

# 1. 언어 계산 전형(paradigm)

컴퓨터 프로그래밍 언어는 4가지 - 명령형 언어, 함수형 언어, 논리형 언어, 객체지향 언어 - 로 구분한다.

## 1. 명령형 또는 절차적 언어(Imperative/Procedural language)

- ① 폰 노이만식 기계 구조에 기초하여 설계된 언어이다.
- ② 프로그래밍 단위는 문장이다. 즉, 일련의 명령문 나열이다.
- ③ 문제를 여러 개의 작은 함수로 나누어 그 문제를 해결할 수 있다.
- ④ 다음 3가지 특징을 갖는다.
  - 변 수 : 자료가 저장되는 기억공간 셀(cell)에 붙인 이름이다.
  - 배정문 : 계산 결과는 셀에 저장되어야 한다.
  - 반 복 : 일련의 명령을 반복 수행하여 문제를 해결한다.

[예] C, Algol, Basic, PL/1, Pascal, FORTRAN, COBOL 등 많은 언어

## 2. 함수형 또는 적용적 언어(Function/Applicative language)

- ① 수학 함수에 기반을 둔다.
- ② 함수를 결합하여 좀 더 강력한 함수를 만들어 간다.(프로그램 방식)
- ③ 변수와 배정문이 없다.(식은 존재한다)
- ④ 반복 연산은 재귀적 함수 이론에 근거한다.

[예] Lisp, Common Lisp, Scheme, ML, Miranda, Haskell, FP, HOPE

// 명령형 언어와 함수형 언어 비교

	명령형 언어(절차적 언어)	함수형 언어
프로그램	일련의 명령문 나열한다.	함수 정의와 함수를 적용한다.
자료구조	여러 가지 종류	리스트(list)
변수	위치가 부여된 메모리 셀	함수로 정의(변수는 사용하지 않는다)
배정문	배정문이 주된 문장	모든 계산에는 함수를 이용
함수	부작용 발생 가능	순수함수는 부작용이 발생하지 않음
제어구조	순차, 조건, 반복 구조	함수 합성
참조투명성	불투명	투명
기계구조	폰 노이만식 기계	주소 개념이 없는 기계에 적절
구문	영문과 대수의 혼합 형태	프로그램 코드와 데이터는 같은 형태

### 3. 논리형 또는 선언적 언어(Logical/Declarative language)

- ① 기호 논리학, 특히 술어 해석학(predicate calculus)에 근거한다.
- ② 논리 프로그래밍은 비절차적으로 선언적이다.
- ③ 예를 들어, 검색 프로그램을 작성하는 경우에
  - 명령형 언어 : 순차검색, 이진검색 등처럼 검색 과정을 알고리즘으로 기술하는 방식이다.
  - 논리형 언어 : 중간 과정은 기술하지 않고, '무엇을 검색할 것이냐?'를 기술하는 방식이다.

### 4. 객체지향 언어

- 객체를 지원한다. → 객체는 프로그램의 기본적인 구성 단위
- 모든 객체는 클래스에 속한다. → 각 객체는 어떤 타입(클래스)의 인스턴스(사례)
- 상속을 지원한다.(Cardelli와 Wagner의 정의) → 클래스는 상속으로 서로 연관
- 객체 간에는 메시지(message)를 통해 명령을 전달한다.
- 예 : 파이썬, Simula67, Smalltalk, C++, Ada95, Java, C#, Eiffel, Modula-3, Object C

객체기반 언어	<ul style="list-style-type: none"><li>• 객체를 지원하고, 모든 객체는 클래스에 속한다.</li><li>• 하지만, 객체기반언어는 계승(상속)을 지원하지 않는다.</li><li>• 계승(상속)을 지원하지 않을 때, 객체기반언어라고 한다.</li><li>• 예 : Ada, Actor, JavaScript</li></ul>
---------	--

### // 인공지능 개발에 사용되는 언어

인공지능	<ul style="list-style-type: none"><li>• 인공지능 : AI(artificial intelligence)</li><li>• 지능(intelligence)은 '똑같은 실수를 되풀이하지 않는 능력'으로 학습 능력이다.</li><li>• 인공지능 프로그램 작성에 과거 초창기에는 주로 lisp과 prolog가 사용되었다.</li><li>• 즉, lisp은 반세기 동안 인공지능 분야에 가장 널리 사용되었다.</li><li>• 지금은 인공지능 프로그램에 파이썬이 가장 많이 사용되고 있다.(통계자료)</li></ul> <p>// 다음은 인공지능 프로그램에 사용되고 있는 언어이다.(최근 통계 자료)</p> <ul style="list-style-type: none"><li>• 많이 사용되는 상위 5개 : 파이썬, 자바, C/C++, 자바스크립트, R</li><li>• 기타 인공지능에 사용되는 언어 : lisp, prolog, lua, julia, swift</li></ul>
------	---

### // 인공지능 개발에 파이썬이 많이 사용되는 이유

- 인공지능 코딩이 쉽다.(프로그래밍이 쉽고, 단순)
- 다양한 인공지능 라이브러리와 모듈 지원
- 파이썬은 메모리를 프로그래머가 관리하지 않는다. - 힙영역 관리
- 파이썬은 느리지만 인공지능에 사용할 때 상관없다.(인터프리터 언어)
- 파이썬은 데이터 분석 능력이 좋다.

**기출문제 분석**

**1. 폰노이만 컴퓨터 구조에서 프로그래밍 언어에 대한 설명으로 옳은 것은? [2012년 서울 9급]**

- ① 명령형 언어는 비 폰노이만식 기계구조에 기초하여 설계된 언어이다.
- ② 폰노이만 컴퓨터구조에서는 함수를 사용할 수 없다.
- ③ 함수형 언어는 폰노이만 컴퓨터구조에서 실행 속도가 빠르다.
- ④ 폰노이만 컴퓨터구조는 변수에 값을 배정하기 위해서는 배정문을 사용한다.
- ⑤ 함수형 언어는 모든 계산에 함수를 사용하며 폰노이만 컴퓨터구조에 적합하다.

♣ 폰노이만 컴퓨터 구조

- 폰노이만 컴퓨터에서는 프로그램과 데이터가 같은 하나의 기억장소에 저장된다.
- 폰노이만 컴퓨터는 폰노이만이 제시한 컴퓨터 구조로 프로그램 내장 방식이라고도 한다.
- 폰노이만 컴퓨터에서 명령어는 코드 세그먼트, 데이터는 데이터 세그먼트 영역에 저장된다.
- 명령어와 데이터는 파이프라인 방식으로 주기억장소에서 CPU로 전달된다.
- 변수, 배정문, 반복문을 주축으로 하는 **명령어 언어에 적합한 컴퓨터 구조**이다.
- 프로그램 실행의 효율성에서는 반복문 권장한다. 재귀함수 사용은 권장하지 않는다.

// 명령형 언어와 함수형 언어 비교

	명령형 언어(절차적 언어)	함수형 언어
프로그램	일련의 명령문 나열한다.	함수 정의와 함수를 적용한다.
자료구조	여러 가지 종류	리스트(list)
변수	위치가 부여된 메모리 셀	함수로 정의(변수는 사용하지 않는다)
배정문	배정문이 주된 문장	모든 계산에는 함수를 이용
함수	부작용 발생 가능	순수함수는 부작용이 발생하지 않음
제어구조	순차, 조건, 반복 구조	함수 합성
참조투명성	불투명	투명
기계구조	폰 노이만식 기계	주소 개념이 없는 기계에 적절
구문	영문과 대수의 혼합 형태	프로그램 코드와 데이터는 같은 형태

// 비 폰노이만 컴퓨터

- 비 폰노이만 컴퓨터에서는 함수형 언어로 작성된 프로그램이 효율적으로 실행된다.
- 함수형 언어에서는 재귀함수 사용이 권장된다.
- 미래에 비 폰노이만 컴퓨터 사용 시대가 도래되면
- 현실에 적합한 함수형 언어를 이용한 병행처리가 프로그래밍에 적용될 것이다.

## 2. 프로그래밍 패러다임(paradigm)에 대한 설명으로 옳지 않은 것은? [2009년 국가 7급]

- ① 명령형 언어는 폰노이만 구조에 근거하였기 때문에 프로그램이 정확하게 실행된다는 것을 증명하기가 비교적 용이하다.
- ② 논리형 언어는 기호 논리학에 근거하였기 때문에 if문이나 for문과 같은 제어문을 사용하지 않더라도 프로그램이 가능하다.
- ③ 순수 함수형 언어는 변수의 개념을 갖고 있지 않아 배정문을 사용하지 않고도 프로그램이 가능하다.
- ④ 객체지향형 언어는 실세계에서 객체의 상호작용을 반영하기 위한 아이디어로 시작되어 코드의 재사용과 수정 용이성 증진에 매우 효과적인 방법으로 알려져 있다.

### ☞ 프로그래밍 패러다임(paradigm)

- 명령형 언어는 폰노이만 구조에 근거하였기 때문에 프로그램이 정확하게 실행된다는 것을 증명하기가 비교적 용이하다.(x)
  - 명령형 언어의 프로그램이 **정확하게 실행된다는 것을 증명하기가 비교적 어렵다.**
  
- 명령형 언어는 폰노이만 구조의 기계적인 관점에 근거하여 모델링 되었다,
  - 명령형 언어의 프로그램은 폰노이만 구조의 기계에서 효율적으로 실행된다.
  - 명령형 언어는 사람 관점인 실세계의 자연스러운 표현에 한계가 있다.
- 실세계의 자연스러운 표현은 재귀함수를 사용하는 함수형 언어가 더 합리적이다.
  - 실세계는 재귀적인 구조이다.
- 단지, 함수형 언어는 폰노이만 구조의 컴퓨터에서는 실행의 효율성이 낮아서 지금 현재에서 권장되지 않을 뿐이다.
- 미래에 함수형 언어로 작성된 프로그램의 효율적인 실행이 가능한 비 폰노이만 구조의 컴퓨터가 탄생되면 함수형 언어가 명령형 언어를 대체하게 될 것이다.

### // 프로그래밍 언어 패러다임 구분

명령형 언어	C, Algol, Basic, PL/1, Pascal, FORTRAN, COBOL 등
함수형 언어	Lisp, Scheme, ML, Miranda, Haskell, FP, HOPE
논리형 언어	Prolog
객체지향형 언어	Simula67, Smalltalk, C++, Ada95, Java, C#, Python

3. 다음 설명에서 ㉠과 ㉡에 가장 알맞은 것은? [2023년 군무 9급]

㉠은(는) 문제를 여러 개의 작은 함수로 나누어 그 문제를 해결한다.  
반면에, ㉡은(는) 문제를 구성하는 객체들을 만들어  
이들 객체들 간의 메시지 교환 방식으로 문제를 해결한다.

- |              |          |
|--------------|----------|
| ① ㉠절차지향언어    | ㉡객체지향언어  |
| ② ㉠인공지능언어    | ㉡절차지향언어  |
| ③ ㉠시스템지향언어   | ㉡객체지향언어  |
| ④ ㉠인터페이스지향언어 | ㉡시스템지향언어 |

☞ 프로그래밍 언어

절차지향언어	<ul style="list-style-type: none"> <li>· 폰 노이만식 기계구조에 기초하여 설계된 언어이다.</li> <li>· 프로그래밍 단위는 문장이다. 즉, 일련의 명령문 나열이다.</li> <li>· 함수를 이용하여 문제를 해결한다.</li> <li>· 예 : C, Algol, Basic, PL/1, Pascal, FORTRAN, COBOL 등 많은 언어</li> </ul>
객체지향언어	<ul style="list-style-type: none"> <li>· 객체를 지원한다. → 객체는 프로그램의 기본적인 구성 단위</li> <li>· 모든 객체는 클래스에 속한다.</li> <li>· 상속을 지원한다.(Cardelli와 Wagner의 정의) → 클래스는 상속으로 서로 연관</li> <li>· 객체들 사이에 메시지를 교환하는 방식으로 문제를 해결한다.</li> <li>· 예 : Simula67, Smalltalk, C++, Ada95, Java, C#, Eiffel, Object C</li> </ul>

정답 : ①

4. 객체지향 기법을 지원하지 않는 프로그래밍 언어는? [2015년 국가 9급]

- |          |        |
|----------|--------|
| ① LISP   | ② Java |
| ③ Python | ④ C#   |

☞ 언어 계산 전형(Paradigm)

- 객체지향 언어 : Simula67, Smalltalk, C++, Ada95, Java, C#, Python
- 논리형 언어 : Prolog
- 함수형 언어 : Lisp, Common Lisp, Scheme, ML, Miranda, Haskell, FP, HOPE
- 명령형 언어 : C, Algol, Basic, PL/1, Pascal, FORTRAN, COBOL 등 많은 언어

정답 : ①

5. 프로그래밍 언어에 대한 설명으로 옳지 않은 것은? [2016년 군무원 9급 유형]

- ① C++은 명령형 프로그래밍과 객체지향형 프로그래밍 모두 가능하다.
- ② Java는 힙에 할당된 객체가 더 이상 의미가 없을 때, JVM이 메모리 상황에 맞추어 실시간으로 객체들을 회수한다.
- ③ LISP은 인공지능 분야에 널리 사용되고 있는 함수형 언어이며, 함수형 언어로 Scheme, ML, Haskell 등이 있다.
- ④ prolog은 인공지능 분야에 널리 사용되고 있는 언어이며, 프로그램들이 명령어들로 이루어져 있는 명령형 언어이다.

☞ 프로그래밍 언어

---

- prolog은 논리형 언어이다.
- 

정답 : ④

6. 객체지향 프로그래밍(object-oriented programming)과 관련된 다음의 설명 중 가장 옳지 않은 것은? [2022년 군무원 9급]

- ① 클래스 내의 세부 구현에 대한 접근제한을 통해 정보은닉(information hiding) 효과를 달성할 수 있다.
- ② 클래스 간의 상속(inheritance) 기능을 써서 부모클래스의 특성과 기능을 물려받을 수 있으며, 오버라이딩(overriding)을 통하여 일부만을 변경할 수도 있다.
- ③ 다형성(polymorphism)을 활용하면 하나의 변수나 함수의 의미를 상황에 따라 다르게 해석할 수 있게 된다.
- ④ 객체지향 프로그래밍은 실행속도가 빨라지는 장점이 있으나 코드의 재사용성이 절차적 프로그래밍 방식에 비하여 저하하는 단점이 있다.

☞ 객체지향 프로그래밍

---

- 객체지향 프로그래밍은 실행속도가 빨라지는 장점이 있으나 코드의 재사용성이 절차적 프로그래밍 방식에 비하여 저하하는 단점이 있다.(×)
  - 대부분의 객체지향 프로그래밍은 실행속도가 늦지만, 코드 재사용성은 좋다.
- 

정답 : ④



9. 객체지향(object-oriented) 언어인 C++에 대한 설명으로 옳지 않은 것만을 <보기>에서 모두 고르면? [2020년 국회 9급]

-----<보기>-----

- ㄱ. 다중상속(multiple inheritance)이 불가능하다.
- ㄴ. 함수 오버로딩(overloading)이 가능하나 한 클래스(class) 내에서 동일한 이름을 가진 다수의 함수는 정의될 수 없다.
- ㄷ. 클래스를 통해 데이터 캡슐화(encapsulation)와 추상화(abstraction)를 지원한다.
- ㄹ. 다형성(polymorphism)과 상속(inheritance)을 지원한다.
- ㅁ. 다형성(polymorphism)의 지원을 위해 동적 바인딩(dynamic binding)을 사용한다.

- ① ㄱ, ㄴ                      ② ㄱ, ㄴ, ㅁ
- ③ ㄱ, ㄷ, ㄹ                ④ ㄴ, ㄹ, ㅁ
- ⑤ ㄷ, ㄹ, ㅁ

☞ 객체지향 언어 C++

- ㄱ. 다중상속(multiple inheritance)이 불가능하다.(×)  
→ C++은 다중상속을 지원한다.
- ㄴ. 함수 오버로딩(overloading)이 가능하나 한 클래스(class) 내에서 동일한 이름을 가진 다수의 함수는 정의될 수 없다.(×)  
→ 한 클래스(class) 내에서 동일한 이름을 가진 다수의 함수가 정의될 수 있다.

정답 : ①

10. 프로그래밍 언어에 대한 설명으로 옳지 않은 것은? [2016년 계리]

- ① Objective-C, Java, C#은 객체지향언어이다.
- ② Python은 정적타입을 지원하는 컴파일러 방식의 언어이다.
- ③ ASP, JSP, PHP는 서버 측에서 실행되는 스크립트 언어이다.
- ④ XML은 전자문서를 표현하는 확장 가능한 표준 마크업 언어이다.

☞ Python

- Python은 정적타입을 지원하는 컴파일러 방식의 언어이다.(×)  
→ Python은 동적타입검사를 수행하는 프로그래밍 언어이고, 인터프리터 언어이다.

정답 : ②

11. 객체지향 언어에 대한 설명으로 옳지 않은 것은? [2017년 국회 9급]

- ① 객체는 재사용이 가능하나 클래스는 불가능하다.
- ② 객체는 하나의 클래스 타입으로 선언된다.
- ③ 오버라이딩(overriding), 오버로딩(overloading) 모두 다형성을 지원한다.
- ④ 추상메소드는 선언만 하고 그 내용은 기술하지 않은 메소드이다.
- ⑤ Smalltalk, Simula, Java 언어는 객체지향 개념을 잘 표현하고 있다.

☞ 객체지향 언어

- 
- 객체는 재사용이 가능하나 클래스는 불가능하다.(×)
    - 클래스도 재사용이 가능하다.
    - 한 번 작성된 클래스를 이용하여 여러 객체를 정의할 수 있다.(객체 생성)
- 

정답 : ①

12. 다음에서 설명하는 객체지향 프로그래밍의 특징은? [2019년 지방 9급]

- 
- 객체를 구성하는 속성과 메서드가 하나로 묶여 있다.
  - 객체의 외부와 내부를 분리하여 외부 모습은 추상적인 내용으로 보여준다.
  - 객체 내의 정보를 외부로부터 숨길 수도 있고, 외부에 보이게 할 수도 있다.
  - 객체 내부의 세부 동작을 모르더라도 객체의 메서드를 통해 객체의 기능을 활용할 수 있다.
- 

- ① 구조성                      ② 다형성
- ③ 상속성                      ④ 캡슐화

☞ 객체지향 프로그래밍 - 캡슐화

- 
- 캡슐화의 주 목적은 외부에 인터페이스는 공개하고, 구현은 숨기는 것이다.(정보은닉, 블랙박스)
  - 인터페이스(interface)는 서로 다른 두 시스템 사이에서 정보를 주고받는 상호작용 접점이다.
  - 구현(implement)은 자세하게 기술된 원시코드이다.(원시코드는 외부에 숨긴다)
  - 외부에 정보를 숨기면, 외부에 영향을 주지 않고 객체 내부의 구현을 변경할 수 있다.
- 

정답 : ④