

4과목-프로그래밍 언어 활용

(Part 2. 프로그래밍 언어 활용-4)

프로그래밍 언어 활용 총 파트

프로그래밍 언어 활용 4과목은 총 3Part로 이루어져 있다.

1장 서버 프로그램 구현(0.69%)

2장 프로그래밍 언어 활용(44.83%)

3장 응용 SW 기초 기술 활용(54.48%)

프로그래밍 언어 활용

프로그래밍 언어 활용 Part는 17개의 섹션으로 구성되어 있다.

001 데이터 타입

002 변수

003 연산자

004 데이터 입·출력

005 제어문

006 반복문

007 배열과 문자열

008 포인터

009 Python의 기초

010 Python의 활용

011 절차적 프로그래밍 언어 012 객체지향프로그래밍 언어

013 스크립트 언어

014 선언형 언어

015 라이브러리

016 예외 처리

017 프로토타입

4. 프로그래밍 언어 활용-SEC_11(절차적 프로그래밍 언어)

1) 절차적 프로그래밍 언어의 개요

; 절차적 프로그래밍 언어는 일련의 처리 절차를 정해진 문법에 따라 순서대로 기술해 나가는 언어이다.

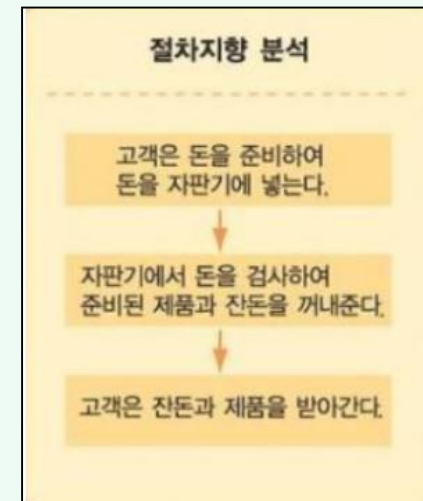
- 절차적 프로그래밍 언어는 프로그램이 실행되는 절차를 중요시 한다.
- 절차적 프로그래밍 언어는 데이터를 중심으로 프로시저를 구현하며, 프로그램 전체가 유기적으로 연결되어 있다.
- 절차적 프로그래밍 언어는 자연어에 가까운 단어와 문장으로 구성된다.
- 절차적 프로그래밍 언어는 과학 계산이나 하드웨어 제어에 주로 사용된다.

프로그램을 작성할 때 프로그램을 기능에 따라 여러 개의 단위로 분해하여 작성하면 쉽게 작성할 수 있고, 프로그램 수정, 관리가 쉬워지고, 다른 프로그램에서 재사용이 가능해진다. 이와 같이 프로그램을 여러 개의 단위로 분해한 것을 **프로시저**라 한다.

4. 프로그래밍 언어 활용-SEC_11(절차적 프로그래밍 언어)

2) 절차적 프로그래밍 언어의 장·단점

- 컴퓨터의 처리 구조와 유사하여 실행 속도가 빠르다.
- 같은 코드를 복사하지 않고 다른 위치에서 호출하여 사용할 수 있다.
- 모듈 구성이 용이하며, 구조적인 프로그래밍이 가능하다.
- 컴퓨터의 작업 처리 방식과 유사하기 때문에 객체지향 언어를 사용하는 것에 비해 더 빨리 처리되어 시간적으로 유리하다.
- 프로그램을 분석하기 어렵다.
- 유지 보수나 코드의 수정이 어렵다. -> 각 코드가 매우 유기적으로 연결되어 있어 새로운 데이터나 기능을 추가하기 어려움, 부분 고장 시 전체 고장으로 확대 되어진다.
- 실행순서가 정해져 있으므로 코드의 순서가 바뀌면 동일한 결과를 보장하기 어렵다.



4. 프로그래밍 언어 활용-SEC_11(절차적 프로그래밍 언어)

3) 절차적 프로그래밍 언어의 종류

언어	특징
C	<ul style="list-style-type: none">• 1972년 미국 벨 연구소의 데니스 리치에 의해 개발되었다.• 시스템 소프트웨어를 개발하기 편리하여 시스템 프로그래밍 언어로 널리 사용된다.• 자료의 주소를 조작할 수 있는 포인터를 제공한다.• 고급 프로그래밍 언어*이면서 저급 프로그래밍 언어*의 특징을 모두 갖췄다.• UNIX의 일부가 C 언어로 구현되었다.• 컴파일러* 방식의 언어이다.• 이식성이 좋아 컴퓨터 기종에 관계없이 프로그램을 작성할 수 있다.
ALGOL	<ul style="list-style-type: none">• 수치 계산이나 논리 연산을 위한 과학 기술 계산용 언어이다.• PASCAL과 C 언어의 모체가 되었다.
COBOL	<ul style="list-style-type: none">• 사무 처리용 언어이다.• 영어 문장 형식으로 구성되어 있어 이해와 사용이 쉽다.• 4개의 DIVISION으로 구성되어 있다.
FORTAN	<ul style="list-style-type: none">• 과학 기술 계산용 언어이다.• 수학과 공학 분야의 공식이나 수식과 같은 형태로 프로그래밍 할 수 있다.

DIVISION : 프로그램을 구성하는 골격

저급 언어와 고급 언어

저급 언어와 고급 언어의 구분은 언어가 저급이나 고급이냐를 말하는 것이 아니라 기계 친화적이나 인간 친화적이나, 즉 기계가 이해하기 쉬우면 저급 언어, 인간이 이해하기 쉬우면 고급 언어이다.

알골(ALGOL, Algorithmic Language의 준말)은 1950년대 미국에서 만들어진 포트란에 대항하여 유럽의 학자들을 중심으로 개발된 프로그래밍 언어이다.

COBOL은 1959년에 그레이스 호퍼(Grace Hopper) 즉, 프로그램 역사에서 두 번째로 유명한 여성 프로그래머로부터 처음으로 나왔다. 이후 여러 번 규격이 개정되어 현재는 객체 지향도 지원한다. 개념이 상당히 오래 묵기는 했지만 처음부터 일반인도 쓸 수 있도록 만들었기에 배우기 어렵지 않은 편이다.

포트란은 1950년대 말에 IBM의 존 배커스(John Backus) 외 6명의 전문가가 완성한 최초의 프로그래밍 언어 중 하나이다.

프로그래밍 언어 활용-SEC_11(절차적 프로그래밍 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(절차적 프로그래밍 언어)

1. C 언어에 대한 설명으로 옳지 않은 것은?

- ① 다양한 연산자를 제공한다.
- ② 이식성이 높은 언어이다.
- ③ 시스템 프로그래밍이 용이하다.
- ④ 기계어에 해당한다.

C 언어는 기계어가 아니고 기계어로 번역해야 실행할 수 있는 컴파일러 방식의 언어이다. 기계어(Machine Language)란 0, 1로 즉 2진수로 표현된 언어를 의미한다.

C 언어

1972년 미국 벨 연구소의 데니스 리치에 의해 개발되었다. 시스템 소프트웨어를 개발하기 편리하여 시스템 프로그래밍, 임베디드 프로그래밍 언어로 지금까지도 범용적으로 널리 사용된다. 데이터의 메모리 주소를 직접 조작할 수 있는 포인터를 제공한다. 고급 프로그래밍 언어이면서, 저급 프로그래밍 언어의 특징을 모두 가지고 있다.

UNIX의 일부가 C 언어로 구현되어 있다. 컴파일러 방식의 언어이며, 타입형 언어이다.

3. 다음 중 절차적 프로그래밍 언어에 대한 설명으로 틀린 것은?

- ① 컴퓨터의 처리 구조와 유사하여 실행 속도가 빠르다.
- ② 실행되는 절차를 중시한다.
- ③ 데이터를 중심으로 프로시저를 구현한다.
- ④ 상속을 통한 재사용성이 높다.

상속을 통한 재사용성이 높은 것은 객체 지향 프로그래밍 언어이다. 객체 지향 프로그래밍 언어는 다음 섹션에서 자세히 공부한다.

절차적 프로그래밍 언어는 일련의 처리 절차(순서)를 정해진 문법에 따라 순서대로 기술해 나가는 언어이다.

- 절차적 프로그래밍 언어는 프로그램이 실행되는 절차를 중요시 한다.
- 절차적 프로그래밍 언어는 데이터를 중심으로 프로시저를 구현하며, 프로그램 전체가 유기적으로 연결되어 있다.
- 절차적 프로그래밍 언어는 자연어에 가까운 단어와 문장으로 구성된다.
- 절차적 프로그래밍 언어는 과학 계산이나 하드웨어 제어에 주로 사용된다.

4. C 언어의 특징으로 옳지 않은 것은?

- ① 이식성이 뛰어나 컴퓨터 기종에 관계없이 프로그램을 작성할 수 있다.
- ② 융통성을 중요시하는 Interpreter 언어의 범주이다.

프로그래밍 언어 활용-SEC_11(절차적 프로그래밍 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(절차적 프로그래밍 언어)

5. 절차적 프로그래밍 언어의 장점이 아닌 것은?

- ① 컴퓨터의 처리 구조와 유사하여 실행 속도가 빠르다.
- ② 같은 코드를 복사하지 않고 다른 위치에서 호출하여 사용할 수 있다.
- ③ 모듈 구성이 용이하며, 구조적인 프로그래밍이 가능하다.
- ④ 실행 순서가 정해져 있으므로 코드의 순서가 바뀌어도 동일한 결과를 보장한다.

실행 순서가 정해져 있으므로 코드의 순서가 바뀌면 동일한 결과를 보장하기가 어렵다.

절차적 프로그래밍 언어의 장,단점

- 컴퓨터의 처리 구조와 유사하여 실행 속도가 빠르다.
- 같은 코드를 복사하지 않고 다른 위치에서 호출하여 사용할 수 있다.
- 모듈 구성이 용이하며, 구조적인 프로그래밍이 가능하다.
- 컴퓨터의 작업 처리 방식과 유사하기 때문에 객체 지향 언어를 사용하는 것에 비해 더 빨리 처리되어 시간적으로 유리하다.
- 프로그램을 분석하기가 어렵다.
- 유지 보수나 코드의 수정이 어렵다. -> 각 코드가 매우 유기적으로

4. 프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어)

1) 객체지향 프로그래밍 언어의 개요

- ; 객체지향 프로그래밍 언어는 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체로 만들어, 기계적인 부품들을 조립하여 제품을 만들 듯이 소프트웨어를 개발할 때도 객체들을 조립해서 프로그램을 작성할 수 있도록 한 프로그래밍 기법이다.
- 프로시저보다는 명령과 데이터로 구성된 객체를 중심으로 하는 프로그래밍 기법으로, 한 프로그램을 다른 프로그램에서 이용할 수 있도록 한다.

4. 프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어)

2) 객체지향 프로그래밍 언어의 장·단점

- 상속을 통한 재사용과 시스템의 확장이 용이하다.
- 코드의 재활용성이 높다.
- 자연적인 모델링에 의해 분석과 설계를 쉽고 효율적으로 할 수 있다.
- 사용자와 개발자 사이의 이해를 쉽게 해준다.
- 대형 프로그램의 작성이 용이하다.
- 소프트웨어 개발 및 유지보수가 용이하다.
- 프로그래밍 구현을 지원해 주는 정형화된 분석 및 설계 방법이 없다.
- 구현 시 처리 시간이 지연된다.

4. 프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어)

3) 객체지향 프로그래밍 언어의 종류

언어	특징
JAVA	<ul style="list-style-type: none">• 분산 네트워크 환경에 적용이 가능하며, 멀티스레드 기능을 제공하므로 여러 작업을 동시에 처리할 수 있다.• 운영체제 및 하드웨어에 독립적이며, 이식성이 강하다.• 캡슐화가 가능하고 재사용성 높다.
C++	<ul style="list-style-type: none">• C 언어에 객체지향 개념을 적용한 언어이다.• 모든 문제를 객체로 모델링하여 표현한다.
Smalltalk	<ul style="list-style-type: none">• 1세대 객체지향 프로그래밍 언어 중 하나로 순수한 객체지향 프로그래밍 언어이다.• 최초로 GUI 를 제공한 언어이다.

위의 나열한 언어 말고도 시뮬라67(최초), C#, Python 등 (Javascript는 객체지향으로 작성 가능)이 존재한다.

GUI(Graphical User Interface) : GUI는 아이콘이나 메뉴를 마우스로 선택하여 작업을 수행하는 그래픽 환경의 인터페이스를 의미한다.

4. 프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어)

4) 객체지향 프로그래밍 언어의 구성 요소

; 객체지향 프로그래밍 언어의 구성 요소에는 객체(Object), 클래스(Class), 메시지(Message)가 있다.

객체(Object)	<ul style="list-style-type: none">•데이터(속성)와 이를 처리하기 위한 연산(메소드)을 결합시킨 실체이다.•데이터 구조와 그 위에서 수행되는 연산들을 가지고 있는 소프트웨어 모듈이다.•속성(Attribute) : 한 클래스 내에 속한 객체들이 가지고 있는 데이터 값들을 단위별로 정의하는 것으로서 성질, 분류, 식별, 수량 또는 현재 상태 등을 표현한다.•메소드(Method) : 객체가 메시지를 받아 실행해야 할 때 구체적인 연산을 정의하는 것으로 객체의 상태를 참조하거나 변경하는 수단이 된다.
클래스(Class)	<ul style="list-style-type: none">•두 개 이상의 유사한 객체들을 묶어서 하나의 공통된 특성을 표현하는 요소이다. 즉 공통된 특성과 행위를 갖는 객체의 집합이라고 할 수 있다.•객체의 유형 또는 타입(Object Type)을 의미한다.
메시지(Message)	<ul style="list-style-type: none">•객체들 간에 상호작용을 하는데 사용되는 수단으로 객체의 메소드(동작, 연산)를 일으키는 외부의 요구 사항이다.•메시지를 받은 객체는 대응하는 연산을 수행하여 예상된 결과를 반환하게 된다.

4. 프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어)

5) 객체지향 프로그래밍 언어의 특징

; 객체지향 프로그래밍 언어의 특징에는 캡슐화, 정보 은닉, 추상화, 상속성, 다형성 등이 있다.

캡슐화 (Encapsulation)	<ul style="list-style-type: none">• 데이터(속성)와 데이터를 처리하는 함수를 하나로 묶는 것을 의미한다.• 캡슐화된 객체의 세부 내용이 외부에 은폐(정보 은닉)되어, 변경이 발생할 때 오류의 파급 효과가 적다.• 캡슐화된 객체들은 재사용이 용이하다.
정보 은닉 (Information Hiding)	캡슐화에서 가장 중요한 개념으로, 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용하는 것이다.
추상화 (Abstraction)	<ul style="list-style-type: none">• 불필요한 부분을 생략하고 객체의 속성 중 가장 중요한 것에만 중점을 두어 개략화하는 것, 즉 모델화하는 것이다.• 데이터의 공통된 성질을 추출하여 슈퍼 클래스를 선정하는 개념이다.
상속성 (Inheritance)	<ul style="list-style-type: none">• 이미 정의된 상위 클래스(부모 클래스)의 모든 속성과 연산을 하위 클래스가 물려받는 것이다.• 상속성을 이용하면 하위 클래스는 상위 클래스의 모든 속성과 연산을 자신의 클래스 내에서 다시 정의하지 않고서도 즉시 자신의 속성으로 사용할 수 있다.
다형성 (Polymorphism)	<ul style="list-style-type: none">• 메시지에 의해 객체(클래스)가 연산을 수행하게 될 때 하나의 메시지에 대해 각 객체(클래스)가 가지고 있는 고유한 방법(특성)으로 응답할 수 있는 능력을 의미한다.• 객체(클래스)들은 동일한 메소드명을 사용하며 같은 의미의 응답을 한다.

캡슐화와 정보 은닉의 장점

- 유지보수의 용이성
- 객체 이용의 용이성

캡슐화는 데이터와 기능을 하나의 단위로 묶는 것

은닉 : 구현은 숨기고, 동작은 노출시킴

느슨한 결합(Loose Coupling)에 유리 : 언제든지 구현을 수정할 수 있다.

캡슐로 된 감기약을 예로 들면 정보 은닉은 감기약에 어떤 재료가 들어 있는지 몰라도 감기가 걸렸을 때 먹는 약이라는 것만 알고 복용하는 것과 같은 의미이다.

추상화는 공통의 속성이나 기능을 묶어 이름을 붙이는 것을 말하며, 필요한(공통적인) 부분만 파악해서 추출하고 필요하지 않는 것은 제거한다.

클래스 정의 시, 메소드와 속성만 정의한 것을 인터페이스라 부른다. 이것이 추상화의 본질이라고 할 수 있다.

추상화(Abstraction)의 종류

- 과정 추상화 : 자세한 수행 과정을 정의하지 않고, 전반적인 흐름만 파악할 수 있게 설계하는 방법
- 데이터 추상화 : 데이터의 세부적인 속성이나 용도를 정의하지 않고, 데이터 구조를 대표할 수 있는 표현으로 대체하는 방법
- 제어 추상화 : 이벤트 발생의 정확한 절차나 방법을 정의하지 않고, 대표할 수 있는 표현으로 대체하는 방법

상속성(Inheritance)의 종류

- 단일 상속 : 하나의 상위 클래스로부터 상속받는 것
- 다중 상속 : 여러 개의 상위 클래스로부터 상속받는 것

프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(객체지향 프로그래밍 언어)

1. 객체지향의 기본 개념 중 객체가 메시지를 받아 실행해야 할 객체의 구체적인 연산을 정의한 것은?

- ① 메소드 ② 추상화
- ③ 상속성 ④ 캡슐화

객체지향 언어의 3요소

1. 객체(Object)

- 데이터(속성)와 이를 처리하기 위한 연산(메소드)을 결합시킨 실체(메모리상에 올라가는 것-인스턴스)이다.
- 데이터 구조와 그 위에서 수행되는 연산들을 가지고 있는 소프트웨어 모듈이다.
- 속성(Attribute) : 한 클래스 내에 속한 객체들이 가지고 있는 데이터 값들을 단위별로 정의하는 것으로서 성질, 분류, 식별, 수량 또는 객체의 현재 상태 등을 표현한다.
- 메소드(Method) : 객체가 서로 상호작용하는 메시지를 받아서 실행해야 할 때 구체적인 연산을 정의하는 것으로 객체의 상태를 참조하거나 변경하는 수단이 되는 것을 의미한다.

2. 클래스(Class)

3. 객체지향 개념에서 이미 정의되어 있는 상위 클래스(슈퍼 클래스 혹은 부모 클래스)의 메소드를 비롯한 모든 속성을 하위 클래스가 물려 받는 것을 무엇이라고 하는가?

- ① Abstraction(추상화) ② Method(기능, 동작, 연산)
- ③ Inheritance(상속성) ④ Message(객체의 상호작용 수단)

객체지향 개념의 특징

1. 캡슐화(Encapsulation)

- 데이터(속성)와 처리하는 함수를 하나로 묶는 것을 의미한다.
- 캡슐화된 객체의 세부 내용이 외부에 은폐(정보 은닉)되어, 변경이 발생할 때 오류의 파급 효과가 적다.
- 캡슐화된 객체들은 얼마든지 재사용 가능하다.

2. 정보 은닉(Information Hiding)

- 캡슐화에서 가장 중요한 개념으로, 다른 객체에게 자신의 정보를 숨기고 자신의 연산만을 통하여 접근을 허용(getter())하는 것이다.

3. 추상화(Abstraction)

- 불필요한 부분을 생략하고 객체의 속성 중에서 가장 중요한 것(공통)에만 중점을 두어 개략화하는 것, 즉 모델화하는 것이다.
- 데이터의 공통된 성질을 추출하여 슈퍼 클래스를 선정하는

프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(객체지향 프로그래밍 언어)

5. 객체지향 분석에서 불필요한 부분을 생략하고 객체의 속성 중 가장 중요한 것에만 중점을 두어 개략화시킨 것을 무엇이라고 하는가?

- ① 상속성 ② 클래스
- ③ 추상화 ④ 메시지

추상화(Abstraction)는 공통의 속성(멤버 변수, 필드)이나 기능(멤버 메소드)을 묶어서 이름을 붙이는 것을 말하며, 필요한

(공통적인) 부분만 파악해서 추출하고 필요하지 않는 것의 과감히 제거한다. 클래스 정의 시, 추상 메소드와 속성(상수)만 정의한 것은 인터페이스이다. 추상화다라고 하면 추상 클래스, 인터페이스를 떠올리면 된다. 추상화의 실질적인 본질은 추상 메서드이다. 추상 메서드가 1개라도 abstract라는 예약어가 붙어

있으면 인스턴스를 생성하지 못한다. 추상 클래스는 상속을 통해서 하위 클래스에서 추상 메서드를 재정의(overriding)하여 구현부가 있어야 비로소 인스턴스를 생성할 수가 있다. 인터페이스 역시 어떠한 클래스가 구현(implements)해야 비로소 인터페이스를 구현한 클래스는 인스턴스를 생성할 수 있다.

추상화의 종류

7. 객체지향 시스템에서는 객체가 시스템을 구성하는 기본 단위인데, 이런 객체 중에는 같은 특성을 갖는 객체들이 많다. 이와 같이 같은 특성을 갖는 객체를 표현한 것을 무엇이라 하는가?

- ① 속성 ② 클래스
- ③ 메시지 ④ 인스턴스

속성 : 객체가 가지고 있는 정보이며, 객체의 상태, 분류 등을 나타내는 클래스의 멤버 변수(필드)를 의미한다.

메시지 : 객체들 간에 상호작용을 하는데 사용되는 수단으로, 객체에게 어떤 행위를 하도록 지시하는 명령 또는 요구사항

인스턴스 : 클래스의 생성자를 호출하여 클래스의 정의된 멤버들을 사용자 메모리 영역 heap공간에 할당되어진 실체

8. 객체지향 프로그래밍 언어의 종류로 틀린 것은?

- ① JAVA ② C++
- ③ Smalltalk ④ ALGOL

알골(ALGOL, Algorithmic Language의 준말)은 1950년대 미국에서 만들어진 포트란에 대항하여 유럽의 학자들을 중심으로 개발된 절차적 프로그래밍 언어이다.

객체지향 프로그램 언어의 종류

프로그래밍 언어 활용-SEC_12(객체지향 프로그래밍 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(객체지향 프로그래밍 언어)

9. 객체지향 프로그래밍 언어의 장점이 아닌 것은?

- ① 상속을 통한 재사용과 시스템의 확장이 용이하다.
- ② 코드의 재활용성이 높다.
- ③ 대형 프로그램의 작성이 용이하다.
- ④ 구현 시 처리 시간이 줄어든다.

객체지향 프로그래밍 언어의 장,단점

- 상속을 통한 재사용과 시스템의 확장이 용이하다.
- 코드의 재활용성이 높다.
- 대형 프로그램의 작성이 용이하다.
- 자연적인 모델링에 의해 분석과 설계를 쉽고 효율적으로 할 수 있다.
- 사용자와 개발자 사이의 이해를 쉽게 해준다.
- 소프트웨어 개발 및 유지보수가 용이하다.
- 프로그래밍 구현을 지원해 주는 정형화된 분석 및 설계 방법은 존재하지 않는다.
- 구현 시 처리 시간이 지연된다.

4. 프로그래밍 언어 활용-SEC_13(스크립트 언어)

1) 스크립트 언어(Script Language)의 개요

; 스크립트 언어는 HTML 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것으로, 기계어로 컴파일 되지 않고 별도의 번역기(인터프리터)가 소스를 분석하여 동작하게 하는 언어이다.

- 게시판 입력, 상품 검색, 회원 가입 등과 같은 데이터베이스 처리 작업을 수행하기 위해 주로 사용한다.
- 스크립트 언어는 클라이언트의 웹 브라우저에서 해석되어 실행되는 클라이언트용 스크립트 언어와 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 서버용 스크립트 언어가 있다.
 - 서버용 스크립트 언어 : ASP(Active Server Page), JSP(Java Server Page), PHP(Professional Hypertext Preprocessor), Python
 - 클라이언트용 스크립트 언어 : 자바 스크립트(JAVA Script), VB 스크립트(Visual Basic Script)

서버 스크립트 : '서버 사이드 스크립트'라고 해서 서버에서 실행된다. 서버 스크립트를 구분하는 구분자는 <% 와 %>(스크립트릿)로, 이 사이에 사용된 스크립트는 서버에서 실행된다. 그렇기 때문에 브라우저에서 HTML 보기를 해도 서버 스크립트 언어의 코드 내용은 보이지 않는다. 서버에서 실행되고 그 결과만을 가져다 보여주기 때문에 HTML 소스에는 그 결과값만 보이게 된다.

클라이언트 스크립트 : 클라이언트가 웹 페이지 주소로 접속하면(웹 페이지 주소에 해당하는 정보를 요청하면)서버는 자바스크립트, VB스크립트 코드는 해석하지 않고 클라이언트의 웹 브라우저로 html을 보내는 기능을 한다.(요청한 정보에 응답)

4. 프로그래밍 언어 활용-SEC_13(스크립트 언어)

2) 스크립트 언어의 장·단점

- 스크립트 언어는 컴파일 언어에 비해 단순하고 쉬운 문법 구조를 갖고 있다.
- 컴파일 없이 바로 실행하므로 결과를 바로 확인할 수 있다.
- 배우고 프로그래밍 하기 쉽다.
- 개발 시간이 짧다.
- 소스 코드를 쉽고 빠르게 수정할 수 있다.
- 컴파일러 없이 명령어를 한 줄씩 읽으면서 실행하므로, 번역 속도는 빠르지만 프로그램 실행 시 매번 같은 코드를 번역해야 한다. 따라서 프로그램의 실행 속도는 컴파일 언어에 비해 느리다.
- CPU의 사용시간의 낭비가 크므로 복잡한 산술연산 혹은 복잡한 구조의 프로그램에서는 효율적이지 않을 수 있다.
- 컴파일 과정이 없기 때문에 프로그램을 실행시켜야 오류를 알 수 있다.
- 컴파일 과정이 없기 때문에, 소스 코드가 그대로 실행파일이 되어 메모리에 적재된다. 그 이후 런타임 시 메모리가 명령어를 실행하기 위해 내부적으로 기계어로 변환하는 과정을 거친다. 이 부분에서 런타임 오류가 많이 발생한다.

4. 프로그래밍 언어 활용-SEC_13(스크립트 언어)

3) 인터프리터 방식의 스크립트 언어(Script Language) vs 컴파일 언어(Compile Language)

	인터프리터 방식의 스크립트 언어 (Script Language)	컴파일 언어 (Compile Language)
번역 단위	행(코드 한 문장)	전체
번역 속도	빠름	느림
실행 속도	느림	빠름
기계어 번역 시기	메모리 적재 이후 내부적으로 번역	메모리 적재 이전 컴파일 과정
OS 고려	X	O
언어 예시	JavaScript, Python, JSP, jQuery 등	C, C++, Java 등

4. 프로그래밍 언어 활용-SEC_13(스크립트 언어)

4) 스크립트언어의 종류

자바스크립트 (JAVA Script)	<ul style="list-style-type: none">•웹 페이지의 동작을 제어하는데 사용되는 클라이언트 스크립트 언어이다.•클래스 기반의 객체 상속을 지원하여 객체지향 프로그래밍 언어의 성격도 갖고 있다.•Prototype Link와 Prototype Object를 통해 프로토타입 개념을 활용할 수 있다.
VB스크립트(Visual Basic Script)	마이크로소프트사에서 자바 스크립트에 대응하기 위해 제작한 언어로, Active X를 사용하여 마이크로소프트사의 애플리케이션들을 컨트롤할 수 있다.
ASP(Active Server Page)	<ul style="list-style-type: none">•서버 측에서 동적으로 수행되는 페이지를 만들기 위한 언어로 마이크로소프트사에서 제작하였다.•Windows 계열에서만 수행 가능한 프로그래밍 언어이다.
JSP(Java Server Page)	JAVA로 만들어진 서버용 스크립트로 다양한 운영체제에서 사용이 가능하다.
PHP(Professional Hypertext Preprocessor)	<ul style="list-style-type: none">•서버용 스크립트 언어로 Linux, Unix, Windows 운영체제에서 사용 가능하다.•C, Java 등과 문법이 유사하므로 배우기 쉬워 웹 페이지 제작에 많이 사용된다.
파이썬(Python)	<ul style="list-style-type: none">•귀도 반 로섬(Guido van Rossum)이 발표한 대화형 인터프리터 언어이다.•객체지향 기능을 지원하고 플랫폼에 독립적이며 문법이 간단하여 배우기 쉽다.
셸 스크립트	<ul style="list-style-type: none">•유닉스/리눅스 계열의 셸(Shell)에서 사용되는 명령어들의 조합으로 구성된 스크립트 언어이다.•컴파일 단계가 없어 실행 속도가 빠르다.•저장 시 확장자로 '.sh'가 붙는다.•셸의 종류 : Bash Shell, Bourne Shell, C Shell, Korn Shell 등•셸 스크립트에서 사용되는 제어문<ul style="list-style-type: none">- 선택형 : if, case- 반복형 : for, while, until
Basic	절차지향 기능을 지원하는 대화형 인터프리터 언어로 초보자도 쉽게 사용할 수 있는 문법 구조를 갖는다.

Prototype Link와 Prototype Object : 자바스크립트에서 프로토타입을 구현하기 위해 사용하는 개념으로 객체가 생성될 때 생성된 객체의 원형을 프로토타입 객체(Object)라고 하고, 생성된 객체와 원형을 연결하는 링크를 프로토타입 링크(Link)라고 한다.

Active X : 마이크로소프트사에서 Windows 환경의 응용 프로그램을 웹과 연결하기 위해 개발한 프로그램 기술로서, Active X를 이용하면 동적(Dynamic)인 콘텐츠와 응용 프로그램 제작이 편리하다.

인터프리터 언어 : 원시 프로그램을 줄 단위로 번역하여 바로 실행해 주는 언어로 목적 프로그램을 생성하지 않고 즉시 실행 결과를 출력한다.

셸(Shell) : 사용자의 명령어를 인식하여 프로그램을 호출하고 명령을 수행하는 명령어 해석기이다.

프로그래밍 언어 활용-SEC_13(스크립트 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(스크립트 언어)

1. 스크립트 언어가 아닌 것은?

- ① PHP ② Cobol
- ③ Basic ④ Python

COBOL은 1959년에 그레이스 호퍼(Grace Hopper) 즉, 프로그램 역사에서 두 번째 유명한 여성 프로그래머로부터 처음으로 개발되었다. 이후 여러 번 규격이 개정되어 현재는 객체 지향도 지원한다. 개념이 상당히 오래 되었지만 처음부터 일반인도 쓸 수 있도록 만들었기에 배우기 어렵지 않은 편이다. COBOL은 절차적 프로그래밍 언어이며, 용도는 사무 처리용 언어이다.

스크립트 언어의 종류

자바스크립트(JavaScript)

- 클라이언트 스크립트 언어의 하나이며, 웹 페이지 동작을 제어하는데 사용된다.
- 클래스 기반의 객체 상속을 지원하여, 객체지향 프로그래밍 언어의 성격도 갖추고 있다.
- Prototype Link와 Prototype Object를 통해 프로토타입 개념을 활용할 수 있다.

3. 다음 중 bash 셸 스크립트에서 사용할 수 있는 제어문이 아닌 것은?

- ① if ② for
- ③ repeat_do ④ while

bash 셸 스크립트에서 사용하는 제어문은 C, Java에서 사용하는 제어문과 다르지 않다. 셸 명령어 중에서는 repeat_do 란 명령어는 존재하지 않는다.

셸의 종류 : Bash Shell, Bourne Shell, C Shell, Korn Shell 등
셸 스크립트에서 사용되는 제어문의 종류

- 선택형 : if, case
- 반복형 : for, while, until

4. 자바스크립트(JavaScript)와 관련한 설명으로 틀린 것은?

- ① 프로토타입(Prototype)의 개념이 존재한다.
- ② 클래스 기반으로 객체 상속을 지원하지 않는다.
- ③ Prototype Link Prototype Object를 활용할 수 있다.
- ④ 객체지향 언어이다.

자바스크립트는 객체 상속은 물론 클래스 기반으로 작성하는 것도 가능하다. Prototype Link와 Prototype Object를 자바스크립트에서

프로그래밍 언어 활용-SEC_13(스크립트 언어) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(스크립트 언어)

5. 다음 중 게시판 입력, 상품 검색, 회원 가입 등과 같은 데이터 베이스 처리 작업을 수행하기 위해 사용하며, 웹 서버에서 작동하는 스크립트 언어들로만 모아 놓은 것은?

- ① HTML, XML, SGML ② Java, Java Applet, JavaScript
- ③ Java Script, VB Script ④ ASP, JSP, PHP

서버용 스크립트 언어 : ASP, JSP, PHP

클라이언트 스크립트 언어 : 자바스크립트, VB스크립트

XML(eXtensible Markup Language) : W3C에서 개발된, 다른 특수한 목적을 갖는 마크업 언어를 만드는데 사용하도록 권장하는 다목적 마크업 언어이다. XML은 SGML의 단순화된 부분집합으로, 다른 많은 종류의 데이터를 기술하는데 사용할 수 있다.

SGML(Standard Generalized Markup Language) : 문서용 마크업 언어를 정의하기 위한 메타 언어이다. IBM사에서 1960년대에 개발한 GML(Generalized Markup Language)의 후속이며, ISO표준이다.

메타 언어 : 언어학 및 논리학에서 대상 언어의 기술 내용을 범주화하거나 규칙화 하는데 사용되는 기술적인 언어이다.

Java Applet : Java로 작성한 프로그램을 미리 업로드하고 브라우저

4. 프로그래밍 언어 활용-SEC_14(선언형 언어)

1) 선언형 언어

; 선언형 언어는 명령형 언어와 반대되는 개념의 언어로, 명령형 언어가 문제를 해결하기 위한 방법을 기술한다면 선언형 언어는 프로그램이 수행해야 하는 문제를 기술하는 언어이다.

- 선언형 언어는 목표를 명시하고 알고리즘은 명시하지 않는다.
- 선언형 언어에는 함수형 언어와 논리형 언어 등이 있다.

함수형 언어	<ul style="list-style-type: none">• 수학적 함수를 조합하여 문제를 해결하는 언어로, 알려진 값을 함수에 적용하는 것을 기반으로 한다.• 적용형 언어라고도 한다.• 재귀호출이 자주 이용된다.• 병렬 처리에 유리하다.• 종류 : LISP
논리형 언어	<ul style="list-style-type: none">• 기호 논리학에 기반을 둔 언어로, 논리 문장을 이용하여 프로그램을 표현하고 계산을 수행한다.• 선언적 언어라고도 한다.• 반복문이나 선택문을 사용하지 않는다.• 비절차적 언어이다.• 종류 : PROLOG

목표를 명시하고 알고리즘은 명시하지 않는다는 것은 대표적인 선언형 언어로 HTML이 있는데, HTML은 웹 페이지를 구성할 때 제목 글꼴, 그림 등 웹 페이지에 표시할 것들을 묘사하지만 화면에 어떤 방법으로 표시할지는 묘사하지 않는 것을 의미한다.

함수형 언어와 논리형 언어는 선언형 언어에는 함수형 언어나 논리형 언어 외에 다른 언어들도 포함된다. HTML의 경우는 완전한 선언형 언어지만 함수형 언어나 논리형 언어에 포함되지는 않는다.

LISP : LISP의 경우 C 언어 처럼 자료형을 선언하거나 변수를 선언할 필요가 없다. 리스프 혹은 리습은 프로그래밍 언어의 계열로서, 오랜 역사와 독특하게 괄호를 사용하는 문법으로 유명하다. 1958년에 초안이 작성된 이 언어는 현재 널리 사용되는 포트란에 이어 두 번째로 오래된 고급 프로그래밍 언어이다.

프롤로그(Prolog) : 1973년 프랑스 마르세유 대학교의 알랭 콜메르(Alan Colmerauer)가 개발한 언어로서, 논리식을 토대로 하여 오브젝트와 오브젝트 간의 관계에 관한 문제를 해결하기 위해 사용한다. 프롤로그는 술어 논리식을 프로그램, 증명하는 것을 계산한다는 것으로 간주하는 관점에서 새로운 계산의 기술 형태를 취하고 있다. 즉, 프로그램 자체는 논리식의 모양으로 만들어지고, 그 프로그램을 실행하는 처리계가 그 증명기로 되어 있다.

인공지능이나 계산 언어학 분야, 특히 프롤로그가 만들어진 목적이었던 자연언어 처리 분야에서 많이 사용된다.

4. 프로그래밍 언어 활용-SEC_14(선언형 언어)

2) 명령형 언어

; 명령형 언어는 순차적인 명령 수행을 기본으로 하는 언어로, 문제를 처리하기 위한 방법에 초점을 두고 코드를 작성한다.

- 폰노이만 구조에 개념적인 기초를 두고 있다.
- 명령형 언어는 알고리즘을 명시하고 목표는 명시하지 않는다.
- 특정 구문의 연산을 이용하여 상태를 변경시키고 프로그램을 동작시킨다.
- 개체의 동작과 상태를 중요시 한다.
- 명령형 언어에는 절차적 언어와 객체지향 언어가 있다.
- 종류 : FORTRAN, COBOL, C, JAVA 등

존 폰 노이만이 제시한 컴퓨터 구조이며, 프로그램 내장 방식이라고도 불리며, 이론상 튜링 머신과 같은 일을 할 수 있다. 폰노이만 구조가 등장하기 이전의 컴퓨터들은 스위치를 설치하고 전선을 연결하여 데이터를 전송하고 신호를 처리하는 식으로 프로그래밍을 하였다. 그리고 폰 노이만 구조는 중앙처리장치(CPU), 메모리, 프로그램 이 세 가지 구성요소로 이루어져 있다.

폰노이만 구조의 디지털 컴퓨터에서는 "저장된 프로그램"(stored-program)의 개념이 도입되었다. 이는 프로그램을 구성하는 명령어들을 임의 접근이 가능한 메모리상에 순차로 배열하고, 동시에 조건 분기를 무제한으로 허용한다는 것을 뜻한다. 폰노이만 구조에서는 같은 메모리 속에 실행 코드와 데이터가 따로 구분되지 않고 함께 섞여 있다.

아울러, 컴퓨터에 다른 작업을 시키려고 할 때 굳이 하드웨어(전선)를 재배치할 필요 없이 소프트웨어(프로그램)만 교체하면 되기 때문에 범용성이 크게 향상된다는 것이다. 전선을 일일이 교체하면 교체 인원도 많이 필요하고 시간도 많이 잡아먹는 등 여러모로 불편함이 있지만, 폰노이만 구조를 도입하면 프로그램을 교체하는 것으로 모든 일이 끝난다.

튜링 머신 : 수학자 앨런 튜링이 1936년에 제시한 개념으로 계산하는 기계의 일반적인 개념을 설명하기 위한 가상의 기계를 말한다.

4. 프로그래밍 언어 활용-SEC_14(선언형 언어)

3) 선언형 언어의 장·단점

- 가독성이나 재사용성이 좋다.
- 작동 순서를 구체적으로 작성하지 않기 때문에 오류가 적다.
- 프로그램 동작을 변경하지 않고도 관련 값을 대체할 수 있다.

4) 선언형 프로그래밍 언어 종류

HTML	인터넷의 표준 문서인 하이퍼텍스트 문서를 만들기 위해 사용하는 언어로, 특별한 데이터 타입이 없는 단순한 텍스트이므로 호환성이 좋고 사용이 편리하다.
LISP	•인공지능 분야에 사용되는 언어이다. •기본 자료 구조가 연결 리스트 구조이며, 재귀(Recursion) 호출을 많이 사용한다.
PROLOG	논리학을 기초로 한 고급 언어로 인공지능 분야에서의 논리적인 추론이나 리스트 처리 등에 주로 사용된다.
XML	•기존 HTML의 단점을 보완하여 웹에서 구조화된 폭넓고 다양한 문서들을 상호 교환할 수 있도록 설계된 언어이다. •HTML에 사용자가 새로운 태그(Tag)를 정의할 수 있으며, 문서의 내용과 이를 표현하는 방식이 독립적이다.
Haskell	•함수형 프로그래밍 언어로 부작용(Side Effect)이 없다. •코드가 간결하고 에러 발생 가능성이 낮다.

프로그래밍 언어 활용-SEC_14(선언형 언어) 출제 예상 문제

출제 예상 문제(선언형 언어)

1. 다음 중 언어의 분류에 대한 설명으로 틀린 것은?

- ① 객체들이 모여서 하나의 프로그램이 되는 것은 객체지향 언어이다.
- ② 폰노이만 구조에 개념적 기초를 둔 언어는 명령형 언어이다.
- ③ 명령문을 순서대로 나열한 것은 선언형 언어이다.
- ④ 수학적 함수를 조합하여 문제를 해결하는 언어는 함수형 언어이다.

명령형 언어

명령형 언어는 순차적인 명령 수행을 기본으로 하는 언어로, 문제를 처리하기 위한 방법에 초점을 맞추어 코드를 작성한다.

- 폰 노이만 구조에 개념적인 기초를 두고 있다.
- 명령형 언어는 알고리즘을 명시하고 목표는 명시하지 않는다.
- 특정 구문의 연산을 이용하여 상태를 변경시키고 프로그램을 동작시킨다.
- 개체의 동작과 상태를 중요시 한다.
- 명령형 언어에는 절차적 언어와 객체지향 언어가 있다.
- 명령형 언어의 종류 : FORTRAN, COBOL, C, JAVA 등

3. 프로그래밍 언어의 전형에 해당하지 않는 것은?

- ① 선언형 언어
- ② 함수형 언어
- ③ 명령형 언어
- ④ 추상형 언어

추상형 언어는 존재하지 않는다. 다만 객체지향 개념에서 추상화란 개념이 존재할 뿐이다.

4. 다음 중 언어의 분류에 관한 설명으로 옳지 않은 것은?

- ① 논리형 언어는 기호 논리학을 기반으로 두고 있다.
- ② 함수형 언어는 적용형 언어라고도 한다.
- ③ 논리형 언어는 선언형 언어에 포함된다.
- ④ 함수형 언어는 폰노이만 구조에 개념적 기초를 두고 있다.

폰노이만 구조에 개념적 기초를 두고 있는 것은 명령형 언어이다.

폰 노이만 구조는 존 폰 노이만이 제시한 컴퓨터 구조이며, 프로그램 내장 방식이라고도 불리며, 이론상 튜링 머신과 같은 일을 할 수 있다.

폰 노이만 구조가 등장하기 이전의 컴퓨터들은 스위치를 설치하고 전선을 연결하여 데이터를 전송하고 신호를 처리하는 식으로 프로그래밍을 하였다. 그리고 폰 노이만 구조는 중앙처리장치(CPU), 메모리, 프로그램 이

4. 프로그래밍 언어 활용-SEC_15(라이브러리)

1) 라이브러리의 개념

; 라이브러리는 프로그램을 효율적으로 개발할 수 있도록 자주 사용하는 함수나 데이터들을 미리 만들어 모아 놓은 집합체이다.

- 자주 사용하는 함수들의 반복적인 코드 작성을 피하기 위해 미리 만들어 놓은 것으로, 필요할 때는 언제든지 호출하여 사용할 수 있다.
- 프로그래밍 언어에 따라 일반적으로 도움말, 설치 파일, 샘플 코드 등을 제공한다.
- 라이브러리는 모듈과 패키지 모두를 의미한다.
 - 모듈 : 하나의 기능이 한 개의 파일로 구현된 형태
 - 패키지 : 하나의 패키지 즉, 폴더 안에 여러 개의 모듈을 모아 놓은 형태
- 라이브러리에는 표준 라이브러리와 외부 라이브러리가 있다.
- 표준 라이브러리 : 프로그래밍 언어에 기본적으로 포함되어 있는 라이브러리로, 여러 종류의 모듈이나 패키지 형태이다.
- 외부 라이브러리 : 개발자들이 필요한 기능들을 만들어 인터넷 등에 공유해 놓은 것으로, 외부 라이브러리를 다운받아 설치한 후 사용한다.

4. 프로그래밍 언어 활용-SEC_15(라이브러리)

2) C 언어의 대표적인 표준 라이브러리

; C 언어는 라이브러리를 헤더 파일로 제공하는데, 각 헤더 파일에는 응용 프로그램 개발에 필요한 함수들이 정리되어 있다.

- C 언어에서 헤더 파일을 사용하려면 '#include <stdio.h>'와 같이 include문을 이용해 선언한 후 사용해야 한다.

헤더 파일	기능
stdio.h	•데이터의 입·출력에 사용되는 기능들을 제공한다. •주요 함수 : printf, scanf, fprintf, fscanf, fclose, fopen 등
math.h	•수학 함수들을 제공한다. •주요 함수 : sqrt, pow, abs 등
string.h	•문자열 처리에 사용되는 기능들을 제공한다. •주요 함수 : strlen, strcpy, strcmp 등
stdlib.h	•자료형 변환, 난수 발생, 메모리 할당에 사용되는 기능들을 제공한다. •주요 함수 : atoi, atof, srand, rand, malloc, free 등
time.h	•시간 처리에 사용되는 기능들을 제공한다. •주요 함수 : time, clock 등

fprintf() : printf()와 유사하지만 앞에 f라는 글자가 붙어 있는데 이는 file을 뜻한다. 파일 스트림에 서식화된 문자열을 출력하는 함수다.

fscanf() : C언어 표준 함수로 scanf 함수와 동일하지만 표준 입력(stdin)이 아닌 파일로부터 데이터를 받아서 저장한다.

fopen() : 파일의 경로를 통해 파일을 여는 함수, fclose : 열린 파일을 닫는 함수

sqrt() : 제곱근 계산 함수, pow() : 제곱 값 계산 함수, abs() : 절대 값 구하는 함수

strlen() : 문자열 길이 구하는 함수, strcpy() : 문자열 복사 함수, strcmp() : 문자열 비교 함수

atoi() : 문자열을 정수 값으로 변환하는 함수, atof() : 문자열을 소수점 값으로 변환

srand() : s는 seed라는 뜻으로 이 seed값에 따라 rand의 값이 바뀌게 하는 함수, rand() : 난수를 생성 함수

time() : 현재 날짜와 시간을 얻기 위한 함수, clock() : 시간 측정 함수

4. 프로그래밍 언어 활용-SEC_15(라이브러리)

3) JAVA의 대표적인 표준 라이브러리

; JAVA는 라이브러리를 패키지에 포함하여 제공하는데, 각 패키지에는 JAVA 응용 프로그램 개발에 필요한 메소드들이 클래스로 정리되어 있다.

- JAVA에서 패키지를 사용하려면 'import java.util.*'과 같이 import문을 이용해 선언한 후 사용해야 한다.
- import로 선언된 패키지 안에 있는 클래스의 메소드를 사용할 때는 클래스와 메소드를 마침표(.)로 구분하여 'Math.abs()'와 같이 사용한다.

패키지명	기능
java.lang	•자바에 기본적으로 필요한 인터페이스, 자료형, 예외 처리 등에 관련된 기능을 제공한다. •import문 없이도 사용할 수 있다. •주요 클래스 : String, System, Process, Runtime, Math, Error 등
java.util	•날짜 처리, 난수 발생, 복잡한 문자열 처리 등에 관련된 기능을 제공한다. •주요클래스 : Date, Calender, Random, StringTokenizer 등
java.io	•파일 입·출력과 관련된 기능 및 프로토콜을 제공한다. •주요 클래스: InputStream, OutputStream, Reader, Writer 등
java.net	•네트워크와 관련된 기능을 제공한다. •주요 클래스 : Socket, URL, InetAddress 등
java.awt	•사용자 인터페이스(UI)와 관련된 기능을 제공한다. •주요 클래스 : Frame, Panel, Dialog, Button, Checkbox 등

프로그래밍 언어 활용-SEC_15(라이브러리) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(라이브러리)

1. 라이브러리의 개념과 구성에 대한 설명 중 틀린 것은?

- ① 라이브러리란 필요할 때 찾아서 쓸 수 있도록 모듈화 되어 제공되는 프로그램을 말한다.
- ② 프로그래밍 언어에 따라 일반적으로 도움말, 설치 파일, 샘플 코드 등을 제공한다.
- ③ 외부 라이브러리는 프로그래밍 언어가 기본적으로 가지고 있는 라이브러리를 의미하며, 표준 라이브러리는 별도의 파일 설치를 필요로 하는 라이브러리를 의미한다.
- ④ 라이브러리는 모듈과 패키지를 총칭하며, 모듈이 개별 파일 이라면 패키지는 파일들을 모아 놓은 폴더라고 볼 수 있다.

라이브러리는 프로그램을 효율적으로 개발할 수 있도록 자주 사용하는 함수나 데이터들을 미리 만들어 모아 놓은 집합체이다.

- 자주 사용하는 함수들의 반복적인 코드 작성을 피하기 위해 미리 만들어 놓은 것으로, 필요할 때는 언제든지 호출하여 사용할 수 있다.
- 프로그래밍 언어에 따라 일반적으로 도움말, 설치 파일, 샘플 코드 등을 제공한다.
- 라이브러리는 모듈과 패키지 모두를 의미한다.

3. C 언어에서 문자열을 정수형으로 변환하는 라이브러리 함수는?

- ① atoi() ② atof()
- ③ itoa() ④ ceil()

atoi() : 문자열을 정수 값으로 변환하는 함수

atof() : 문자열을 소수점 값으로 변환

itoa() : 정수를 문자열로 변환하는데 다만, itoa()함수의 경우 비 표준 함수이다. C99/C11 규격에서 정의되지 않고, MS 컴파일러에서만 이용 가능한 함수이다. 즉 Linux/mac OS 등 비 마이크로 소프 운영 체제(컴파일러)에서는 stdlib.h 헤더파일을 include를 한다고 해도 itoa()함수를 사용할 수가 없다는 제약이 있다.

ceil() : 무조건 소수점을 올림 하는 함수이고, 정수 값을 반환하지만 타입은 double형으로 반환된다.

floor() : 무조건 소수점을 내림 하는 함수이고, 정수 값을 반환하지만 타입은 double형으로 반환된다.

4. C언어에서 malloc() 함수에 대한 설명으로 틀린 것은?

- ① 원하는 시점에 원하는 만큼 메모리를 동적으로 할당한다.
- ② 사용자가 입력한 bit만큼 메모리를 할당한다.
- ③ free 명령어로 할당된 메모리를 해제한다.

프로그래밍 언어 활용-SEC_15(라이브러리) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(라이브러리)

5. JAVA의 대표적인 표준 라이브러리에 대한 설명이 잘못된 것은?

① JAVA에서 패키지를 사용하려면 'import java.util'와 같이

import문을 이용해 선언한 후 사용해야 한다.

② import로 선언된 패키지 안에 있는 클래스의 메소드를 사용할 때는 클래스와 메소드를 콜론(:)으로 구분하여 Math.abs()와 같이 사용한다.

③ JAVA의 주요 패키지에는 java.lang, java. util, java.io, java.net 등이 있다.

④ java.lang' 패키지는 자바에 기본적으로 필요한 인터페이스, 자료형, 예외 처리 등에 관련된 기능을 제공한다.

JAVA에서 클래스의 메서드를 호출할 때는 클래스와 정적 메서드를 마침표(접근 연산자, .)를 이용하며, 방식은 Math.abs()와 같이 사용한다.

JAVA언어의 주요 패키지

java.lang

- 자바에 기본적으로 필요한 인터페이스, 자료형, 예외 처리 등에 관련된 기능을 제공한다.

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

1) 오류의 개요

; 에러(Error)

- 의도치 않게 프로그램이 종료되는 것을 에러라 한다.
- 하드웨어의 잘못된 동작 또는 고장으로 인한 오류(ex. 정전, 배드섹터 등)
- 에러가 발생하면 프로그램이 비정상 종료가 된다.
- 정상 실행 상태로 돌아갈 수가 없다.

; 예외(Exception)

- 사용자의 잘못된 조작 또는 개발자의 잘못된 프로그래밍으로 인한 오류를 말한다.
- 예외가 발생하면 프로그램이 종료된다.
- 단, 예외 처리를 추가하면 정상 실행 상태로 되돌릴 수가 있다.

● 예외 처리의 목적 : 프로그램을 정상 실행 상태로 돌리는 것!!

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

2) 예외 처리의 개요

; 프로그램의 정상적인 실행을 방해하는 조건이나 상태를 예외(Exception)라고 하며, 이러한 예외가 발생했을 때 프로그래머가 해당 문제에 대비해 작성해 놓은 처리 루틴을 수행하도록 하는 것을 예외 처리(Exception Handling)라고 한다.

- 예외가 발생했을 때 일반적인 처리 루틴은 프로그램을 종료시키거나 로그를 남기도록 하는 것이다.
- C++, Ada, JAVA, 자바스크립트와 같은 언어에는 예외 처리 기능이 내장되어 있으며, 그 외의 언어에서는 필요한 경우 조건문을 이용해 예외 처리 루틴을 작성한다.
- 예외의 원인에는 컴퓨터 하드웨어 문제(에러), 운영체제의 설정 실수, 라이브러리 손상, 사용자의 입력 실수, 받아들이지 못하는 연산, 할당하지 못하는 기억장치 접근 등 다양하다.

Ada : 에이다(Ada)는 구조화되고, 정적인 형태를 가지고, 명령적이며, 객체지향적인 고급 컴퓨터 프로그래밍 언어이다. 처음에는 1977년에서 1983년까지 수백 개의 프로그래밍 언어를 대신할 목적으로 고안된 것이다. 에이다는 C나 C++와 몇 가지 작업이 같지만, 매우 강력한 유형 시스템의 언어이다.

에이다는 컴퓨터 프로그래밍을 발명하는데 공헌한 에이다 러브레이스의 이름을 딴 것이다.

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

3) JAVA의 예외의 종류와 처리

; 예외의 종류

- 일반 예외(컴파일 예외, checked exception) : 예외 처리 코드가 없다면, 컴파일 예외 발생함
(try ~ catch문 이용)
- 실행 예외(Runtime-Exception, unchecked exception)
 - 예외 처리 코드를 생략하더라도 컴파일이 되는 예외를 말한다.
(ex. NullPointerException, ArrayIndexOutOfBoundsException, NumberFormatException 등)
 - 단, 프로그래머의 경험 혹은 노하우에 따라 예외 처리 코드 작성이 필요하다.

; JAVA는 잘못된 동작이나 결과에 영향을 줄 수 있는 예외를 객체로 취급하며, 예외와 관련된 클래스를 java.lang 패키지에서 제공한다.

- JAVA에서는 try ~ catch 문을 이용해 예외를 처리한다.
- try 블록 코드를 수행하다 예외가 발생하면 예외를 처리하는 catch 블록으로 이동하여 예외 처리 코드를 수행하므로 예외가 발생한 이후의 코드는 실행되지 않는다.
- catch 블록에서 선언한 변수는 해당 catch 블록에서만 유효하다.
- try ~ catch 문 안에 또 다른 try ~ catch 문을 포함할 수 있다.
- try ~ catch 문 안에서는 실행 코드가 한 줄이라도 중괄호({})를 생략할 수 없다.

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

3) JAVA의 예외의 종류와 처리

; 예외 발생 시 예외 처리 코드를 작성하면, 프로그램의 종료를 막고 정상적으로 실행을 유지할 수 있도록 처리한다.

- 일반 예외 : 반드시 작성해야 한다.(컴파일 조차 시켜주지 않는다.)
- 실행 예외 : 컴파일러가 체크를 해주지 않는다. 하여, 개발자의 경험으로 작성한다.

```
Thread.sleep(1000);
```

일반 예외 발생하는 예

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

4) JAVA의 예외 처리 형식

예외 처리의 기본 형식

```
public static void main(String[] args) {  
    try {  
        //예외가 발생할 가능성이 있는 코드;  
    }  
    catch (NumberFormatException e) {    //(예외 객체1 매개변수)  
        //NumberFormatException 예외에 해당하는 예외 발생시 처리 코드;  
    }  
    catch (ArithmeticException e) { //(예외 객체2 매개변수)  
        //ArithmeticException 예외에 해당하는 예외 발생시 처리 코드;  
    }  
    catch (ArrayIndexOutOfBoundsException e) { //(예외 객체3 매개변수)  
        //ArrayIndexOutOfBoundsException 예외에 해당하는 예외 발생시 처리 코드;  
    }  
    catch (Exception e) {    //Exception 클래스는 예외에서 최고 조상 클래스  
        //위 3개의 catch블록에서 처리하지 못한 것들은 다형성에 의거하여  
        //이 블록에서 다 처리된다.  
    }  
    finally {  
        //예외의 발생 여부와 관계없이 무조건 처리되는 코드;  
    }  
}
```

위의 코드에서는 만약 예외가 발생하면 catch문 중 하나만 실행하고 예외처리 블록을 빠져 나온다.

즉, catch문은 오로지 하나만 실행하고 빠져 나오는 것을 기억하자.

Exception 클래스는 예외의 최상위 클래스이다. 발생 예외는 무조건 Exception클래스를 상속 받으므로 하위 catch구문은 절대 실행이 안 된다.

즉, 다시 말해 Exception의 작성은 catch문의 맨 아래에 작성을 해야 한다.

4. 프로그래밍 언어 활용-SEC_16(예외 처리)

5) JAVA의 주요 예외 객체

예외 객체	발생 원인
ClassNotFoundException	클래스를 찾지 못한 경우
NoSuchMethodException	메소드를 찾지 못한 경우
FileNotFoundException	파일을 찾지 못한 경우
ArithmeticException	0으로 나누는 등의 산술 연산에 대한 예외가 발생한 경우
IllegalArgumentException	잘못된 인자를 전달한 경우
NumberFormatException	숫자 형식으로 변환할 수 없는 문자열을 숫자 형식으로 변환한 경우
ArrayIndexOutOfBoundsException	배열의 범위를 벗어난 접근을 시도한 경우
NegativeArraySizeException	0보다 작은 값으로 배열의 크기를 지정한 경우
NullPointerException	존재하지 않는 객체를 참조한 경우

프로그래밍 언어 활용-SEC_16(예외 처리) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(예외 처리)

1. JAVA의 예외(exception)와 관련한 설명으로 틀린 것은?

- ① 문법 오류로 인해 발생한 것
 - ② 오동작이나 결과에 악영향을 미칠 수 있는 실행 시간 동안에 발생한 오류
 - ③ 배열의 인덱스가 그 범위를 넘어서는 경우 발생하는 오류
 - ④ 존재하지 않는 파일을 읽으려고 하는 경우에 발생하는 오류
- 문법 오류로 인해 발생한 것은 Syntax Error이며 예외에 속하지 아니한다.

예외 처리(Exception Handling)

프로그램의 정상적인 실행을 방해하는 조건이나 상태를 예외(Exception)라고 하며, 이러한 예외가 발생했을 때 프로그래머가 해당 문제에 대비해 작성해 놓은 처리 루틴을 수행하도록 하는 것을 예외 처리라고 한다.

- 예외가 발생했을 때 일반적인 처리 루틴은 프로그램을 종료 시키거나 로그를 남기도록 하는 것이다.
- C++, Ada, JAVA, 자바스크립트와 같은 언어에는 예외 처리 기능이 내장되어 있으며, 그 외의 언어에서는 필요한 경우 조건문을

3. 다음은 JAVA에서의 주요 예외 처리 객체에 대한 원인을 설명한 것이다. 객체에 대한 원인이 잘못 설명된 것은?

- ① ClassNotFoundException: 클래스를 찾지 못한 경우
- ② FileNotFoundException: 파일을 찾지 못한 경우
- ③ ArithmeticException: 숫자 형식으로 변환할 수 없는 문자열을 숫자 형식으로 변환하는 경우
- ④ ArrayIndexOutOfBoundsException: 배열의 범위를 벗어난 접근을 시도하는 경우

예외 객체의 종류

ClassNotFoundException : 클래스를 찾지 못하는 경우

NoSuchMethodException : 메소드를 찾지 못하는 경우

FileNotFoundException : 파일을 찾지 못한 경우

ArithmeticException : 0으로 나누는 등의 산술 연산에 대한 예외가 발생한 경우

IllegalArgumentException : 잘못된 인자가 전달된 경우

ArrayIndexOutOfBoundsException : 배열의 범위를 벗어난 접근을 시도한 경우

NegativeArraySizeException : 0보다 작은 값(음수)으로 배열의 크기를

4. 프로그래밍 언어 활용-SEC_17(프로토타입)

1) 프로토타입(Prototype)의 개념

; 프로그래밍 언어에서 프로토타입이란 함수 원형(Function Prototype)이라는 의미로, 컴파일러에게 사용될 함수에 대한 정보를 미리 알리는 것이다.

- 함수가 호출되기 전에 함수가 미리 정의되는 경우에는 프로토타입을 정의하지 않아도 된다.
- 프로토타입은 본문이 없다는 점을 제외하고 함수 정의와 형태가 동일하다.
- 프로토타입에 정의된 반환 형식은 함수 정의에 지정된 반환 형식과 반드시 일치해야 한다.

4. 프로그래밍 언어 활용-SEC_17(프로토타입)

2) C 언어에서의 프로토타입 선언

; C 언어에서 프로토타입은 main() 함수 바깥쪽에 선언해야 한다.

● 기본 형식

```
int func(int i, int j);
```

- int : 반환될 값의 자료형을 적는다. 반환될 값이 없으면 void를 적는다.
- func : 사용할 함수의 이름이다. 사용자가 임의로 지정하면 된다.
- (int i, int j) : 함수에서 사용할 매개변수다. 호출하는 곳에서 보내준 값의 순서와 자료형이 일치해야 한다.

4. 프로그래밍 언어 활용-SEC_17(프로토타입)

2) C 언어에서의 프로토타입 선언

; 예1)과 같이 main() 함수가 시작되기 전에 함수를 정의한 경우에는 프로토타입을 선언하지 않아도 되지만, 예2)와 같이 main() 함수가 시작된 후에 함수를 정의한 경우에는 main() 함수 전에 사용될 함수에 대한 프로토타입을 선언해야 한다.

예제1)

```
#include<stdio.h>

int func(int i, int j) {
    return i + j;
}

main(void) {

    int a = 10, b = 20, result = 0;
    result = func(i:a, j:b);    //func()함수 호출
    printf(_Format: "%d\n", result);
}
```

예제2)

```
#include<stdio.h>
int func(int i, int j); //func() 함수의 프로토타입 선언

main(void) {

    int a = 10, b = 20, result = 0;
    result = func(a, b);    //func()함수 호출
    printf("%d\n", result);
}

//func()의 구현(실체)
int func(int i, int j) {
    return i + j;
}
```

main() 함수 앞에 프로토타입을 선언하는 것은 main 함수가 시작되기 전에 이런 함수를 사용할 것이라고 알려주는 것이다. 최근에는 컴파일러가 발전하여 사용할 함수를 main() 함수 뒤에 배치한 경우에도 프로토타입 선언 없이 컴파일이 가능하지만, 프로토타입을 미리 선언하는 것이 원칙이라는 것을 기억하자.

프로그래밍 언어 활용-SEC_17(프로토타입) 출제 예상 문제

출제 예상 문제(프로토타입)

1. 다음 중 프로토타입에 대한 설명으로 잘못된 것은?

① 프로그래밍 언어에서 프로토타입이란 함수 원형(Function Prototype)이라는 의미이다.

② 프로토타입은 컴파일러에게 사용될 함수에 대한 정보를 미리 알리는 것이다.

③ 함수가 호출되기 전에 함수가 미리 정의되는 경우에는 프로토타입을 선언하지 않아도 된다.

④ 프로토타입에 정의된 반환 형식은 함수 정의에 지정된 반환 형식과 반드시 일치할 필요는 없다.

프로토타입에 정의된 반환 형식은 함수 정의에 지정된 반환 형식 과 반드시 일치해야 한다.

프로토타입(Prototype)의 개념

프로그래밍 언어에서 프로토타입이란 함수의 원형(Function Prototype)이라는 의미로, 컴파일러에게 사용될 함수에 대한 정보를 미리 알려주는 것이다.

- 함수가 호출되기 전에 main()위에 함수가 미리 정의된 경우에는 함수의 프로토타입을 정의하지 않아도 된다.

3. 다음 중, 함수의 프로토타입에서 반환값이 없을 때 사용하는 예약어는?

① void ② int

③ float ④ int *

반환값(리턴값)이 없을 때는 void(공허한, ~없는)를 사용한다. 아울러 구현부에서는 반환값이 void이면 return문을 생략하거나 return; 과 같이 사용할 수 있다.

4. 다음 중, 아래 코드에서 함수의 매개변수가 int* 타입일 때 인자 값으로 올 수 없는 것은?

```
void func(int* x);
```

```
main() {
```

```
    int a = 10, b = 20, result = 0;
```

```
    int arr[] = { 1,2,3,4,5 };
```

```
    func( ? );
```

```
}
```

① &result ② &a

③ &b ④ &arr

arr은 배열명이기에 배열명은 주소이다. 하여 주소 연산자 &를 붙이지



감사합니다.