**2과목 소프트웨어 개발**

1장 데이터 입•출력 구현

036 자료구조

- 자료를 기억장치의 공간 내에 저장하는 방법, 저장된 그룹 내에 존재하는 자료 간의 간계, 처리 방법을 연구 분석하는 것

선형 구조 : 배열 / 선형리스트(연속, 연결 리스트) / 스택 / 큐 / 데크

비선형 구조 : 트리 / 그래프

배열 – 동일한 자료형의 데이터들이 같은 크기로 나열되어 순서를 가지고 있는 집합

선형리스트 – 일정한 순서에 의해 나열된 자료구조

-> 연속리스트 – 배열과 같이 연속되는 기억장소에 저장되는 자료구조

-> 연결리스트 – 순서에 따라 노드의 포인터 부분을 이용하여 서로 연결시킨 자료구조

스택 – 리스트의 한쪽 끝으로만 자료의 삽입, 삭제 작업이 이루어지는 자료 구조 / 후입선출

큐 - 리스트의 한쪽 끝으로만 자료의 삽입 작업이 이루어지고 다른 한쪽에서 삭제 작업이 이루어짐 / 선입선출

037 트리

정점과 선분을 이용하여 사이클을 이루지 않도록 구성한 그래프의 특수한 형태

노드 / 근 노드 / 디그리 / 단말노드 / 자식 노드 / 부모 노드 / 형제 노드 / 트리의 디그리

트리의 운행법

Preorder Inorder Postorder

Root -> Left -> Right Left -> Root -> Right Left -> Right -> Root

수식의 표기법

PreFix InFix PsotFix

연산자 -> Left -> Right Left -> 연산자 -> Right Left -> Right -> 연산자

038 정렬

삽입 정렬 – 이미 순서화된 파일에 새로운 하나의 레코드를 순서에 맞게 삽입시켜 정렬

평균과 최악 모두 수행 시간 복잡도 O()

이미 순서화된 파일에, n번째 키를 앞의 n-1개의 키와 비교

쉘 정렬 – 매개 변수의 값으로 서브파일을 구성하고 서브파일을 삽입 정렬 방식으로 순서 배열하는 과정을 반복하는 정렬 방식

평균 수행 시간 복잡도 O() 최악의 수행 시간 복잡도 O()

매개변수

선택 정렬 - n개의 레코드 중에 최소값을 찾아 첫 번째 레코드 위치에 놓고 나머지 중에서 다시 최솟값을 찾아 두 번째 레코드 위치에 놓는 방식을 반복하여 정렬하는 방식

평균과 최악 모두 수행 시간 복잡도 O()

n개의 레코드 중에서 최소값을 찾아서

버블 정렬 – 인접한 두 개의 레코드 키 값을 비교하여 그 크기에 따라 레코드 위치를 서로 교환하는 정렬 방식

평균과 최악 모두 수행 시간 복잡도 O()

인접한 두 개의 레코드

퀵 정렬 – 하나의 파일을 부분적으로 나누어 가면서 정렬하는 방식

평균 수행 시간 복잡도 O) 최악의 수행 시간 복잡도 O()

하나의 파일을 부분적으로 나누어

힙 정렬 – 전이진 트리를 이용한 정렬 방식

평균과 최악 모두 수행 시간 복잡도 O)

전이진 트리

2-Way 합병 정렬 - 이미 정렬되어 있는 두 개의 파일을 한 개의 파일로 합병하는 정렬방식

평균과 최악 모두 수행 시간 복잡도 O)

이미 정렬된 두 개의 파일을 한 개의 파일로

기수 정렬 – 큐를 이용하여 자릿수별로 정렬하는 방식

평균과 최악 모두 수행 시간 복잡도 O(dn)

버킷

039 검색 – 이분 검색 / 해싱

이분 검색 – 전체 파일을 두 개의 서브파일로 분리해가며 key 레코드를 검색하는 방식

찾고자 하는 key 값을 파일의 중간 레코드 key 값과 비교하면서 검색

중간 레코드 번호 M =

해싱 – 해시 테이블이라는 기억공간을 할당하고 해시함수를 이용하여 레코드 키에 대한 해시 테이블 내의 홈 주소를 계산한 후 주어진 레코드를 해당 기억장소에 저장하거나 검색 작업을 수행하는 방식

해시 테이블 – 버킷 / 슬롯 / collision / Synonym / Overflow

해싱함수 – 제산법 / 제곱법 / 폴딩법 / 기수 변환법 / 대수적 코딩법 / 숫자 분석법 / 무작위법

040 데이터베이스 개요

데이터 저장소 – sw 개발 과정에서 다루어야 할 데이터들을 논리적인 구조로 조직화하거나 물리적인 공간에 구축한 것을 의미

데이터베이스 – 통합된 데이터 / 저장된 데이터 / 운영 데이터 / 공용 데이터

DBMS – 사용자와 데이터베이스 사이에서 사용자의 요구에 따라 정보를 생성해주고 관리해주는 소프트웨어

기능 – 정의 / 조작 / 제어

스키마 – 데이터베이스의 구조와 제약 조건에 관한 전반적인 명세를 기술한 메타데이터의 집합

사용자 관점 - 외부 스키마 / 개념 스키마 / 내부 스키마

041 데이터 입•출력

데이터 입•출력 – 소프트웨어 기능 구현을 위해 데이터베이스에 데이터를 입력하거나 데이터베이스의 데이터를 출력하는 작업을 의미

sql을 사용

데이터 접속 – 개발 코드 내에 SQL 코드를 삽입하거나 객체와 데이터를 연결하는 것

SQL – 국제표준 데이터베이스 언어

데이터 정의어(DDL) / 데이터 조작어(DML) / 데이터 제어어(DCL)

데이터 접속 – 소프트웨어의 기능 구현을 위해 프로그래밍 코드와 데이터베이스의 데이터를 연결하는 것

SQL Mapping / ORM

트랜잭션 – 데이터베이스의 상태를 변환시키는 논리적 기능을 수행하기 위한 작업의 단위

COMMIT / ROLBACK / SAVEPOINT

042 절차형 SQL

프로그래밍 언어와 같이 연속적인 실행이나 분기, 반복 등의 제어가 가능한 SQL

데이터베이스 전용의 간단한 프로그래밍이라고 할 수 있음

프로시저 – 특정기능을 수행하는 일종의 트랜잭션 언어, 호출을 통해 실행되며 저장해 놓은 SQL 작업 수행

트리거 – 데이터베이스 시스템에서 데이터 입력, 갱신, 삭제 등의 이벤트가 발생할 때마다 관련 작업이 자동으로 수행

사용자 정의 함수 – 프로시저와 유사하게 SQL을 사용하여 일련의 작업을 연속적으로 처리, 종료시 결과를 단일값으로 반환

디버깅을 통해 기능의 적합성 여부를 검증하고 실행을 통해 결과를 확인하는 테스트 과정 수행

쿼리 성능 최적화 – 데이터 입•출력 애플리케이션의 성능 향상을 위해 SQL 코드를 최적화하는 것

문제정리

자료 구조 분류

선형 구조 : 배열 , 스택, 큐, 테크, 선형 리스트

비선형 구조 : 트리, 그래프

PUSH – 스택에 자료 입력하는 명령

POP – 스택에서 자료를 출력하는 명령

오버플로 – 큐가 꽉 채워져 있는 상태로 자료를 삽입 할 수 없는 상태

언더플로 – 자료가 없어서 자료를 제거할 수 없는 상태

선택 정렬 - 1단계를 마치면 가장 작은 값이 맨 앞으로 오고 2단계를 마치면 두 번째로 작은 값이 두 번째에 옴

버블 정렬 - 1단계를 마치면 가장 큰 값이 맨 뒤로 가고 2단계를 마치면 두 번째로 큰 값이 뒤에서 두 번째에 위치함

삽입 정렬 - 1단계를 마치면 첫 번째와 두 번째 값만 비교하여 교환

선형 검색 – 정렬 되어 있지 않은 파일에서 순차적으로 검색하는 방식, KEY값을 이용하여 검색하는 방법

통합된 데이터 – 자료의 중복을 배제한 데이터의 모임

저장된 데이터 – 컴퓨터가 접근할 수 있는 저장 매체에 저장된 자료

운영 데이터 – 조직의 고유한 업무를 수행하는 데 존재 가치가 확실하고 없어서는 안 될 반드시 필요한 자료

공용 데이터 – 응용 시스템들이 공동으로 소유하고 유지하는 자료

깊이 우선 탐색 – 정점에서 자식 노드 방향으로 운행하면서 형제 노드와 자식 노드가 있을 때 자식 노드를 우선 탐색하는 기법 모든 노드를 한번씩 방문함

TCL : COMMIT – 변경 내용을 반영 / ROLLBACK – 이전 상태를 되돌림 / SAVEPOINT – 중간 저장점을 지정

키워드 : 1. 배열 2. 스택 3. 그래프 4. 트리 5. 정렬 6. 데이터베이스 7. DBMS 8. 스키마 9. SQL 10. 트랜잭션

2장 통합 구현

043 단위 모듈 구현

단위 모듈 – 소프트웨어 구현에 필요한 여러 동작 중 한 가지 동작을 수행하는 기능을 모듈로 구현한 것

단위 기능 명세서 작성 -> 입•출력 기능 구현 -> 알고리즘 구현

단위 기능 명세서 – 설계서를 작성하는 기능 및 코드 명세서나 설계 지침과 같이 단위 기능을 명세화한 문서들을 의미

입•출력 기능 구현 – 단위 기능 명세서에 정의한 데이터 형식에 따라 입•출력 기능을 위한 알고리즘 및 데이터를 구현

알고리즘 구현 – 입•출력 데이터를 바탕으로 단위 기능별 요구 사항들을 구현 가능한 언어를 이용하여 모듈로 구현

디바이스 드라이버 모듈 / 네트워크 모듈 / 파일 모듈 / 메모리 모듈 / 프로세스 모듈

044 단위 모듈 테스트

- 프로젝트의 단위 기능을 구현하는 모듈이 정해진 기능을 정확히 수행하는 지 검증하는 것

화이트 박스 테스트 / 블랙박스 테스트

테스트 케이스 – 구현된 소프트웨어가 사용자의 요구사항을 정확하게 준수했는지를 확인하기 위해 설계된 입력값, 실행 조건, 기대 결과로 구성된 테스트 항목에 대한 명세서

식별자 / 테스트 항목 / 입력 명세 / 출력 명세 / 환경 설정 / 특수 절차 요구 / 의존성 기술

테스트 프로세스 – 테스트를 위해 수행하는 모든 작업들이 테스트의 목적과 조건을 달성할 수 있도록 도와주는 과정

테스트 프로세스 5단계

1. 계획 및 제어 단계

2. 분석 및 설계 단계

3. 구현 및 실현 단계

4. 평가 단계

5. 완료 단계

045 개발 지원 도구

통합 개발 환경 – 모든 작업을 하나의 프로그램에서 처리할 수 있도록 제공하는 소프트웨어적인 개발 활동

코딩 / 컴파일 / 디버깅 / 배포

빌드 – 소스 코드 파일들을 컴퓨터에서 실행할 수 있는 제품 소프트웨어로 변환하는 과정 또는 결과물

Ant / Maven / Gradle

협업 도구 – 개발에 참여하는 사람들이 서로 다른 작업 환경에서 원활히 프로젝트를 수행할 수 있도록 도와주는 도구

프로젝트 및 일정 관리 / 정보 공유 및 커뮤니케이션 / 디자인

키워드 : 1. 단위 기능 명세서 2. IPC 3. 테스트 케이스 4. 단위 모듈 테스트 5. 테스트 프로세스 6. IDE 7. 빌드 도구 8. 단위 모듈 9. Ant 10. Maven