



PATH

정보처리기사 실기

핵심 요약

**이 파일은 인쇄나 편집이 제한된 파일입니다.
영리목적으로 사용할 수 없고
개인적인 목적으로만 사용이 가능합니다.**

**컴퓨터자격증은PATH는 정보처리 이외에도
많은 자격증에 대한 안내를 계속할 예정입니다.
앞으로도 많은 관심 부탁드립니다.**



알고리즘

알고리즘의 규칙

수의 변수는 어디선가 증감 한다 $\text{ex} \rangle i = i + 1, i = i - 1$

합의 초기값은 별도의 설정이 없을때 0 으로 선언 $\text{ex} \rangle \text{sum} = 0$

합의 공식 : 합 = 합 + 수 $\text{ex} \rangle \text{sum} = \text{sum} + i$

평균의 공식 : 평균 = 합 / 개수 $\text{ex} \rangle \text{avg} = \text{sum} / \text{cnt}$

부호의 변경(+ \leftrightarrow -) $\text{ex} \rangle s = s * (-1)$

알고리즘의 규칙

배열의 시작값은 대부분 0으로 시작한다. ex) $ar[5] = ar[0] \ ar[1] \dots$

최대값의 초기값은 가장 작은수 ex) $max = 0$

최대값과 수를 비교하여 수가 크면 최대값 교체 ex) $max < i \rightarrow max = i$

최소값의 초기값은 가장 큰수 ex) $min = 100$

최소값과 수를 비교하여 수가 작으면 최소값 교체 ex) $min > i \rightarrow min = i$

알고리즘의 규칙

몫 은 / 로 표시 ex) $5 / 2 = 2$, $4 / 2 = 2$

나머지는 % 또는 mod 로 표시 ex) $5 \bmod 2 = 1$, $5 \% 2 = 1$

▶프로그래밍에서는 %만 사용

나머지를 활용하여 배수 표현 ex) 수 $\bmod 2 == 0$ 은 짝수

수 $\bmod 2 != 0$ 은 홀수

수 $\bmod 3 == 0$ 은 3의배수

순위의 초기값은 1 ex) rank = 1

각 값을 비교하여 작을 경우 순위를 증가 시킨다.

프로그래밍

산술연산자 : **+** , **-** , ***** , **/** , **%**

산술연산자 우선순위 : (괄호) → /, *, % → +, -

▶ 동일한 우선순위에서 연산은 왼쪽부터

관계연산자 : **>** **<** **>=** **<=** **==** **!=**
초과 **미만** **이상** **이하** **같다** **다르다**

▶ **!** 는 **not** 연산자로 사용되며, 논리값을 반전시켜 출력
ex> **!(10 > 5) // 결과:0(False)**

비트연산자 : 10진수 값을 2진수로 변환하여 논리 연산

```
ex> a = 5, b = 2 //           a  0101  
    c = a&b      //           b  0010  
    c = 0        // a와b를 and 연산 0000
```

증감연산자 : 값을 증가 하거나 감소하는 연산자

```
ex> 전위 : ++a, --a > 피연산자 증감후, 연산 수행  
    후위 : a++, a-- > 연산 수행후, 피연산자 증감  
    a = 5, b = 5, c = 0  
    c = ++a + b-- // a는 증가후 연산, b는 연산후 감소  
    c = 11, a = 6, b = 4
```

삼항연산자 : 조건 ? TRUE : FALSE

ex> num1 = 7, num2 = 3

result = num1 < num2 ? num1 : num2

위 조건을 만족하지 않으므로 false 인 3이 입력

시프트연산자 : \gg \ll 비트의 위치를 **방향**만큼 이동 Left Right

ex> num = 5 ; // 0000 0101

result = num \ll 2 // 0001 0100 (20)

result = 20

FOR문 : **지정한 횟수 만큼 반복** for(기준값; 조건; 증감)

```
ex> for(int i=0;i<5;i++){           // 0 부터 5미만까지 진행
    a += i;    }                     // 0 , 1, 2, 3, 4 가 누적
    printf("%d %d",a , i);           // i 값은 for문에서 4까지 진행후
출력 : 10 5                         1이 증가한 5가 저장됨
```

WHILE문 : 조건에 부합하는 동안 반복 while(조건)

ex> a=0, i=0;

```
while(i<5) {
```

```
    a += i;
```

```
    i++; }
```

```
printf("%d %d",a , i);
```

출력 : 10 5

// 5미만까지 진행

// 0 , 1, 2, 3, 4 가 누적

// 증감문 없을시 무한루프

// i 값은 반복문에서 4까지 진행후

1이 증가한 5가 저장됨

Switch case: **case의 값과 일치 하면 해당문 실행**

▶ break문 유무에 따라 실행 결과 상의

```
ex> int num = 2;  
    switch(num){  
    case 1:  
        printf("1\n");  
    case 2:  
        printf("2\n");  
    case 3:  
        printf("3\n");  
    default:  
        printf("exit\n");  
    }
```

출력 : 2
3
exit

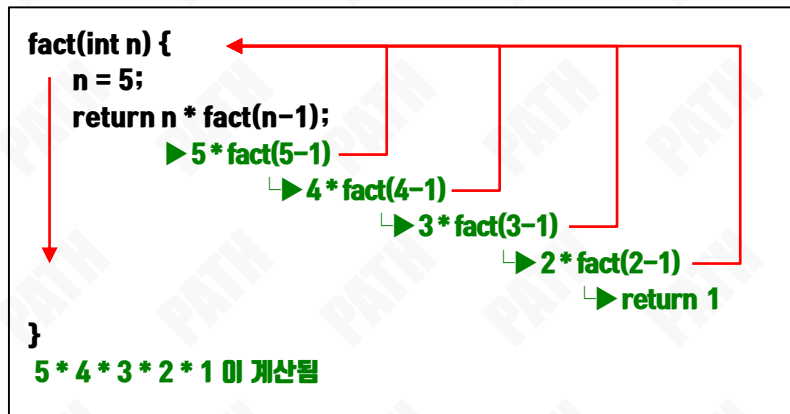
```
ex> int num = 2;  
    switch(num){  
    case 1:  
        printf("1\n"); break;  
    case 2:  
        printf("2\n"); break;  
    case 3:  
        printf("3\n"); break;  
    default:  
        printf("exit\n"); break;  
    }
```

출력 : 2

재귀함수 : 함수에서 자기 자신을 다시 호출하여 수행

```
ex) int fact(int n){  
    if(n==1)  
        return 1;  
    return n*fact(n-1);  
}  
void main(){  
    int n=5;  
    printf("%d",fact(n));  
}
```

출력 : 120



진수 변환 : 프로그래밍은 기본적으로 10진 형태

ex> 2진수 = 0b1001 ex> 진수 입,출력 : scanf, printf 에서의 진수 사용법
8진수 = 067 10진수 = %d 8진수 = %o 16진수 = %x
16진수 = 0xA2

```
ex> int num = 0b11010;  
    printf("10진수 : %d\n",num);  
    printf("8진수 : %o\n",num);  
    printf("16진수 : %x\n",num);
```

```
// 2진수 11010 입력  
// 10진수 : 26  
// 8진수 : 32  
// 16진수 : 1a
```

데이터베이스

DBA : 데이터베이스 관리자

DataBase Administrator : DBMS를 사용하여 DB 관리하는 사람 혹은 그룹

DBMS : 데이터베이스 관리 시스템

DataBase Management System : DBA가 사용하는 데이터베이스 관리 시스템

ex) MySQL : 오라클 사의 RDBMS

데이터베이스 : 자료(Data)의 모임

특징 : 실시간 접근성, 지속적인 변화, 동시 공유, 상호 참조, 데이터 논리적 독립성

※ 데이터베이스 정규화 : 불필요한 데이터 제거

데이터베이스 설계

요구조건분석 → 개념적 설계 → 논리적 설계 → 물리적 설계 → 구현

스키마(Schema) : 데이터베이스 전반적인 명세

외부스키마 : 개인의 관점 / 서브스키마

개념스키마 : 조직적 관점 / 논리스키마

내부스키마 : 시스템 프로그래머의 관점

Degree(디그리; 차수) : 속성의 개수

속성(attribute), 필드(field)

Cardinality(카디널리티; 기수) : 튜플의 개수

튜플(Tuple), 레코드(record)

트랜잭션 : 하나의 작업 수행을 위한 연산들의 집합

**ACID : 원자성(Atomicity), 일관성(Consistency),
독립성(Isolation), 지속성(Durability)**

SQL : 데이터베이스 질의어

Structured Query Language ; 정의어(DDL), 조작어(DML), 제어어(DCL)

DDL : 데이터베이스 정의(Definition) 언어

생성 : CREATE TABLE 테이블명

변경 : ALTER TABLE 테이블명 ADD 컬럼명 데이터타입

제거 : DROP TABLE 테이블명 [CASCADE/RESTRICT]

초기화 : TRUNCATE 테이블명

CREATE 컬럼명 FROM 테이블 WHERE 조건문

**CREATE VIEW 뷰이름 AS SELECT 컬럼1, 컬럼2...
FROM 테이블 WHERE 조건문**

VIEW : 유도된 가상 테이블

ALTER TABLE 테이블명 ADD 컬럼명 데이터타입

ALTER ADD : ALTER문의 열 추가

ALTER TABLE 테이블명 MODIFY 컬럼명 데이터타입

ALTER MODIFY : ALTER문의 타입 변경

ALTER TABLE 테이블명 DROP 컬럼명

ALTER DROP : ALTER문의 열 삭제

DROP TABLE 테이블명 RESTRICT

RESTRICT : 참조 시 삭제 취소

DROP TABLE 테이블명 CASCADE

CASCADE : 참조 삭제

SELECT DISTINCT 컬럼명 FROM 테이블명

DISTINCT : 중복제거

DML : 데이터베이스 조작(Manipulation) 언어

검색 : SELECT 컬럼명 FROM 테이블명

갱신 : UPDATE 테이블명 SET 컬럼명 = 수정값

삭제 : DELETE [FROM] 테이블명

삽입 : INSERT INTO 테이블명 VALUES 입력값1, 입력값2...

SELECT 컬럼명 FROM 테이블명 WHERE 조건문

WHERE : SQL 기본 조건문

SELECT 컬럼명 FROM 테이블명 ORDER BY 컬럼명 ASC

ORDER BY : 정렬 (ASC : 오름차순 / DESC : 내림차순)

**SELECT 컬럼명 FROM 테이블명 GROUP BY 컬럼명
HAVING 조건문**

GROUP BY : 그룹 (조건 작성 시 HAVING)

**SELECT 컬럼명 FROM 테이블1 LEFT JOIN 테이블2
ON 테이블1.컬럼명 = 테이블2.컬럼명**

조인 : INNER JOIN / OUTER JOIN(LEFT/RIGHT/FULL)

SELECT **작성** 순서

SELECT -> FROM -> WHERE -> GROUP BY -> HAVING -> ORDER BY

SELECT **실행** 순서

FROM -> WHERE -> GROUP BY -> HAVING -> SELECT -> ORDERBY

DCL : 데이터베이스 제어(Control) 언어

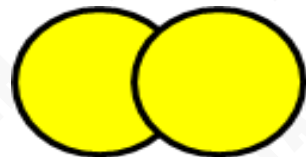
완료 : COMMIT

취소 : ROLLBACK

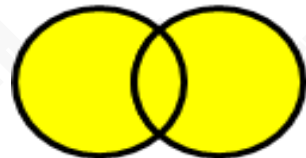
권한부여 : GRANT SELECT ON 테이블명 TO 컬럼명

권한제거 : REVOKE SELECT ON 테이블명 FROM 컬럼명

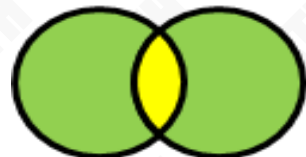
UNION : 합집합(중복제거)



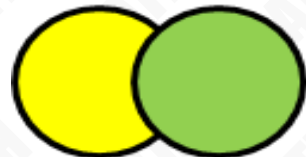
UNION ALL : 합집합(중복포함)



INTERSECT : 교집합



MINUS : 차집합



기본키 : 후보키 중 선정된 키 / **중복 불가**

후보키 : 유일성 만족 / 최소성 만족

슈퍼키 : 유일성 만족 / 최소성 불만족

대체키 : 기본키로 선택되지 못한 후보키

외래키 : 다른 테이블의 행 식별 키

데이터베이스 이상현상(Anomaly; **아노말리**)

데이터 불일치 현상

삽입이상 / 삭제이상 / 갱신이상

관계 대수 : 정보 유도 절차적 언어

관계 해석 : 정보 명시 비절차적 언어

운영체제

운영체제(Operating System)

컴퓨터 하드웨어와 사용자 간에 위치
하드웨어 및 소프트웨어의 자원 등을 관리

운영체제의 목적

처리능력(Throughput) , 반환시간(Turn around Time)
사용가능도(Availability), 신뢰도(Reliability)

운영체제의 발전 과정

일괄 -> 실시간 -> 다중프로그래밍 -> 시분할 -> 다중처리
-> 범용 시스템 -> 분산 처리

윈도우(Windows) : 선점형멀티테스킹, GUI, PNP, NTFS

마이크로소프트 개발 / GUI : 그래픽 사용자 인터페이스 지원 / PNP : Plug & Play

도스(DOS) : CLI(CUI)

CLI : 문자기반 유저 인터페이스(혹은 CUI)

유닉스(Unix) : 다중 사용자, 멀티 태스킹, CLI, 시분할

셸(Shell) : 명령어 번역기

커널(Kernel) : 운영체제의 핵심 / 자원을 통제

리눅스(Linux) : 오픈 소스 운영체제

유닉스를 기반으로 개발

PNP : Plug & Play

외부 장치 연결 시, 자동으로 해당 프로그램을 찾아 실행하는 기능

핫 스왑(Hot Swap) : [전원ON] 장치연결 ○

시스템 전원이 켜진 상태에서, 외부 장치를 안전하게 연결 및 제거하는 기능

핫 플러그(Hot Plug) : [전원OFF] 장치연결 ○

새로운 장치를 연결할 시, 시스템 전원이 꺼진 상태에서 사용하는 기능

아이노드(i-node) : 자료 구조 (Unix)

정규 파일, 디렉터리 등 파일 시스템에 관한 정보 보유

심볼릭 링크, 소프트 링크 : 바로가기 (Unix)

링크파일 삭제 시, 원본 유지(윈도우 운영체제의 바로가기와 동일)

하드 링크 : 원본과 동기화된 바로가기 (Unix)

링크파일 삭제 시, 원본 삭제

FAT12, FAT16, FAT32 파일 시스템 (windows)

연결리스트 형태의 자료 구조 / 긴 검색 시간

NTFS 파일 시스템 (windows)

FAT과 HPFS의 단점을 개선

UFS 파일 시스템 (Unix)

EXT2 : UFS에서 불필요한 구조들을 제거한 유닉스 파일 시스템

EXT3 : EXT2에서 저널링 기능을 추가한 유닉스 파일 시스템

가상화(Virtualization)

단일 호스트에서 다수의 서로 다른 운영체제를 구동 지원하는 기능

하이퍼바이저(Hypervisor)

단일 호스트에서 다수의 운영체제를 가상으로 구동 지원하는 플랫폼

윈도우 대표 단축키

Win() + D : 바탕화면 표시

M : 열려있는 창 최소화

E : 윈도우 탐색기 실행

R : 윈도우 실행창 실행

F : 찾기

X : 윈도우 시스템 관리

I : 윈도우 설정(제어판)

L : 사용자 전환 / 윈도우 잠금

P : 프로젝트 창 실행(다른 화면 표시 , 듀얼모니터 프로젝트 설정)

Pause : 시스템 구성 요소 확인

Tab : 태스크 바(Task bar)실행

Shift + S : 캡처도구 실행 (스크린샷)

윈도우 & 도스 vs 리눅스 명령어

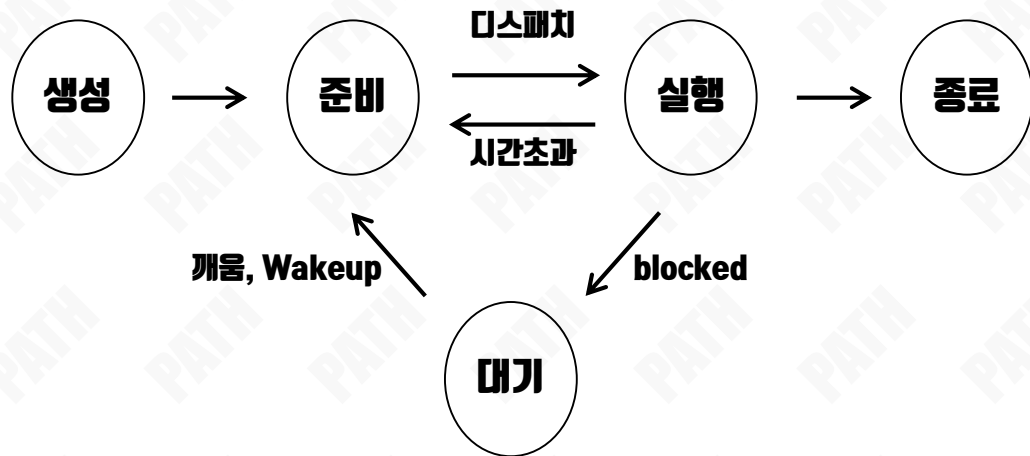
| 기능 | 윈도우 / MS-DOS | 리눅스 |
|-----------|--------------|------------|
| 경로 변경 | cd | cd |
| 목록 출력 | dir | ls |
| 구조복사 | xcopy | cp |
| 디렉토리 생성 | mkdir, md | mkdir |
| 하위파일 삭제 | del | rm |
| 속성 설정 | attrib | chmod |
| 화면 표시 | type | cat |
| 목적지 까지 경로 | tracert | traceroute |

리눅스 명령어

| 기능 | 리눅스 |
|-------------|------|
| 프로세스 종료 | kill |
| 실행중 프로세스 표시 | ps |
| 디렉토리 경로 표시 | pwd |
| 네트워크 상태 점검 | ping |
| 접속한 사용자 표시 | who |

프로세스(Process): 실행 중인 프로그램

프로세스 상태 전이



프로세스 스케줄링 : **자원들의 우선순위를 관리**

선점형 : 이미 실행중인 프로세스라도 강제로 빼앗아 선택하여 사용할 수 있음
- RR(라운드 로빈), SRT, 다단계 큐, 다단계 피드백 큐

비선점형 : 실행중인 프로세스를 빼앗아 사용할 수 없음 순서대로 실행
- FIFO(FCFS), SJF, HRN, 우선순위

네트워크

IPv4 : 32bit .(옥텟) 유니/멀티/브로드캐스트

10진수로 구성 ex) 192.182.0.32

IPv6 : 128bit :(콜론) 멀티/애니/유니캐스트

16진수로 구성 ex) fe80:0230:e8f8:7707:a5bb::a41a:1111

※ 0은 생략가능

OSI 7계층 : 네트워크 통신을 계층으로 나눈 참조모델

응용 계층 : HTTP, FTP, DNS...

표현 계층 : JPG, MPEG, AFP...

세션 계층 : NetBIOS, SSH...

전송 계층 : TCP, UDP...

네트워크 계층 : IP, RIP, ARP, ICMP...

데이터링크 계층 : 이더넷, PPP, HDLC...

물리 계층 : RS:232, RS:449...

TCP/IP 4계층 : 인터넷 정보교환 프로토콜 계층

응용 계층 : HTTP, FTP, DNS---

전송 계층 : TCP, UDP---

인터넷 계층 : IP, RARP, ARP---

네트워크 계층 : 이더넷, Token Ring, PPP---

정보처리기사 실기 핵심 요약

네트워크

OSI 7 계층

| |
|------------------|
| 응용(Application) |
| 표현(Presentation) |
| 세션(Session) |

TCP/IP 4 계층

| |
|-----------------------|
| 응용계층 (Application) |
|-----------------------|

계층별 프로토콜

| | | |
|--------|------|-----|
| HTTP | FTP | DNS |
| SMTP | SNMP | SSL |
| telnet | DHCP | SSH |

| |
|---------------|
| 전송(Transport) |
|---------------|

| |
|-----------------|
| 전송계층(Transport) |
|-----------------|

| |
|-----|
| TCP |
|-----|

| |
|-----|
| UDP |
|-----|

| |
|---------------|
| 네트워크(Network) |
|---------------|

| |
|---------------|
| 인터넷(Internet) |
|---------------|

| |
|------|
| ICMP |
|------|

| |
|------|
| IGMP |
|------|

| |
|----|
| IP |
|----|

| |
|-----|
| ARP |
|-----|

| |
|------------------|
| 데이터링크(Data Link) |
|------------------|

| |
|------------------------------|
| 네트워크 액세스 (Network Access) |
|------------------------------|

| |
|----------|
| Ethernet |
|----------|

| |
|------------|
| Token ring |
|------------|

| |
|--------------|
| 물리(Physical) |
|--------------|

| |
|-------------|
| Frame Relay |
|-------------|

| |
|-----|
| ATM |
|-----|

네트워크 단말 장치

허브 : 하나의 **대역폭 분배 장치**(스위칭허브, 더미허브) ; 물리계층

리피터 : 네트워크 전송 신호 **증폭** ; 물리계층

라우터 : 네트워크 상 **최적의 경로** 제공 ; 네트워크계층

브리지 : **같은 구조** 네트워크 **연결** ; 데이터링크계층

게이트웨이 : **다른 구조** 네트워크 **연결** ; 세션계층(전 계층 사용)

프로토콜(Protocol) 기본요소

1. 구문(Syntax) : 데이터 형식 규정
2. 의미(Semantic) : 오류 제어 정보 규정
3. 시간(Timing) : 속도 조정 규정

TELNET : 원격 통신 프로토콜 (비암호화로 위험성 ↑)

SSH : 원격 통신 프로토콜 (암호화로 위험성 ↓)

ARP : 논리주소(IP)로 물리주소(MAC) 검색

RARP : 물리주소(MAC)로 논리주소(IP) 검색

TCP : 데이터 (검수)전송 프로토콜

UDP : 데이터 (비검수)전송 프로토콜

FTP : 파일 전송 프로토콜

HTTP : 웹 데이터 전송 프로토콜

노드(Node) : 단말 접합점

단말을 이어주는 분기와의 접합점

애드혹 네트워크(Adhoc Network) : 자율 구조 네트워크

노드에 의해 자율적으로 구성된 기반 구조 없는 네트워크

애플리케이션 테스트

V-모델

소프트웨어 개발 프로세스인 **폭포수 모델의 확장된 형태**



결함(Defect)

프로그램과 명세서 간의 차이, 업무 내용 불일치
기대 결과와 실제 관찰 결과 간의 차이

결함(Defect) 심각도 : 결함이 전체 시스템에 미치는 영향의 척도

High - 프로세스를 진행할 수 없을 정도의 결함 / 시스템 다운

Medium - 시스템 흐름에 영향을 미치는 결함 / 보안 관련 오류

Low - 상황에 맞지 않는 결과 및 화면구성 결함 / 예외 메시지 미출력

결함(Defect) 검사

Fixed : 결함 수정

Assigned : 결함 할당

Open : 결함 보고(분석 전)

Closed : 수정 후 결함 미발견

Deferred : 수정 연기

Clarified : 비결함

결함(Fault) : 의도와 다른 동작 & 결과

개발자가 설계한 의도와 다른 동작과 결과를 발생시키는 것

결함(Fault) 관리 프로세스 7과정

관리계획 - 기록 - 검토 - 수정 - 재확인 - 상태추적 - 최종분석

에러(Error) : 개발 중 발생한 부정확한 결과

오류(Fault) : 프로그램 버전간의 차이로 발생

실패(Failure) : 프로그램 버전간의 실행 결과의 차이

휴먼에러(Human Error) : 원인이 인간의 실수인 에러

코드 인스펙션 : 자동화 도구 사용, 결함 발견/수정

워크스루 : 코드 품질 평가, 개선 목적 검토

살충제 패러독스 : 동일한 테스트의 비정상적인 결함 검수

테스트 케이스 : 요구사항 준수 여부 확인용 입력값

스텝 : 하위 모듈 테스트 진행

상위 모듈은 있지만 하위 모듈이 없을 때 진행하는 하향식 테스트

드라이버 : 상위 모듈 테스트 진행

하위 모듈은 있지만 상위 모듈이 없을 때 진행하는 상향식 테스트

유스케이스 : 사용자 시스템 동작 시나리오

시스템 요구사항을 알아내는 과정이자 동작을 표현한 시나리오

유스케이스 다이어그램 : 시스템 간 상호작용 표현

시스템 범위, 액터, 유스케이스, 관계

블랙박스 테스트 : 사용자 관점 / 명세 기반

균등분할 / 한계값 테스트 / 원인효과그래프테스트 / 비교테스트

화이트박스 테스트 : 개발자 관점 / 내부구조&동작 검사

기초경로 테스트 / 제어흐름 테스트 / 조건 테스트
루프 테스트 / 데이터흐름 테스트 / 분기 테스트

테스트 오라클 : 사전 정의된 참 값을 대입 비교

1. 참 오라클(True Oracle) : 기대 결과 생성 후 오류 검출
2. 샘플링 오라클(Sampling Oracle) : 특정 입력 값의 기대 결과 제공
3. 휴리스틱 오라클(Heuristic Oracle) : 확률/직관, 추정 에 의한 예상결과
4. 일관성 검사 오라클(Consistent Oracle) : App. 변경 전후의 값 동일 여부 검증

소프트웨어 아키텍처(**Software Architecture**)

소프트웨어의 골격이 되는 기본 구조

구성요소 간의 관계를 표현하는 시스템 구조체

설계 기본원리 : 모듈화 / 추상화 / 단계적 분해 / 정보은닉

JAVA : 썬 마이크로시스템즈 개발 객체 지향 프로그래밍 언어

C언어 : B언어에서 파생된 프로그래밍 언어

xUnit : 테스트 프레임 워크(JAVA : Junit, C++ : CppUnit)

JSON : XML의 단점을 보완한 JSP 기반 독립형 언어