위 밖에 사각형 형태로 표시

4. 소프트웨어 개발 객관식 해설

Q4

<정답> 2번

A4

1. 결합도

- 각 모듈 간에 변경 발생시 영향을 미치는 정도로 낮을 수록 좋다

자료	스텝	제어	외부	공통	내용
모듈간 매개 변수로 통신. 매개변수를 최소화 하는 것이 좋음	모듈 간 매개변수로 구조체를 지정 구조체를 변수화	제어문을 이용, 타 모듈의 내부 제어 변경이 발생되면 caller, callee 모두 변경. 다형성 적용으로 해결	Comfile 등 모듈 간 공통적인 SW외부 환경과 정의 공유 명령시 호출되는 g함수를 매핑하여 해결 (lookup table)	전역변수 등 모듈간 동일 자료영역 공통 조회 전역변수를 캡슐화하여 모듈에 제공 (싱글톤디자인패턴)	다른 모듈의 내부 데이터를 직접 수정하는 경우 변수를 캡슐화하여 방지
Void main() {Local (int x, int y);}	Struct 좌표 (int x, int y;); Void main() {local(좌표 xy); }	Void main(){local(1);} Void local(intisExec) {if(isExec){ } else { } }	Void main(){ #include config dat} Void local(){#include config dat}	Static int A; Void main(){ A= 1; } Void local(){ A= 2; }	Void main(){ GO TO LOCAL; } Void local (int x, int y){LOCAL... }

자료 결합도가 가장 낮아 좋다.

A4

2. 응집도

- 각 모듈이 하나의 기능을 중심으로 잘 뭉쳐져 있는지를 판단하는 것으로 높을수록 좋다

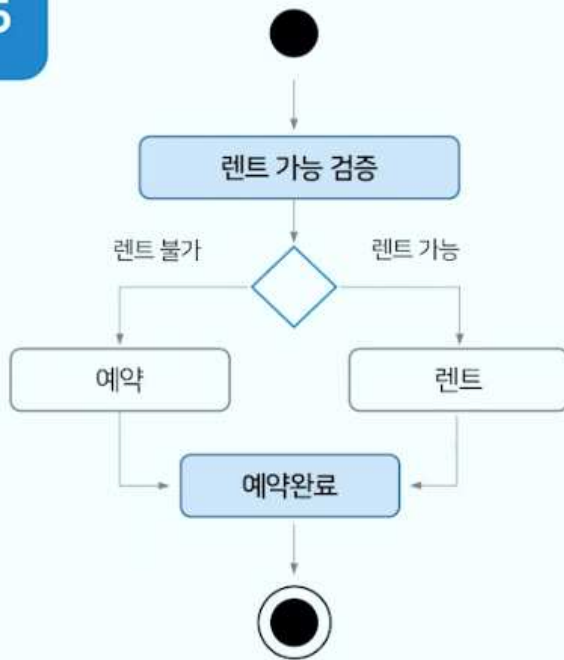
기능	순차	통신	절차	일시	논리	우연
하나의 모듈이 하나의 기능만 수행	모듈내의 동일한 요소에 값을 출력하고 다시 입력 값으로 사용	모듈 내 요소들이 동일한 자료를 이용하여 서로 다른 기능을 수행	모듈 내 요소간에 실행되어야 하는 순서 존재	변수 초기화 처럼 1회 실행되는 요소, 요소들 동시 수행	논리적으로 유사 기능을 수행하나 밀접한 관련은 없음	모듈내 요소들의 연관관계가 거의없음
NA	A = update(); Deleta(A)	DB에 저장된 nmae Print(name); Select(name)	Init() Lisiten() Request()	Init() Memset()	Switch(1) Case: 1 Case: 2 Case : 3	

기능 응집도가 가장 높고 우연 응집도가 가장 낮다. (기능 응집도가 좋은 것)

5. 소프트웨어 개발 수행형 해설

Q5

A5



동기화 막대는 위 그림에 없음

6. 소프트웨어 개발 객관식 해설

Q6

[정답] 4번

[해설]

[보기]를 실행하면 RecursionError: maximum recursion depth exceeded in comparison 에러가 발생한다.(Python에서는 기본 설정값을 변경하지 않으면 recursion 횟수를 1000번으로 제한하고 있으며, 이 소스코드는 998까지 계산 가능)

7. 소프트웨어 개발 객관식 해설

Q7

<정답> 3번

A7

[보기]

A	B	A or B 결과
T	T	T
F	F	F

(T:true, F:false)

1. 조건 커버리지(Condition Coverage):

- 모든 내부 조건이 True, False 각각 1번 수행
- 조건A(T,F), 조건B(T,F)로 만족

2. 결정 커버리지(Decision Coverage):

- 전체 결과가 True, False 각각 1번 수행

3. 조건/결정 커버리지(Condition/Decision Coverage):

- 내부 조건은 조건 커버리지, 결과는 결정[분기] 커버리지

A7

4. 변경 조건/결정 커버리지(Modified Condition/Decision Coverage):

- 각 개별 조건식이 다른 개별 조건식에 영향을 받지 않음
- 전체 결정(분기)에 독립적으로 영향을 줌

A	B	A or B 결과
T	F	T
F	T	T
F	F	F

5. 다중 조건/결정 커버리지 : 모든 가능한 논리적인 조합 :

A	B	A or B 결과
T	T	T
T	F	T
F	T	T
F	F	F

- 4번이 조금 어려울 수 있는데 보면 둘다 T, T인 것이 빠져있다. 독립적으로 영향을 주지 않기 때문이다. A가 False로 바뀌었을 때 A or B에 영향을 주지 않기 때문에 해당 케이스는 제외되어 있다. (B와 관련없이 A가 true 또는 false로 바뀌었을 때 A or B의 결과가 영향을 준다는 것이다.)