1 프로그래밍 언어

01 프로그래밍 언어

1 프로그래밍 언어(Programming Language : PL)의 정의와 특성

- 주어진 어떤 문제를 해결하기 위해 인간과 컴퓨터 사이에서 의사소통을 가능하게 하는 인공적인 언어를 말함

- 프로그래밍 언어를 공부해야 하는 이유
  - 효율적인 알고리즘을 개발할 수 있는 능력의 향상
  - 현재 사용하는 프로그래밍 언어의 능력을 향상
  - 주어진 과제를 해결하는 최적의 언어를 선택
  - 새로운 언어를 쉽게 배울 수 있음

01 프로그래밍 언어

2 프로그래밍 언어의 특성

- 간결성(Simplicity)
  - 사람이 프로그램을 쉽게 이해하고, 읽을 수 있도록 간결하게 표현할 수 있는 특성

- 직교성(Orthogonality)
  - 언어의 각 구성 요소가 상호 독립적이면서도 어떤 환경에서도 그 구성 요소가 같은 의미로 사용된다는 의미를 내포

01 프로그래밍 언어

2 프로그래밍 언어의 특성

- 가독성(Readability)
  - 사람이 이해 하기 쉽도록 작성된 프로그램이나 프로그래밍 언어의 문법, 주석 등이 가독성의 향상에 도움이 됨

- 정확성(Preciseness)
  - 잘 정의된 문법은 정확성을 보장함, 각 언어의 문법은 대부분이 세계 표준으로 확정

- 기계 독립성(Machine independence)
  - 서로 다른 컴퓨터 상에서 항상 같은 결과를 요구

01 프로그래밍 언어

3 프로그래밍 언어의 종류

- 저급 언어
  - 기계어와 어셈블리 언어를 의미
  - 하드웨어에 관련된 직접제어 가능
  - 프로그램 작성시 상당한 지식과 노력이 필요

01 프로그래밍 언어

3 프로그래밍 언어의 종류

- 고급 언어
  - 하드웨어에 관련된 지식 없이도 프로그램 작성 가능
  - 사용자의 명령을 컴파일러가 해석, 기계어보다 낮은 효율성
  - 일상 적인 언어, 기호 등을 그대로 이용
  - 기억장소를 임의의 기호(Symbol)에 저장하여 사용
  - 하나의 명령으로 다수의 동작 가능
    - 예 : A = B + C * D

01 프로그래밍 언어

4 컴파일러 언어와 인터프리터 언어

	컴파일러 언어	인터프리터 언어
번역 방식	프로그램 전체 번역	행(명령어) 단위 번역
장점	빠른 실행시간	- 유연성 증가 및 효율적인 - 메모리 사용
단점	- 낮은 유연성 - 메모리 낭비	늦은 실행시간

01 프로그래밍 언어

5 주요 프로그래밍 언어

FORTRAN

- 엔지니어, 수학, 과학 등을 위한 수식 계산에 강한 2세대 언어
- 1966년 ANSI(American National Standard Institute)에 의해 FORTAN IV로 표준화
- 1977년 FORTRAN 77로 버전 업

01 프로그래밍 언어

5 주요 프로그래밍 언어

FORTRAN

- FORTRAN의 특징
  - 최초의 고급 언어 중 하나,
    다른 언어의 설계에 많은 모델이 됨
  - 매우 단순하고 간결하며, 수치와 계산에 강함
  - 실행 시 자료의 크기가 고정, 동적 배열이나
    재귀 호출 등은 지원하지 않음

01 프로그래밍 언어

5 주요 프로그래밍 언어

COBOL(Common Business Oriented Language)

- 주된 목적은 사무 처리에 적합하도록 설계됨
- 1960년 COBOL- 60의 최초 버전 발표
- 1968년 ANSI 표준 승인
- 1974년 최종 버전인 ANSI COBOL 발표

01 프로그래밍 언어

5 주요 프로그래밍 언어

COBOL(Common Business Oriented Language)

- COBOL의 특징
  - 컴퓨터와 독립적으로 설계
  - 사무처리를 목적으로 설계되어, 파일 처리에서 강점을 보임
  - 일상적인 영어 문장 구조로 쉬운 가독성을 보임
  - 자연어(영어) 문장 구조는 프로그램의 커지는 결과를 초래, 효율성이 떨어짐

01 프로그래밍 언어

5 주요 프로그래밍 언어

Pascal

- 스위스에 니클라우스 워스 교수에 의해 1971년 탄생
- 구조적 프로그래밍과 알고리즘 학습에 적합
- 1990년대 초반까지 대부분의 컴퓨터 관련 교재로 채택
- 파스칼 컴파일러로 Borland의 '터보 파스칼'이 유명

01 프로그래밍 언어

5 주요 프로그래밍 언어

Pascal

- Pascal의 특징
  - 교육용으로 적합, 알고리즘, 프로그램의 연습에 알맞은 문법
  - 구조적인 프로그램의 작성 가능
  - 컴파일러의 효율성이 좋고, 컴파일러를 만들기가 쉬움
  - 객체지향 등의 새로운 개념이나 기술을 채택하여 새로운 언어로 발전

01 프로그래밍 언어

5 주요 프로그래밍 언어

C 언어

- 1972년 데니스 리치가 설계, PDP-11에서 구현
- 기존의 언어에 비해서 신뢰성, 규칙성, 간소함 등의 장점을 내포
- 저급언어의 기능 구현 가능
- 융통성과 이식성이 좋아 고급프로그래밍 언어의 개발 속도 향상에 기여

01 프로그래밍 언어

5 주요 프로그래밍 언어

C 언어

- C 언어의 특징
  - 매우 유연한 구조
  - 대부분의 운영체제에서 기본으로 지원
  - 고급언어와 저급언어 양쪽의 장점을 모두 포함
  - 모든 실행 단위가 함수로 구성 됨

01 프로그래밍 언어

5 주요 프로그래밍 언어

C++

- 객체지향 프로그래밍을 지원하기 위해 탄생
- 대다수의 응용 프로그램을 만들 때 가장 많이 사용
- 강력함과 편리함의 양쪽 장점을 골고루 내포하여 효율성을 제공

01 프로그래밍 언어

5 주요 프로그래밍 언어

C++

- **C++의 특징**
  - C의 유연성에 객체지향의 편리성을 접목
  - 기존의 C언어로 개발된
    모든 프로그램을 수정 없이 사용 가능
  - C언어에 익숙해지면 C++도 빠른 적응 가능
  - 대부분의 운영체제에서 C++을 지원

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- 제 4세대 컴퓨터가 사용된 시기에 개발된 프로그래밍 언어를 말하며 보통 4GL이라 불림

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Power Builder

- 사이베이스(Sybase)사에서 만든 객체지향 개발 언어
- 주로 데이터베이스용 응용 프로그램을 작성에 쓰임
- Visual Basic, Delphi 등과 같이 연동가능
- 그림을 그리듯이 디자인,
  최소한의 프로그래밍으로 효율성 극대화

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Power Builder의 특징

- 클라이언트/서버 응용 프로그램 개발 환경
  : 주로 데이터베이스 관련 프로그램으로 서버는 데이터베이스와 관련된 작업, 클라이언트는 사용자 입력/출력 등의 사용자 부분을 처리하는 프로그램을 말함
- 4GL의 RAD(Rapid Application Development) Tool
  : 그래픽 유저 인터페이스 환경에서 간단한 마우스 동작 만으로 빠른 개발이 가능함
- 다양한 운영체제를 지원하는 개발 환경
  : Windows, UNIX, Mac 등의 다양한 버전을 지원함

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- Power Builder의 정리
  - 특정 업무용 프로그램을 여러 운영 체제에서 사용가능
  - 편리한 데이터베이스 관련 기능으로 인기가 많음
  - 프로그램 작성 후 배포 시에 여러 가지 불편한 점이 많음
  - 데이터베이스 이외의 프로그램에서는 다른 4GL언어보다 성능이 떨어지고 불편함

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Delphi

- 볼랜드 사의 Object Pascal을 RAD Tool로 변형해 1995 출시
- 기본적인 내부 구조는 Object Pascal을 이용함
- Windows의 각종 컨트롤이나 도구 등은 VCL(Visual Component Library)이라는 개념으로 지원함

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Delphi의 특징

- Object Pascal 언어와 컴파일러를 사용함
- VCL이라는 편리한 컨트롤 제공함
- Windows와 100% 호환이 가능하여 Windows OS하 에서는 강력한 능력을 발휘

- 컨트롤, 컴포넌트 : 단순한 기능을 하는 작은 독립적 프로그램 조각으로 많은 프로그램에서 재사용이 가능, 이런 컨트롤(컴포넌트)의 재사용은 프로그램의 개발시간 단축과 품질 향상에 기여함

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- **Delphi의 근황**
  - Visual C++ 등과 함께
    대부분의 상용프로그램에서 많이 사용
  - Object Pascal의 사용과
    한글화의 부족이 문제점으로 지적

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Visual Basic

- 마이크로 소프트에서 제작
- 다른 RAD Tool과 마찬가지로 사용의 용이성과 빠른 개발성 등을 특징으로 함

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- Visual Basic의 특징
  - 가장 배우기 쉬운 Basic을 사용
  - 개발 시간은 아주 빠른 편이나 큰 프로그램을 작성하거나 객체지향적 프로그램을 하기엔 무리가 있음
  - 개발 시에는 인터프리터를 사용하고 개발이 끝나면 컴파일러를 사용하여 양쪽 모두의 장점을 가짐
  - 다른 마이크로 소프트사의 도구들을 간편하게 사용 가능

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Visual C++

- 기존의 C++에 여러 가지 Windows의 기능을 추가
- 강력한 기능으로 여러 프로그래밍 전문가들이 사용

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- **Visual C++의 특징**
  - Microsoft Foundation Class
    : MFC 란 강력하고 방대한 라이브러리를 제공
  - Windows의 모든 기능을
    가장 강력하고 자연스럽게 사용
  - Delphi 나 Visual Basic 같은 마우스로 하는
    디자인적 요소는 거의 미비한 반면 프로그래밍의
    코딩, 디버깅, 프로젝트 관리 면에서 탁월한 기능을
    발휘

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- Visual C++의 특징
  - 객체지향형 설계 도구('비주얼 모델러')를 포함해 클래스를 포함한 프로그램 설계 시에 순수하고 강력한 객체지향적 설계와 구현이 가능

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

Java

- 1994년 선(SUN)사의 가전제품을 제어하기 위한 언어 개발을 시작한 것이 그 시초
- 가전제품을 목적으로 만들어져 낮은 시스템에서도 운영이 가능하도록 설계
- 운영 체제나 중앙처리 장치에 관계없이 모든 플랫폼에서 사용 가능
- 자바와 플랫폼 사이에 자바 가상 머신이 인터페이스 역할을 수행

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

- Java 언어의 특징
  - 객체지향 언어
  - C와 C++과 비슷한 모양(문법)을 가져 사용이 쉬움
  - 객체지향의 다형성을 위해
    실행 시간에 함수 호출을 결정
  - 동적이고 편리한 메모리 관리를 지원하며
    자동으로 쓰레기 수거(Garbage collection)를 실행

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

UNIX 기반의 언어

- ProC
  - C언어의 UNIX 확장판
  - C언어를 사용한 UNIX의 데이터베이스 관리에 많이 사용
- C
  - UNIX의 가장 중심적 언어
  - UNIX 자체를 C 언어로 제작하여 UNIX 프로그래밍에 필수

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

UNIX 기반의 언어

- Shell 프로그래밍
  - 여러 가지 Shell명령어를 순차적으로 수행
- Shell
  - 쉘은 UNIX에서 대화형 사용자 인터페이스를 부르는 용어로, 사용자가 입력하는 명령어를 이해하고, 실행하는 역할을 수행

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

UNIX 기반의 언어

- Perl
  - 인터넷의 등장과 함께 인기를 누린 스크립트형 언어
  - 초보자도 배우기 쉽고 객체지향적인 특징을 지원

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

UNIX 기반의 언어

- Python
  - 가장 최근에 등장한 강력하고 배우기 쉬운 언어
  - 쉬운 문자열 제어와 객체지향적 특성을 제공
  - C언어와 연계성을 제공하며, Windows에서도 사용 가능

01 프로그래밍 언어

5 주요 프로그래밍 언어

제 4세대 언어 (Fourth-Generation Programming Language : 4GL)

UNIX 기반의 언어

- PHP
  - 인터넷에서 사용하는 대표적인 스크립트 언어
  - 컴퓨터에서 실행하지 않고 웹 서버에서 실행되는 대표적인 Server Side Script 언어

2 개발 환경

02 개발 환경

1 개발환경 준비

개발환경
- 개발에 필요한 하드웨어, 소프트웨어 및 툴을 설치
- 프로그래밍 환경, 물리적 데이터베이스 및 파일을 구축

개발환경 점검 항목
- 소프트웨어 및 툴 준비
- 프로그래밍 환경 구축
- 프로그래밍 절차를 정의 및 방법에 대한 절차 수립
- 물리적인 DB 또는 파일 구축
- 물리 데이터베이스 생성

02 개발 환경

2 좋은 프로그래밍을 위한 참고

- 개발표준
  - 프로그램에 대한 관리를 용이하게 함
  - 조직의 생산성을 높이기 위한 목적

- 좋은 프로그램이란?
  - 간결하고 명확
  - 가독성이 뛰어남
  - 유지보수에 노력과 시간이 절감

02 개발 환경

3 변수명의 부여

변수명
- 프로그램에서 변수명을 지정할 때,
  인간의 이해를 도울 수 있는 표현이 필요

변수명의 예
- Integer Area
  Integer Height
  Integer Length

  Area = Height * Length

02 개발 환경

5 주석(Comments)

- 주석
  - 프로그램 내부의 설명을 문장으로 서술한 것
  - 향후 유지보수 과정에서
    상당히 도움이 될 수 있는 내용을 제공

- 주석에 서술되어야 할 내용
  - 코딩의 이력
  - 모듈의 목적을 서술
  - 중요 변수의 사용한계
  - 인터페이스에 대한 서술

3 프로그래밍과 소스코드 인스펙션

03 프로그래밍과 소스코드 인스펙션

1 의미

- 소스코드에 내재되어 있는
  결함을 찾아서 개선하기 위한 활동

- 정적테스트의 일종이지만
  조직적인 절차와 방법에 의해 해결하려는 노력

03 프로그래밍과 소스코드 인스펙션

2 특징

- 공식적인 절차에 따라 수행
  - 조직의 활동표준에 의해 지침으로 지정
  - 품질의 수준을 보장할 수 있도록 하는 품질활동
- 긍정적인 취지에서 개발자 동료 간에 서로 배운다는 취지가 중요
- 서로를 비난해서는 곤란
- 품질을 위한 공동의 노력과 이해가 필요

03 프로그래밍과 소스코드 인스펙션

3 수행 역할

- 프로젝트 관리자
  - 자원(인력, 시간, 장소 등)을 제공
  - “소스코드 인스펙션 계획 및 결과서”에 대한 보고를 받음

- 조정자(moderator)
  - 소스코드 인스펙션을 진행하는 의장, 활동의 중심적 역할
  - 계획을 수립하고, 결과를 정리하며, 보고하기 위한 산출물을 작성

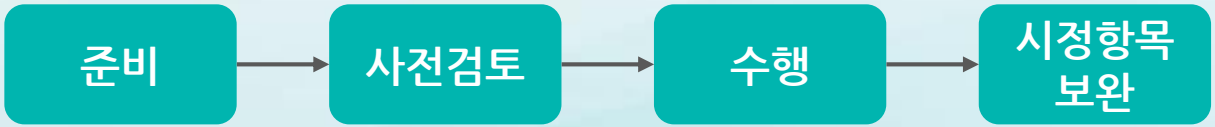
03 프로그래밍과 소스코드 인스펙션

3 수행 역할

- 개발자
  - 소스코드에 대한 정보를 제공, 문제점에 대한 답변
- 기록자
  - 회의록 작성
- 검사자
  - 소스코드를 사전에 검사 및 회의시 의견을 개진

03 프로그래밍과 소스코드 인스펙션

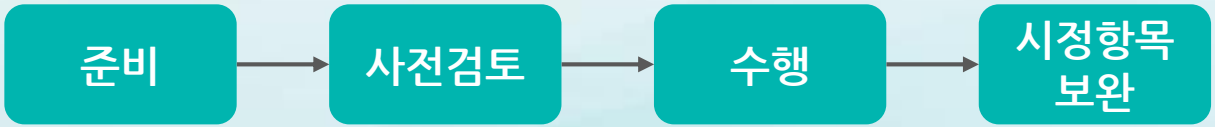
4 수행 절차



- 준비
  - 수행일정을 수립하고 공지
  - 소스코드를 검사자에게 배포
- 사전검토
  - 검사자는 대상 소스코드에 대한 개별 검토 수행, 개별 검토결과서 기록

03 프로그래밍과 소스코드 인스펙션

4 수행 절차



- 수행
  - 조정자에 의해 검토회의를 진행 및 회의록 작성

- 시정항목 보완
  - 검토회의에서 도출된 결함, 즉 시정항목을 보완하고 분석
  - 결과는 프로젝트관리자에게 보고