





컴퓨터 과학과 기계학습을 활용한 인공지능이 발전함에 따라서 이전에는 상상만 가능했던 다양한 서비스들이 개발되고 여러 가지 분야에 적용되고 있다.

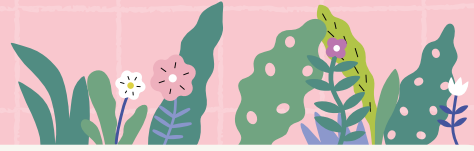
예를 들어, 상상하지도 못했던 세밀한 동영상과 이미지를 몇 번의 요청만으로 생성할 수 있고, 때로는 인간인지 아닌지 구분하기 어려운 수준의 상담 서비스와 대화하는 경험을 하기도 한다.

한편에서는 기계학습과 인공지능을 활용한 새로운 웹 서비스, PC 응용 프로그램, 스마트폰 앱을 설계·구현하여 이전에는 생각하지 못했던 막대한 수익을 창출하는가 하면, 다른 한편에서는 컴퓨터를 통해 생성한 데이터와 계산으로 만들어진 기계학습의 결과를 인간처럼 생각하는 진짜 지능이라 상상하거나 맹목적인 믿음을 가지기도 한다.

1980년대 개인용 컴퓨터가 일반화되기 시작하면서부터 시작된 컴퓨터 프로그래밍에 대한 관심과 교육은 수십여 년이 지난 지금도 계속되고 있다. 하지만, 프로그래밍을 가르치고 배우는 방법, 내용, 목표는 컴퓨터와 네트워크의 발전·보급, 다양한 스마트 기기의 개발과 확산, 전세계적인 감염병의 유행 등 시대적인 상황과 사회적 필요, 요구에 따라 조금씩 바뀌어 왔다고 할 수 있다.

앞으로 다가올 세상에 우리 중 누군가가 만든 새로운 것들이 사회의 모습을 바꾸게 되더라도 수천 년 동안 전해 내려오는 기초 지식과 방법을 후세에게 가르치고 배우는 것이 크게 변하지 않고 있는 것처럼, 컴퓨터와 로봇과 인공지능을 우리 생각과 의지에 따라 움직일 수 있는 가장 기본적인 방법인 프로그래밍을 가르치고 배우는 것은 컴퓨터와 네트워크를 기반으로 폭발적으로 발전하는 인류 역사 문화와 함께 지속될 것이다.

본 교과서는 급격하게 변화하는 시대적 요구와 상황에 발맞추어 프로그래밍 언어 활용에 필요한 기초와 지식을 빨리 효과적으로 학습할 수 있도록, 교육과정 내용을 충실히 반영하였다. 또한, 간결하고 핵심적인 예시 코드와 자기 주도적 학습이 즉시 가능한 코드 작성 활동을 중심으로 다음과 같이 3개 단원으로 구성하였다.



### I 프로그래밍의 개요

- 프로그래밍 언어의 종류와 특징을 이해하고, 개발 분야와 용도에 따라 적절한 프로그래밍 언어를 선택할 수 있는 지식을 습득한다.
- 프로그램 개발을 위한 절차와 단계를 이해하고, 알고리즘의 개념과 알고리즘을 표현하는 방법을 학습한다.

### II 프로그래밍의 기초

- 데이터의 종류에 따른 자료형을 이해하고, 변수를 사용하여 데이터를 입력하고 출력하는 방법을 학습한다.
- 데이터를 입력받아 산술, 비교, 논리 연산을 수행하며, 문제 상황에 적합한 순차, 선택, 반복 구조를 사용하는 방법을 학습한다.
- 원하는 작업을 수행할 수 있는 함수를 설계하여 사용하는 방법과 데이터를 파일로 입출력하여 사용하는 방법을 학습한다.

### III 프로그램의 설계와 구현

- 프로그램을 기획, 설계, 구현, 테스트하는 실제적인 과정과 방법을 학습하고 경험한다.
- 여러 가지 프로그램 개발 프로젝트를 프로그램 개발 단계에 따라 수행하며 실무 중심의 프로그래밍 능력을 함양한다.

이 과정을 통해 프로그래밍 언어를 효과적으로 활용하여 자신의 목적에 맞는 프로그램을 설계하고 구현할 수 있는 능력을 함양할 수 있기를 바란다.

- 학생들과 함께 배우고 공부하며 나아가는 선생님 일동 -

# 구성과 특징

## Q 도입

### 1 프로그래밍의 개요

1 프로그래밍의 의미  
2 프로그래밍의 과정과 알고리즘  
3 프로그래밍 개발 환경

**이 단원을 배운다면**

- 프로그래밍이란 무엇인가?
- 프로그래밍의 과정과 알고리즘을 설명할 수 있는가?
- 프로그래밍 개발 환경을 설명할 수 있는가?

**이 단원을 배울 때 주의할 점**

- 프로그래밍의 개념을 이해하는 데 중점을 두고, 프로그래밍의 과정과 알고리즘을 설명할 수 있는가?
- 프로그래밍 개발 환경을 설명할 수 있는가?

### 2 프로그래밍의 절차와 알고리즘

**Think**

프로그래밍이란 무엇인가? 프로그래밍의 과정과 알고리즘을 설명할 수 있는가? 프로그래밍 개발 환경을 설명할 수 있는가?

**중단원 프로젝트 활동**

이 단원에서 배운 내용을 바탕으로 프로그래밍을 이용하여 간단한 게임을 만들어 보자.

```

1 #include <stdio.h>
2 int main()
3 {
4     printf("안녕하세요!\n");
5     return 0;
6 }
    
```

**중단원 프로젝트 활동**

이 단원에서 배운 내용을 바탕으로 프로그래밍을 이용하여 간단한 게임을 만들어 보자.

### 개발 환경

**02 공통 개발 환경**

프로그래밍을 위한 개발 환경을 설정하는 방법을 알아보자.

**요약 정리**

프로그래밍을 위한 개발 환경을 설정하는 방법을 알아보자.

**동형 개발 환경**

프로그래밍을 위한 개발 환경을 설정하는 방법을 알아보자.

**자기 평가**

이 단원을 배운 후 자신의 학습 성과를 평가해보자.

## 대단원

학습 요소에 따른 성취 기준과 맞추어 대단원에서 배우게 될 학습 내용(이 단원에서는), 학습 과정(단원을 시작하며), 학습 결과(단원을 배우고 나면)로 나누어 학습 목표와 방향을 제시하였습니다.

## 중단원 Think

앞으로 배울 학습 요소와 목표 전체를 아우르는 내용이 나 이론 등을 그림, 사진으로 보여줍니다. 해당 학습 요소를 배우고 이해해야 하는 이유와 필요, 유의점 등을 알 수 있습니다.

공부할 내용에 관한 궁금증과 흥미를 느끼고, 생각할 수 있도록 질문으로 마무리하였습니다.

## 중단원 프로젝트 활동

중단원 마지막에는 학생들의 흥미와 관심을 이끌어낼 수 있는 프로그래밍 프로젝트를 제시하였습니다.

## 소단원 도입

수업을 통해 배워야 할 학습 목표와 학습 개념을 먼저 제시하고, 학습 내용을 이해하는 데 도움이 될 수 있는 그림과 보충 자료를 충실히 제시하였습니다.

## 요약 정리와 자기평가

요약 정리와 자기평가를 통해 배운 내용을 정리하고 스스로 평가할 수 있도록 하였습니다.

## Q 전개

### 19세기에 컴퓨터와 프로그램을 만든 사람들

영국의 수학자 찰스 배비지(Charles Babbage)와 영국의 과학자이자 수학자 앨버트 뉴턴 애딩턴(Ada Lovelace)은 현대 컴퓨터 과학의 기초를 마련한 인물로 평가받는다. 애딩턴은 '기계 노트'라는 것을 통해 배비지의 기계에 대한 아이디어를 구체화하고, 최초의 프로그래밍 언어인 '퍼시어스'를 발명했다.

1837년, 찰스 배비지는 '분석기'라는 기계를 발명했다. 이 기계는 현대 컴퓨터와 유사한 구조를 가지고 있었다. 이 기계는 수학적 계산을 할 수 있었고, 프로그램에 따라 다른 작업을 할 수 있었다. 이 기계는 '기계 노트'라는 것을 통해 프로그램이 실행되는 방식을 나타내었다.

애딩턴은 '기계 노트'라는 것을 통해 배비지의 기계에 대한 아이디어를 구체화하고, 최초의 프로그래밍 언어인 '퍼시어스'를 발명했다. 이 언어는 현대 프로그래밍 언어의 기초를 마련했다.

## 본문

프로그래밍 언어 기초를 효과적으로 학습하기 위해 **“기본 형식”** 및 기초 설명 → **“먼저 해 보기”** 코드 → 코드 작성 **“활동”** 순으로 구성하였습니다.

프로그래밍 기초와 원리 학습을 하고 실제 예시 코드를 통해 기초와 원리를 이해한 다음, 이를 바탕으로 코드를 작성하는 활동을 하면서 자기주도 학습을 할 수 있도록 제시하였습니다.

## 읽기 자료

학습 내용의 이해를 돕고 흥미를 유발할 수 있도록 프로그래밍과 관련한 역사적 사실, 소프트웨어/도구/플랫폼 설명, 특별한 프로그램 코드 등 다양한 보충 자료를 함께 제시하였습니다.

### 01 비반도 측정

비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**문제 제시**

비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**문제 해결**

비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**답변 요점**

비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**활동 요점**

비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**읽기 자료**

구분	비반도	비반도
비반도	100	100
비반도	100	100
비반도	100	100

## 프로그래밍 개발 프로젝트

실제 프로그램 개발 과정에서 활용할 수 있는 순서와 형태의 다양한 프로젝트형 활동 과정을 제시하였습니다.

## Q 마무리

**11** 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**12** 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**13** 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

**14** 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다. 비반도를 측정하는 것은 매우 중요한 작업입니다.

## 대단원 마무리

학생 스스로 배운 내용을 정리하고 평가하며 이해한 것을 확인해 볼 수 있도록, 선택형, 서술형 문제를 다양하게 제시하였습니다.





# I

## 프로그래밍의 개요

### 1 프로그래밍 언어

01 프로그래밍 언어의 개념 .....	13
02 프로그래밍 언어의 종류와 특징 .....	16

### 2 프로그래밍의 절차와 알고리즘

01 프로그래밍의 절차 .....	23
02 알고리즘 .....	25

### 3 프로그램 개발 환경

01 통합 개발 환경 .....	31
02 공동 개발 환경 .....	34
■ 중단원 프로젝트 활동 .....	37
■ 대단원 마무리 .....	38





# II

## 프로그래밍의 기초

### 1 데이터 입출력

01 데이터 출력 .....	43
02 데이터 입력 .....	52
■ 중단원 프로젝트 활동 .....	67

### 2 데이터 연산

01 산술 연산 .....	69
02 비교 연산 .....	74
03 논리 연산 .....	79
04 비트 단위 연산 .....	84
■ 중단원 프로젝트 활동 .....	94

### 3 제어 구조

01 선택 구조 .....	99
02 반복 구조 .....	109
■ 중단원 프로젝트 활동 .....	123

### 4 함수

01 함수의 종류와 사용 .....	127
02 함수의 설계 .....	132
■ 중단원 프로젝트 활동 .....	146

### 5 파일 처리

01 파일의 개념 .....	149
02 파일의 입력과 출력 .....	155
■ 중단원 프로젝트 활동 .....	166
■ 대단원 마무리 .....	168



# 차 례

## III

### 프로그램의 설계와 구현

#### 1 프로그램 개발 과정

01 프로그램 기획 .....	173
02 프로그램 설계 .....	180
03 프로그램 구현 .....	185
04 프로그램 테스트 .....	187

#### 2 프로그램 개발 프로젝트

01 비만도 측정 .....	192
02 일차 방정식 $ax \pm b=c$ 의 해 구하기 .....	197
03 이달은 며칠까지 있을까? .....	201
04 철근 자르기 .....	205
05 바코드 검증 .....	209
06 암호 해독하기 .....	214
07 파이( $\pi$ )값 근사하기 .....	218
08 종이 자르기 .....	224





<b>09</b> 복면산 문제 .....	229
<b>10</b> 원의 겹침 여부 판별하기 .....	233
<b>11</b> 철광석 제련 .....	237
<b>12</b> 3등 찾기 .....	241
<b>13</b> 도서 대출 시스템 .....	245
<b>14</b> 교집합과 합집합 .....	251
■ 이 책의 정답 .....	257
■ 찾아보기 .....	268
■ 이미지 및 내용 출처 .....	271



# I

# 프로그래밍의 개요

- 01 프로그래밍 언어
- 02 프로그래밍의 절차와 알고리즘
- 03 프로그램 개발 환경





### 이 단원에서는

프로그래밍 언어의 종류와 특징을 이해하고, 개발 분야와 용도에 따라 적절한 프로그래밍 언어를 선택할 수 있는 지식을 습득한다. 또한 프로그램 개발을 위한 절차와 단계를 체계적으로 표현하는 방법을 살펴본다.

주어진 문제를 해결하기 위한 알고리즘의 개념과 표현 방법을 학습하고, 프로그래밍 개발 환경을 이용하여 해당 문제를 협업 과정으로 해결한다.

### 단원을 시작하며

- ✓ 프로그래밍 언어의 다양한 종류와 특징을 알아보고, 프로그래밍의 절차와 유의 사항, 알고리즘의 이해와 표현 방법을 학습한다.
- ✓ 협업을 위한 프로그래밍 개발 환경과 협업 방법의 학습 등을 통하여 관련 직무를 효율적으로 수행하는 능력을 향상시킨다.

### 단원을 배우고 나면

- ✓ 다양한 프로그래밍 언어의 특성을 이해하고, 용도에 따라 적합한 언어를 선택할 수 있다.
- ✓ 체계적인 프로그램 개발 절차와 유의 사항을 익히고 설명할 수 있다.
- ✓ 알고리즘의 표현 방법을 이해하고, 문제 해결을 위한 알고리즘을 작성할 수 있다.
- ✓ 디지털 기술을 활용한 프로그래밍 공동 개발 환경에서 협업을 통한 경험을 쌓을 수 있다.



# 프로그래밍 언어



Think

프로그래밍 언어는 컴퓨터 프로그램을 만들 때 사용하는 언어이다. 컴퓨터를 이용해 문제를 해결하기 위한 바탕이 되며, 해결하려는 문제의 특성에 따라 사용되는 언어가 달라질 수 있다.

컴퓨터 프로그램을 만들 때 사용하는 프로그래밍 언어에는 여러 가지가 있으므로 프로그램의 목적과 환경, 작성 방법 등에 따라 적절한 프로그래밍 언어를 선택하는 일은 매우 중요하다.

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		Python	12.46%	+0.26%
2	2		C	12.37%	+0.46%
3	4	▲	C++	11.36%	+1.73%
4	3	▼	Java	11.28%	+0.81%
5	5		C#	6.71%	+0.59%
6	6		Visual Basic	3.34%	-2.08%
7	7		JavaScript	2.82%	+0.73%
8	13	▲	PHP	1.74%	+0.49%
9	8	▼	SQL	1.47%	-0.47%
10	9	▼	Assembly language	1.29%	-0.56%
11	12	▲	Delphi/Object Pascal	1.26%	-0.07%

▲ 티오베 지수(Tiobe Index)



어떤 프로그래밍 언어를 가장 많이 사용하고 있을까?

# 01

## 프로그래밍 언어의 개념

### ▶ 학습 목표

- 프로그래밍 언어의 개념을 설명할 수 있다.

### ▶ 학습 개념

- 프로그래밍 언어의 개념



### 1 프로그래밍과 프로그래밍 언어

컴퓨터 프로그램을 만드는 작업을 프로그래밍이라고 하고, 프로그래밍에 사용하는 언어를 프로그래밍 언어라고 한다. 사람이 컴퓨터를 활용하여 원하는 작업을 실행할 때 프로그래밍 언어는 프로그램 작성의 수단이 된다.

사람이 사용하는 자연어는 맥락적 의미, 모습과 소리 관련 표현, 비논리적 내용 등을 포함하고 있어 특별한 형식을 갖추지 않은 비형식적 상태로도 사용할 수 있다. 하지만 컴퓨터를 통하여 실행하는 프로그래밍 언어는 제한된 개수의 기호와 단어를 사용하므로 정확한 형식에 맞게 논리적으로 작성해야 한다.

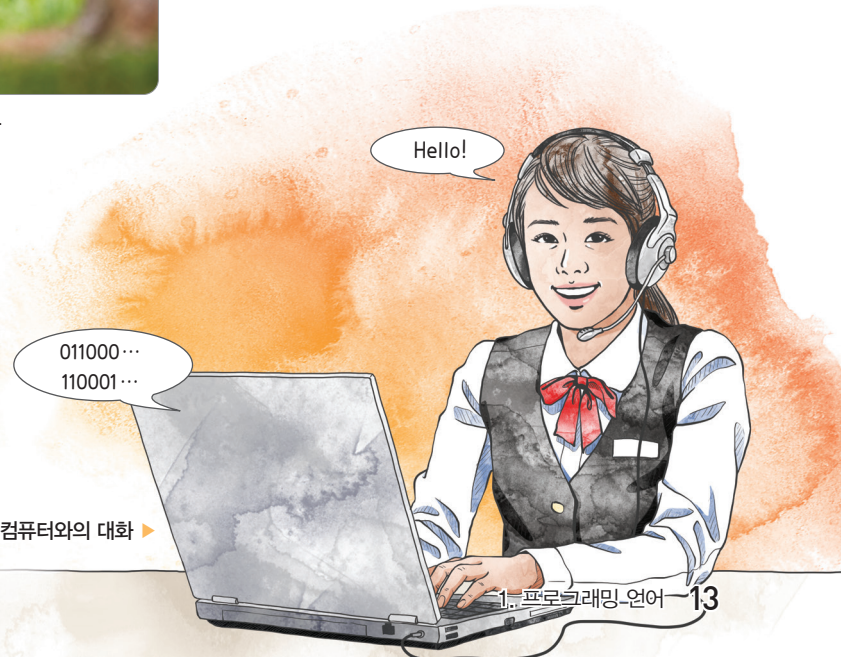


'언어'는 자기 생각을 다른 사람들에게 나타내고 전달하기 위하여 사용하는 말과 글이다. 따라서 '프로그래밍 언어'는 컴퓨터가 실행해야 할 작업을 표현하는 특별한 언어라고 할 수 있다.



▲ [그림 I-1] 사람과의 대화

프로그램은 사람이 원하는 작업을 컴퓨터를 통하여 수행하도록 명확하고 구체적인 실행 명령을 순서대로 나열한 것으로, 이렇게 작성된 '컴퓨터 프로그램'을 간단히 줄여 '프로그램'이라고 부른다.

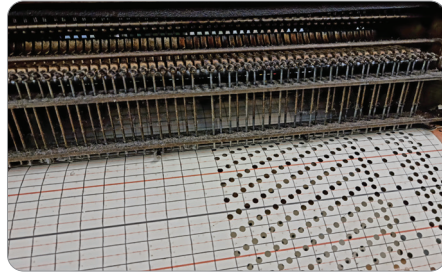


[그림 I-2] 컴퓨터와의 대화 ▶

## ■ 방직 기계

실을 뽑아서 천을 만드는 기계

컴퓨터 발명 이전, 방직 기계에서 특별한 무늬가 있는 직물을 출력하기 위하여 사용한 펀치 카드나 오르골에서 반복해서 연주하는 멜로디를 작동시키기 위한 원통 모양 부품은 간단한 형태의 프로그램으로 볼 수 있다.



▲ [그림 1-3] 펀치 카드



▲ [그림 1-4] 오르골

컴퓨터가 발명되고 프로그램도 점차 발전하면서 컴퓨터를 사용하는 목적과 방법에 따라 다양한 프로그래밍 방법과 여러 가지 프로그래밍 언어가 개발되어 왔다.

요즘에는 빠르고 편리하게 프로그램을 작성하여 실행시킬 수 있을 뿐만 아니라, 여러 가지 형태의 다양한 컴퓨터 시스템에서도 동일하게 작동할 수 있는 프로그래밍 언어와 도구들로 발전하고 있다.

## 2 절차적 프로그래밍과 객체 지향 프로그래밍

프로그래밍 방식은 특징에 따라 여러 가지 방법으로 분류할 수 있는데, 절차적 프로그래밍과 객체 지향 프로그래밍으로 구분할 수도 있다.

### (1) 절차적 프로그래밍

절차적 프로그래밍은 실행해야 할 명령이나 명령이 포함된 함수를 순서대로 불러 실행하는 형태로, 프로그램 코드를 설계하고 작성하는 방법이라고 할 수 있다. 절차적 프로그래밍에서는 변수, 데이터 구조, 함수 등으로 나누어 프로그램을 작성하는 것이 중심이 된다.

이러한 형태의 코드 설계 및 작성 방법은 1950년대부터 1970년대에 걸쳐 만들기 시작한 Fortran, ALGOL, COBOL, PL/I, BASIC, Pascal, C에서 지원된다.

### (2) 객체 지향 프로그래밍

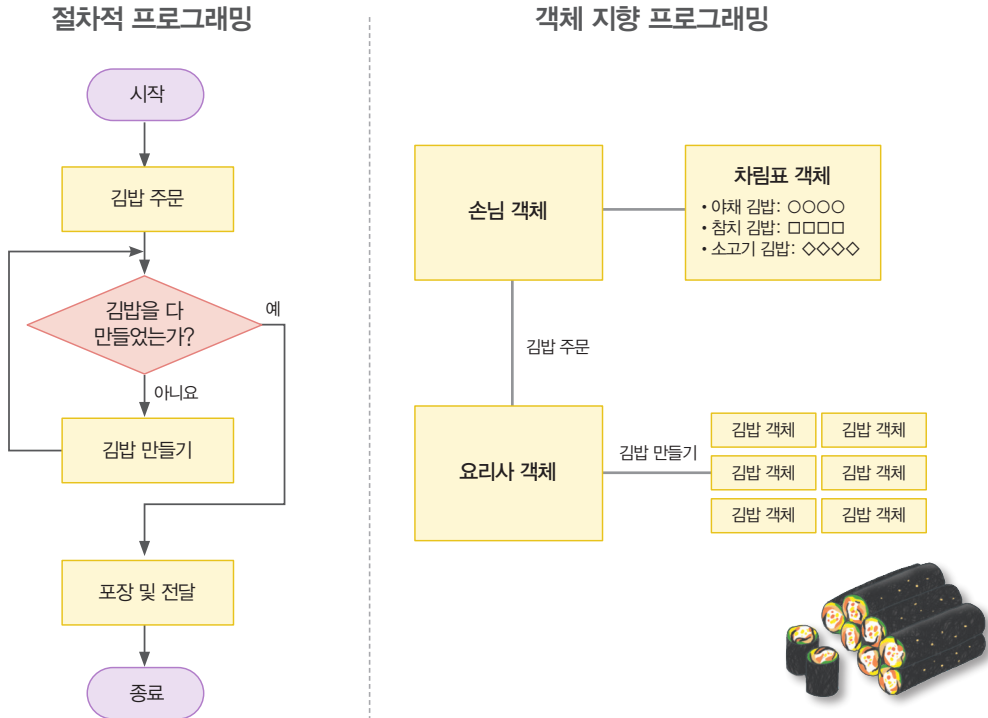
객체 지향 프로그래밍은 어떤 데이터와 그 데이터에 접근할 수 있는 코드를 포함하는 객체를 중심으로 프로그램의 코드를 설계하고 작성하는 방법이다. 객체들 사이의 상호 작용과 변화를 중심으로 프로그램 설계와 작성이 이루어진다.

이러한 형태의 코드 설계 및 작성 방법은 기존의 여러 프로그래밍 언어에 추가되기도



프로그램을 만드는 코드를 소스 코드라고 부른다. 프로그램을 만드는 재료지만, 일반 언어와는 다르게 암호처럼 작성하기 때문이다.

하였으며, C++, C#, Java, JavaScript, Python, R 등과 같은 최신 프로그래밍 언어에서 대부분 지원된다.



▲ [그림 I-5] 절차적 프로그래밍과 객체 지향 프로그래밍의 비교

### 하나 더 알기

## 소프트웨어 위기(software crisis)

‘소프트웨어 위기’는 네덜란드의 유명한 컴퓨터 과학자인 다익스트라(Edsger Dijkstra)가 1972년도에 처음 이야기한 용어이다.

컴퓨터 성능의 비약적 발전과 함께 수많은 컴퓨터가 사용되었지만, 필요한 프로그램이 다양해지고 프로그램으로 다루어야 할 문제들이 복잡해졌다. 이러한 배경에서 필요한 프로그램을 원하는 시간 내에 만들 수 없게 되자 ‘소프트웨어 위기’라는 말이 등장하였다.

기존의 절차적 프로그래밍 코드들은 다른 목적의 프로그램을 작성할 때 코드의 재사용이 매우 어려웠다. 하지만 캡슐화, 상속, 다형성 등의 특징을 가진 객체 지향 프로그래밍 코드들은 다른 목적의 프로그램을 위하여 재사용하는 것이 비교적 편리하였으므로 짧은 시간 동안 다양한 목적의 많은 프로그램을 만들 때 효과적으로 재사용되었다.

▲ [그림 I-6] 다익스트라

**소프트웨어**  
어떤 특별한 목적을 위해서 사용하는 여러 개의 프로그램과 그 프로그램들이 사용하는 데이터를 모두 합한 것

구조나 성능을 바꿀 수 없는 하드웨어와 대비하여 구조나 성능을 개선할 수 있는 프로그램을 소프트웨어라고 부르기도 한다.

# 02

## 프로그래밍 언어의 종류와 특징

### ▶ 학습 목표

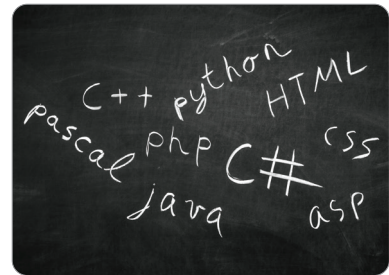
- 각종 프로그래밍 언어의 종류와 특징을 설명할 수 있다.
- 개발 분야와 용도에 따라 적절한 프로그래밍 언어를 선택할 수 있다.

### ▶ 학습 개념

- 프로그래밍 언어의 종류와 특징



기존의 프로그래밍 언어들은 최신 기술 및 새로운 요구 사항을 반영하여 지속해서 개선되며 발전하고 있을 뿐만 아니라, 새로운 프로그래밍 언어들도 꾸준히 개발되어 사용되고 있다. 지금까지 가장 많이 다루는 대표적 컴퓨터 프로그래밍 언어로는 C, C++, C#, Java, Python 등이 있다.



### 1 C

C 언어는 1970년대 데니스 리치(Dennis Ritchie)가 개발한 프로그래밍 언어이다. 포인터를 사용하여 컴퓨터 하드웨어 메모리에 직접 접근할 뿐만 아니라 함수, 데이터형, 구조체 등의 편리한 기능을 제공하면서도 생성된 프로그램의 실행 속도가 빠르다. 이러한 이유로 빠른 실행 및 처리 속도가 필요한 운영 체제 개발이나 하드웨어를 작동시키는 드라이버 프로그램 개발, 다양한 목적의 프로그램 개발에 광범위하게 사용되었다.

C 언어는 이후 많은 프로그래밍 언어 생성에 직간접적으로 큰 영향을 끼쳤다. 1989년에 처음 국제 표준인 C89가 지정되기 시작하여 2018년에 C17까지 지속해서 국제 표준이 만들어졌다.

#### ▼ [표 I-1] C의 특징

장단점	주요 활용 분야
<b>〈장점〉</b> ① 코드가 간결하다. ② 프로그램 실행 속도가 빠르다. ③ 하드웨어를 직접 제어할 수 있다.	① 운영 체제 개발 ② 하드웨어 드라이버 개발 ③ 컴파일러 및 인터프리터 개발 ④ 각종 라이브러리 개발 ⑤ 임베디드 시스템 개발
<b>〈단점〉</b> ① 특정 프로그램 코드를 다른 목적으로 재사용하기 어렵다. ② 메모리에 직접 접근하거나 하드웨어를 직접 제어하는 프로그램을 개발할 때는 주의가 필요하다. ③ 절차적 프로그래밍만 가능하다.	

## 2 C++

C++ 언어는 비야네 스트루스트룹(Bjarne Stroustrup)이 개발한 프로그래밍 언어로, 1985 년도에 처음 공개되었다. C 언어의 특징을 모두 포함하면서도 클래스, 상속, 다형성 등과 같은 객체 지향 프로그래밍이 가능하도록 확장성을 띠며 만들어졌다.

C++ 언어는 시스템 프로그래밍, 임베디드 프로그래밍과 같은 대규모 시스템을 편리하게 프로그래밍할 수 있도록 설계되어 데스크톱 응용 프로그램, 비디오 게임, 서버 프로그램 등에도 효과적으로 사용되었다. 1989년 처음 국제 표준인 C++89가 지정되기 시작하여 2020년 C++20까지 지속해서 국제 표준이 만들어졌다.

### ▼ [표 I-2] C++의 특징

장단점	주요 활용 분야
<p><b>&lt;장점&gt;</b></p> <ul style="list-style-type: none"> <li>① C 언어를 그대로 사용할 수 있다.</li> <li>② 객체 지향 프로그래밍 기능이 추가되었다.</li> <li>③ 템플릿, 객체, 연산자 오버로딩(overloading), 다형성, 예외 처리 기능 등을 사용하여 편리하게 프로그램을 개발할 수 있다.</li> <li>④ 더 많은 표준 라이브러리와 표준화된 STL 라이브러리를 사용할 수 있다.</li> </ul> <p><b>&lt;단점&gt;</b></p> <ul style="list-style-type: none"> <li>① C 언어에 대한 기본적인 이해가 필요하다.</li> <li>② 프로그램 코드가 더 복잡해졌다.</li> <li>③ 프로그램 컴파일 시간이 더 오래 걸린다.</li> <li>④ 메모리 가비지 컬렉션 기능이 없다.</li> </ul>	<ul style="list-style-type: none"> <li>① 데스크톱 응용 프로그램 개발</li> <li>② 고성능 응용 프로그램 개발</li> <li>③ 비디오 게임 개발</li> <li>④ 웹 서버, 검색 엔진, 데이터베이스 서버 개발</li> </ul>

## 3 C#

C# 언어는 마이크로소프트의 아네르스 하일스베르(Anders Hejlsberg)가 2000년에 처음 개발한 언어이다. 윈도 운영 체제에서 작동하는 .NET(프로그램을 개발하는 소프트웨어 프레임워크)의 일부로 다양한 플랫폼에서 실행되는 응용 프로그램 개발에 사용한다.

다양한 목적으로 사용할 수 있는 객체 지향 프로그래밍 언어로 설계되었으며, 자동 가비지 컬렉션, 분산 처리 환경 지원, 대규모 시스템뿐만 아니라 임베디드 시스템용 응용 프로그램 개발에 적합하다. 2023년에는 C# 12.0이 .NET 8.0에 포함되었다.



### ■ 클래스

객체의 구조와 행동을 정의하는 설계도

### ■ 상속

클래스가 다른 클래스의 특성을 계승하여 확장하는 기능

### ■ 다형성

동일한 함수 호출에 주어진 정보를 바탕으로 다른 형태의 작업을 수행하는 특성



표준 템플릿 라이브러리(STL: Standard Template Library)는 1993년 알렉산더 스테파노프(Alexander Stepanov)가 설계한 것이다. 알고리즘, 컨테이너, 함수, 반복자의 네 가지 요소를 편리하게 사용할 수 있다.

### ■ 가비지 컬렉션

#### (Garbage Collection)

필요 없게 된 메모리 영역을 주기적으로 삭제하는 메모리 관리 기법

▼ [표 I-3] C#의 특징

장단점	주요 활용 분야
<p><b>&lt;장점&gt;</b></p> <ul style="list-style-type: none"> <li>① 기본 문법이 C/C++과 비슷하다.</li> <li>② 임의의 자료형인 var형을 지원한다.</li> <li>③ 데이터형 변환을 명확하게 작성해야 하므로 안정적이다.</li> <li>④ 메모리 가비지 컬렉션 기능을 자동으로 수행한다.</li> <li>⑤ 데이터베이스용 SQL 질의어를 사용할 수 있다.</li> <li>⑥ .NET에 포함된 다양한 라이브러리를 사용할 수 있다.</li> </ul> <p><b>&lt;단점&gt;</b></p> <ul style="list-style-type: none"> <li>① C/C++과 달리 전처리기의 사용이 제한적이다.</li> <li>② 전역 함수나 전역 변수가 허용되지 않는다.</li> <li>③ 배열과 포인터를 정의하는 문법이 C/C++과 다르다.</li> <li>④ 마이크로소프트의 시스템과 .NET에 종속되는 경향이 크다.</li> </ul>	<ul style="list-style-type: none"> <li>① 마이크로소프트 윈도우 플랫폼용 응용 프로그램 개발</li> <li>② 게임 개발</li> </ul>

## 4 Java

자바(Java) 언어는 썬의 자바 플랫폼에서 작동하는 여러 프로그램을 위한 언어로, 썬 마이크로시스템의 제임스 고슬링(James Gosling)이 1995년부터 개발하였다. 플랫폼이나 하드웨어에 상관없이 자바 가상 머신을 설치한 모든 시스템에서 실행되도록 설계하여 웹 응용 프로그램 및 모바일 앱 개발에 널리 사용하고 있다. 2023년에는 Java SE 21 (LTS) 버전이 배포되었다.



앱(App)은 일반적으로 스마트폰 기기에서 실행하는 응용(Application) 프로그램을 말하는데, 데스크톱 응용 프로그램을 앱이라고 부르기도 한다.

▼ [표 I-4] Java의 특징

장단점	주요 활용 분야
<p><b>&lt;장점&gt;</b></p> <ul style="list-style-type: none"> <li>① 기본 문법이 C++과 비슷하다.</li> <li>② 메모리 가비지 컬렉션 기능을 자동으로 수행한다.</li> <li>③ 한 번 생성된 프로그램은 특별한 작업 없이 자바 가상 머신을 설치한 모든 시스템에서 플랫폼이나 하드웨어와 관계없이 실행된다.</li> <li>④ 데스크톱 응용 프로그램뿐만 아니라 웹 서비스와 모바일 앱 등에서도 다양한 목적으로 사용된다.</li> </ul> <p><b>&lt;단점&gt;</b></p> <ul style="list-style-type: none"> <li>① C/C++보다 하드웨어 제어 기능이 부족하다.</li> <li>② 프로그램을 실행하기 위해서는 자바 실행 환경이 추가로 필요하다.</li> <li>③ 자바 실행 환경을 통하여 실행되므로 실행 속도가 느리고 많은 메모리가 필요하다.</li> </ul>	<ul style="list-style-type: none"> <li>① 모바일 앱 개발</li> <li>② 웹용 응용 서비스 개발</li> <li>③ 플랫폼 및 시스템 독립형 응용 프로그램 개발</li> </ul>

## 5 Python

파이썬(Python) 언어는 들여쓰기를 활용해 코드 가독성을 높인 고급 범용 프로그래밍 언어로, 귀도 반 로섬(Guido van Rossum)이 1980년대 후반부터 개발하였다. 동적 데이터형과 메모리 가비지 컬렉션 기능이 지원될 뿐만 아니라 대부분의 프로그래밍 패러다임을 지원하는 고급 프로그래밍 언어로 설계되었다. 수십만 개 이상의 파이썬 패키지의 지원을 통하여 데이터 처리 및 기계 학습 분야에서 널리 사용 중이며, 2023년도에는 3.12 버전이 배포되었다.

### ▼ [표 I-5] Python의 특징

장단점	주요 활용 분야
<p><b>〈장점〉</b></p> <ul style="list-style-type: none"> <li>① 들여쓰기를 활용한 코드 블록 작성으로 코드 가독성이 높다.</li> <li>② 프로그램 실행 중 데이터형을 바꿀 수 있는 동적 데이터형을 사용한다.</li> <li>③ 메모리 가비지 컬렉션 기능을 자동으로 수행한다.</li> <li>④ 모듈화된 패키지 소프트웨어를 추가하여 사용할 수 있으므로 확장성이 뛰어나다.</li> <li>⑤ 데스크톱 응용 프로그램뿐만 아니라 웹 서비스, 데이터 처리, 이미지 처리, 기계 학습, 게임 개발 등 다양한 목적으로 사용된다.</li> </ul> <p><b>〈단점〉</b></p> <ul style="list-style-type: none"> <li>① 한 줄씩 해석되어 실행하는 인터프리터 방식으로, 프로그램 실행 중 자료형 타입 오류 등 런 타임(run time) 오류가 발생할 수 있다.</li> <li>② 프로그램 실행 시간이 비교적 느리다.</li> </ul>	<ul style="list-style-type: none"> <li>① 웹 서비스 및 웹 응용 프로그램 개발</li> <li>② 과학 및 수학 데이터 처리 및 분석</li> <li>③ 데이터 시각화 및 이미지 처리</li> <li>④ 기계 학습 및 인공지능 개발</li> <li>⑤ 게임 개발</li> </ul>



**하나 더 알기**

프로그래밍 언어별 객체 지향 프로그래밍 지원

구분	C	C++	C#	Java	Python
절차적 프로그래밍	가능	가능	가능	가능	가능
객체 지향 프로그래밍	불가능	가능	가능	가능	가능



### 하나 더 알기

### 프로그래밍 언어별 주요 활용 분야

범용으로 만든 프로그래밍 언어들은 대부분 여러 가지 목적으로 활용할 수 있지만, 추가적인 소프트웨어 라이브러리나 운영 체제의 지원 여부에 따라 주로 활용할 수 있는 분야를 특정할 수 있다.

주요 활용 분야	C	C++	C#	Java	Python
시스템 및 드라이버 개발	○	○			
데스크톱 앱 개발		○	○	○	
웹 서비스 개발			○	○	○
게임 개발		○	○	○	○
모바일 앱 개발			○	○	
데이터 처리		○	○	○	○
기계 학습		○	○	○	○

### 요약 정리

- 컴퓨터 프로그램을 만드는 작업을 프로그래밍이라고 하고, 프로그래밍에 사용되는 언어를 프로그래밍 언어라고 한다.
- 프로그래밍 언어는 컴퓨터가 실행해야 할 작업을 제한된 개수의 기호와 단어를 사용해서 표현하는 특별한 언어이다.
- 절차적 프로그래밍은 실행해야 할 명령이나 명령이 포함된 함수를 순서대로 불러 실행시키는 프로그램 코드의 설계 및 작성 방법이다.
- 객체 지향 프로그래밍에서는 객체를 설계하고, 객체들 사이의 상호 작용과 변화를 중심으로 프로그램 설계와 작성이 이루어진다.
- 프로그래밍 언어들은 최신 기술 및 새로운 요구 사항을 반영하며 지속해서 개선되어 발전하고 있으며, 새로운 프로그래밍 언어들도 꾸준히 개발·사용되고 있다.

### 자기 평가

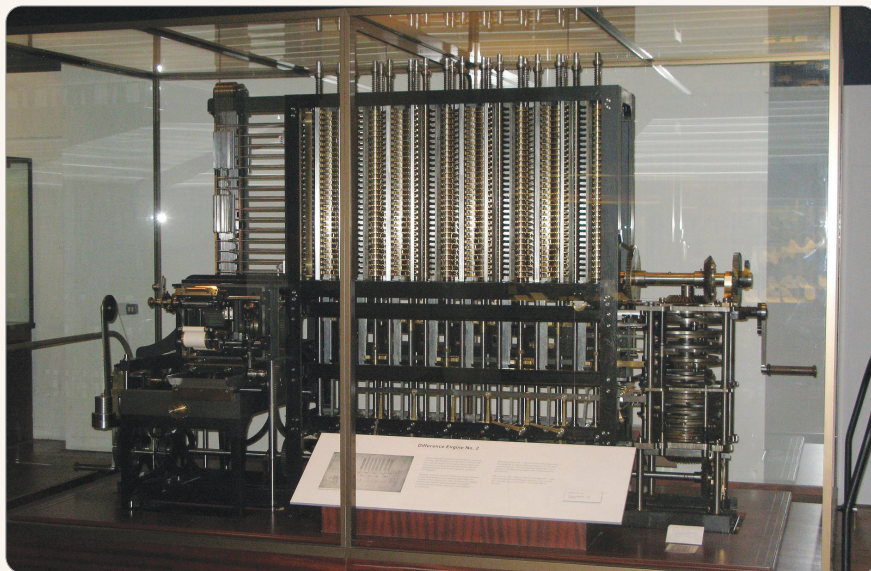
범주	평가 항목	성취도		
		😊	🙂	😞
지식·이해	프로그래밍 언어의 개념과 종류를 설명할 수 있다.			
과정·기능	프로그램 개발 분야 및 용도에 따라 적절한 프로그래밍 언어를 선택할 수 있다.			
가치·태도	프로그램 개발 목적에 적절한 프로그래밍 언어를 선택할 수 있다.			



# 19세기에 컴퓨터와 프로그램을 만든 천재들

19세기 초, 찰스 배비지(Charles Babbage)와 에이다 러브레이스(Ada Lovelace)는 현대 컴퓨터 과학의 기초를 마련하는 데 중요한 역할을 하였다.

배비지는 ‘차분 기관’이라는 기계적 계산 장치를 설계하고 만드는 과정에서 보다 범용적으로 사용할 수 있는 기계적 범용 컴퓨터인 ‘해석 기관’을 설계하였다. 비록 당시 기술적 한계로 완성이 되지 못하였지만, 이 아이디어는 컴퓨터 과학 발전에 크게 기여하였다.



▲ [그림 I-7] 1991년 런던 과학박물관에서 찰스 배비지의 설계도를 토대로 만든 차분 기관 2호

에이다 러브레이스는 1842년 배비지와 연구 결과를 바탕으로 ‘배비지의 해석 기관에 대한 분석’을 출판하였는데, 이 논문에는 기계 작동 방식을 설명하는 최초의 알고리즘이 포함되어 있었다. 러브레이스의 논문은 존재하지 않는 기계를 위한 프로그래밍 설계의 획기적인 성과로, 이후 러브레이스를 최초의 프로그래머로 인정하는 계기가 되었다.

러브레이스는 또한 서브루틴, 점프, 루프, 조건문 등 프로그래밍 언어의 핵심 개념을 고안하고, 베르누이수 계산을 위한 알고리즘을 제시하였다. 이러한 공헌으로 미국 국방성은 1980년에 개발한 프로그래밍 언어에 그녀의 이름을 따와 ‘에이다(Ada)’라고 명명하였다. 이는 컴퓨터 공학 발전에 대한 에이다 러브레이스의 기여를 인정하고 기리는 행위였다.



▲ [그림 I-8] 에이다 러브레이스



# 프로그래밍의 절차와 알고리즘



Think

음식 조리법에 나오는 재료들을 준비한 후, 조리 순서와 방법을 따라 조리하면 원하는 음식을 만들 수 있다. 음식 조리법은 어떤 문제를 해결하는 명확한 단계와 과정인 알고리즘과 비슷하다.

컴퓨터를 사용해서 원하는 작업을 수행하고 문제를 해결하기 위해서는 명확하고 구체적으로 컴퓨터가 실행해야 할 작업을 순서대로 나열해야 한다.

**준비물**

김밥용 김      밥  
 깻잎      당근  
 김밥용 햄      단무지      어묵

**조리법**

- 김밥용 김은 4등분으로 자르고, 당근은 가능하게 채 썰고, 나머지 재료는 자른 김 길이에 맞추어 얇게 2등분한다.
- 밥에 참기름, 소금, 통깨를 넣어 밀간한 후 한 김 식힌다.
- 식용유를 두른 팬에 당근을 넣고 소금을 뿌려 볶은 후 햄도 함께 볶는다. 어묵은 간장과 올리고당을 넣고 볶아 밀간한다.
- 김 위에 밥을 얇게 펴 발라 주고 깻잎을 먼저 깔아 재료들을 하나씩 올려 돌돌 만든다. 그 후 참기름을 바르고 통깨를 뿌리면 완성!

▲ '꼬마 김밥 만들기'의 준비물 및 조리법

**💡 컴퓨터를 사용해서 원하는 작업을 수행하거나 문제를 해결하기 위한 알고리즘은 어떻게 작성할 수 있을까?**

# 01

## 프로그래밍의 절차

### ▶ 학습 목표

- 프로그램을 효율적으로 개발하기 위한 체계적 절차와 단계별 유의 사항을 설명할 수 있다.

### ▶ 학습 개념

- 프로그램 개발 절차, 프로그램 개발 단계별 유의 사항



프로그래밍을 할 때 '요구 사항 분석, 설계, 구현, 테스트, 유지·보수'의 단계를 거치면, 원하는 작업의 수행뿐만 아니라 문제 해결에 필요한 프로그램을 빠르고 효과적으로 개발할 수 있다.



▲ [그림 I-9] 프로그래밍 절차

### 1 요구 사항 분석

요구 사항 분석 단계에서는 프로그램에 필요한 요구 사항들을 수집, 분류, 조직화한 후 명확하게 기술하고 검증하게 된다.

#### ▼ [표 I-6] 요구 사항 분석 단계

요구 사항 수집	인터뷰, 설문 조사, 사용자 시나리오 분석 등을 통하여 프로그램 사용자와 이해관계자로부터 요구 사항을 수집한다.
요구 사항 분류 및 조직화	수집된 요구 사항을 분류하고 정리한다. 이때 분류는 기능적 요구 사항과 비기능적 요구 사항으로 나눈다.
요구 사항 기술	요구 사항을 명확하고 구체적으로 기술한다. 이 과정은 모호한 요구 사항을 명확히 하며, 이해관계자 간의 혼란을 최소화하는 데 도움이 된다.
요구 사항 검증	사용자와 이해관계자로부터 받은 요구 사항의 정확성과 완전성을 검사·증명하는 과정이다.

요구 사항 분석 단계에서는 프로그램의 목적과 사용자의 요구 사항을 정확하게 파악하여 문서화하는 것이 중요하다. 요구 사항이 불명확하게 작성되거나 오류가 있는 경우 이후의 프로그램 개발 과정에서 더 큰 문제가 발생할 수 있기 때문이다.

### 2 설계

설계 단계에서는 요구 사항 분석 결과를 바탕으로 프로그램의 구조를 설계함으로써 프로그램 구현에 필요한 구체적인 구조와 상세한 기능을 검토하게 된다.

설계 단계는 크게 세 부분으로 나눌 수 있다.

▼ [표 I-7] 설계 단계

고수준 설계	프로그램 구조 설계, 데이터 흐름 다이어그램, 시스템의 구성 요소 및 상호 작용 등을 고려해 프로그램의 전체 구조를 결정한다.
저수준 설계	시스템 구성 요소의 동작 방식, 사용할 데이터 구조, 인터페이스 디자인을 결정한다.
설계 검토	설계의 완전성, 일관성, 정확성 등을 검증한다. 문제점을 발견하고 수정하여 개발 단계에서 발생할 수 있는 문제를 사전에 방지한다.

설계 단계는 프로그램 개발에 결정적인 역할을 하며, 이후 프로그램 개발 과정에서 효율적인 구현, 테스트, 유지 보수를 가능하게 해 준다.

### 3 구현

구현 단계에서는 설계 결과에 따라 프로그램의 목적과 실행 환경 등에 가장 적합한 프로그래밍 언어를 선택하고 프로그램의 입력, 출력, 처리를 구체화하여 프로그램 코드를 작성한다.

구현 단계에서는 표준화한 프로그래밍 방식과 규칙에 따라 프로그램을 작성하여 프로그램 코드의 품질을 유지하고, 테스트 및 유지 보수 단계에서의 프로그램 코드 수정을 쉽게 할 수 있게 작성한다. 또한 프로그램 코드에 대한 설명 주석을 충분히 함께 기록하는 것이 좋다.

### 4 테스트

테스트 단계에서는 구현 단계에서 작성한 프로그램의 정확한 작동 여부를 확인하기 위하여 다양한 시나리오와 데이터를 활용하여 프로그램을 검증하고 오류를 수정한다.

단위 테스트와 시스템 테스트를 포함한 여러 유형의 테스트를 진행 중에 발견한 문제점을 수정하여 프로그램의 안정성을 확보한다.



### 5 유지 보수

프로그램을 성공적으로 개발한 후에도 지속적인 업데이트와 개선 작업이 필요하다. 유지 보수 단계에서는 오류 수정, 성능 향상, 새로운 기능의 적용 등을 진행한다.

사용자와 기술적인 변화 등에 따른 프로그램 개선과 발전을 위해서는 끊임없는 유지 보수가 필수적이다.

# 02

## 알고리즘

### ▶ 학습 목표

- 알고리즘의 개념 및 표현 방법을 설명할 수 있다.
- 문제를 해결하기 위한 알고리즘을 순서도나 의사 코드 등으로 작성할 수 있다.

### ▶ 학습 개념

- 알고리즘의 개념, 알고리즘의 표현 방법, 순서도, 의사 코드



### 1 알고리즘의 개념

알고리즘(algorithm)은 9세기 페르시아 수학자인 알콰리즈미(Muhammad ibn Musa al-Khwarizmi)의 이름에서 유래한 말로, 어떤 문제의 해결을 위한 방법과 절차를 말한다.

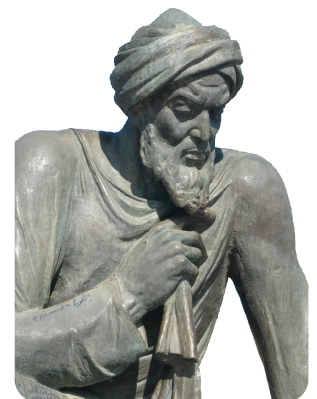
예  $48 - \{63 \div (5 + 4)\} \times 5 = 48 - (63 \div 9) \times 5$

$$= 48 - 7 \times 5$$

$$= 48 - 35$$

$$= 13$$

▲ 사칙 연산 알고리즘



▲ [그림 I-10] 알콰리즈미

### 2 알고리즘의 특징

알고리즘은 잘 정의된 형식 언어를 사용해서 표현한다.

알고리즘은 어떤 입력이나 초기 상태로 시작하여 구체적이고 명확한 계산이나 처리 과정에 따라 명령이 실행되면, 정확하게 정의되어 있는 유한한 상태로 바뀌며 마지막에 는 결과를 출력하고 종료된다.

#### ▼ [표 I-8] 알고리즘의 특징

특징	설명
입력	작업이나 계산을 위하여 어떤 입력이나 초기 상태로 시작한다.
명확성	구체적이고 명확한 계산이나 처리 방법에 따라 명령을 실행한다.
유한성	정확하게 정의되어 있는 유한한 상태들로 바뀐다.
출력	마지막 상태에서는 작업이나 계산의 결과를 출력하고 종료된다.

### 3 알고리즘 표현 방법


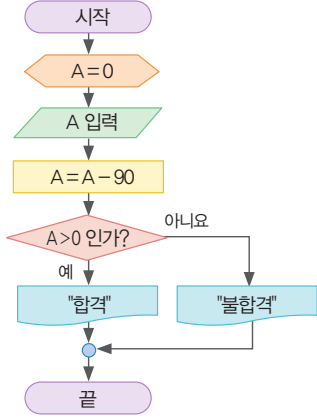






알고리즘은 순서도, 의사 코드, 프로그래밍 언어 등을 사용해서 표현한다.

#### (1) 순서도

순서도는 특별한 작업이나 처리를 의미하는 여러 가지 도형과 흐름선을 사용해서 알고리즘을 표현한 것이다.

다음은 어떤 점수를 입력받아 그 값이 90보다 크면 '합격', 그렇지 않으면 '불합격'을 출력하는 알고리즘을 표현한 순서도이다.

▼ [표 I-9] 순서도 도형의 의미와 예시

도형	의미	예시
	흐름	
	시작/끝	
	준비	
	입력/출력	
	처리	
	조건 판단	
	인쇄	
	연결자	

#### (2) 의사 코드



의사 코드는 일반적인 자연어를 특정 프로그래밍 언어의 문법과 형태로 모방하여 작성한 코드이다.

의사 코드(Pseudo Code)는 일반적으로 사용하는 언어를 프로그램 코드처럼 사용해서 알고리즘을 표현한 것이다. 컴퓨터로 바로 실행할 수 없으며, 특정 프로그래밍 언어로 프로그램을 작성하기 전에 알고리즘을 어렵잡아 표현하는 데 사용한다.

다음은 정숫값이 순서대로 저장된 리스트에서 가장 큰 값을 찾는 알고리즘을 파이썬 언어의 들여쓰기를 사용해서 표현한 의사 코드이다.

#### 의사 코드

```

정수 리스트를 준비한다.
최댓값을 저장할 변수를 준비한다.
리스트의 첫 번째 값을 최댓값 변수에 저장한다.
리스트에 들어 있는 값들을 한 번에 하나씩 확인하면서:
    그 값이 최댓값 변수에 저장된 값보다 크면:
        최댓값 변수에 그 값을 저장한다.
최댓값 변수에 저장된 값을 출력한다.
    
```

### (3) 프로그래밍 언어

컴퓨터가 실행할 수 있는 프로그래밍 언어를 이용해서 알고리즘을 표현할 수 있다. 특정 프로그래밍 언어에 맞는 문법과 규칙에 따라 알고리즘을 표현하고 컴퓨터를 통해서 바로 실행한다.

다음은 두 정수를 입력받아 차잇값을 계산하는 알고리즘을 표현한 파이썬 코드이다.

```


프로그래밍 언어



```

a = int(input())
b = int(input())
if a>b:
    t = a-b
else:
    t = b-a
print(t)

```


```



#### 문제

두 개의 자연수 a, b의 최대 공약수를 출력하는 알고리즘을 표현해 보자.

최초의 알고리즘으로 알려진 유클리드 알고리즘은 어떤 두 자연수 a와 b의 최대 공약수를 찾는 방법이다. a를 b로 나눈 나머지가 r일 때 a와 b의 최대 공약수가 b와 r의 최대 공약수와 같다는 점을 활용한다.

유클리드 알고리즘을 순서도, 의사 코드, 프로그래밍 언어로 표현하면 다음과 같다.

▼ [표 I - 10] 유클리드 알고리즘의 표현(순서도, 의사 코드, 프로그래밍 언어)

순서도	의사 코드	프로그래밍 언어
<pre> graph TD     Start([시작]) --&gt; Input[/a, b 입력/]     Input --&gt; Decision{b == 0}     Decision -- 예 --&gt; OutputA[a 출력]     OutputA --&gt; End([종료])     Decision -- 아니요 --&gt; Process[t = b b = a % b a = t]     Process --&gt; Decision </pre>	<p>두 정수 a와 b를 입력받는다.</p> <p>반복문 <math>b \neq 0</math> 동안:</p> <p style="padding-left: 20px;"><math>t = b</math></p> <p style="padding-left: 20px;"><math>b = a</math>를 <math>b</math>로 나눈 나머지</p> <p style="padding-left: 20px;"><math>a = t</math></p> <p>a를 출력한다.</p>	<pre> a = int(input()) b = int(input())  while b != 0:     t = b     b = a % b     a = t  print(a) </pre>



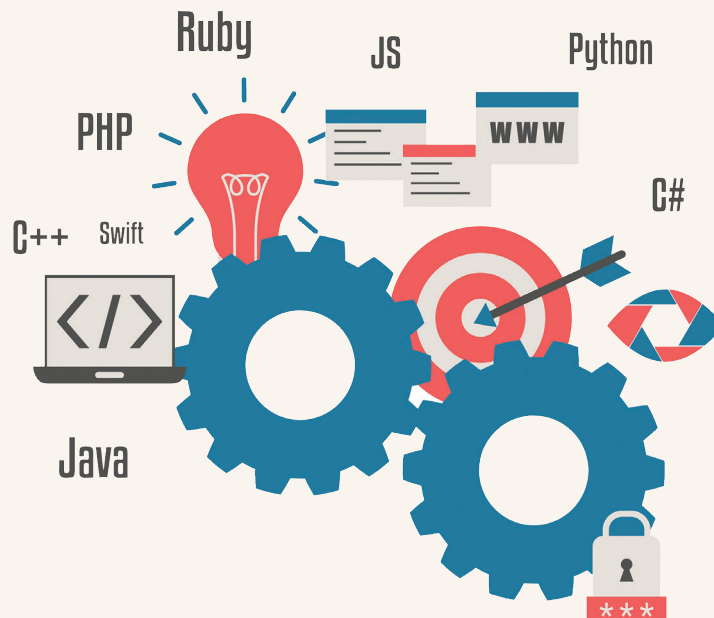


## 기계만 이해하기 쉬운 언어에서 인간 중심의 프로그래밍 언어로의 진화

초창기 프로그래밍 언어는 컴퓨터가 직접 이해하고 실행할 수 있는 기계어로 시작되었다. 기계어는 0과 1, 즉 2진수 형태로 표현하는데, 컴퓨터에는 직관적이지만, 인간에게는 사용하기 어렵고 복잡한 언어이다. 기계어는 컴퓨터마다 호환성이 없어 다른 컴퓨터에서는 같은 명령어가 작동하지 않는 문제가 있었다.

기계어의 문제점에 대한 해결책으로 인간이 이해하기 쉽도록 설계된 2세대 프로그래밍 언어인 어셈블리어가 개발되었다. 어셈블리어는 기계 언어와 일대일 대응하는 기호로 이루어져 있어서 인간이 한층 쉽게 코드를 작성하고 이해할 수 있도록 만들어졌다. 오늘날 어셈블리어는 특히 시스템 해킹이나 컴퓨터 구조 연구 등에서 여전히 사용 중이다.

1950년대에 컴퓨터 사용이 널리 퍼지면서 더 많은 사람이 소프트웨어를 이해하고 개발할 수 있도록 3세대 프로그래밍 언어가 개발되기 시작하였다. 포트란(Fortran)과 코볼(COBOL)이 그 첫 사례로, 이들 언어는 인간의 언어에 가까워 고급 언어라고도 하며 컴파일러를 통해서 컴퓨터가 이해할 수 있는 기계어로 번역한다. 고급 언어는 사용 편의성을 높이고 다양한 방향으로 발전해 나갔으며, 현재는 정해진 문법 구조와 형식으로 구성된 프로그래밍 언어가 널리 사용된다.



# 3



## 프로그램 개발 환경



Think

조리 도구 세트를 사용하면 요리를 편리하게 할 수 있고, 공구 세트가 있으면 여러 가지 간단한 수리를 쉽게 할 수 있다.

마찬가지로 프로그램 개발과 관련한 여러 가지 도구 프로그램들을 모아 놓은 통합 개발 환경을 사용하면, 쉽고 편리하게 프로그램을 작성할 수 있다.



 통합 개발 환경에는 어떤 도구 프로그램들이 세트로 포함되어 있을까?

# 01

## 통합 개발 환경

### ▶ 학습 목표

- 통합 개발 환경의 개념과 필요성을 말할 수 있다.

### ▶ 학습 개념

- 통합 개발 환경의 개념, 통합 개발 환경의 필요성



### 1 통합 개발 환경의 이해

프로그램 개발은 요구 사항 분석, 알고리즘 작성, 프로그래밍 언어로 소스 코드 작성, 작성한 소스 코드를 컴퓨터가 실행할 수 있는 실행 파일로 변환하는 번역 과정, 발견한 오류를 수정하는 디버깅(Debugging) 과정을 거친다.

통합 개발 환경(IDE: Integrated Development Environment)이란 프로그램 개발과 관련한 여러 가지 도구 프로그램, 즉 코드 편집기, 컴파일러/인터프리터, 디버거 등의 프로그램이 유기적으로 연결되어 동작함으로써 편리하고 빠르게 프로그램 개발을 할 수 있도록 돕는 소프트웨어이다.

코드 편집기는 프로그램 소스 코드를 작성하고 수정하는 데 사용되는 텍스트 편집 프로그램이다.

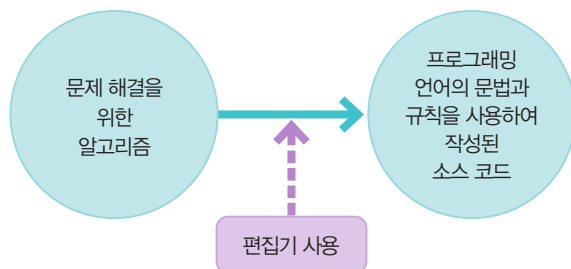
코드 편집기는 프로그래밍 언어의 문법과 규칙에 따라 예약어나 기호들을 서로 다른 색깔로 바꾸어 표시해 줌으로써 프로그램 코드의 가독성을 높여 준다. 또한, 들여쓰기를 비롯한 코드 자동 완성 기능을 제공할 뿐만 아니라 원하는 코드 부분을 빠르게 찾거나 바꿀 수 있는 기능도 제공한다.

### ■ 소스 코드

프로그램을 만드는 재료가 되는 코드



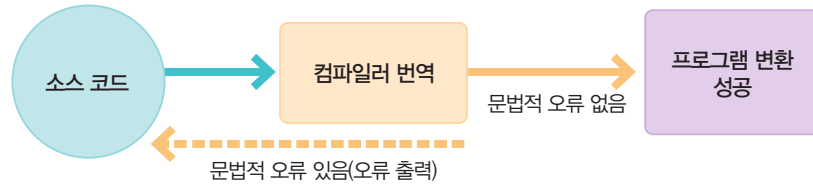
파이썬 프로그래밍 언어를 설치하면, IDLE(Integrated Development and Learning Environment)라는 통합 개발 환경이 기본으로 함께 설치된다.



▲ [그림 I-11] 편집기를 사용한 소스 코드 작성

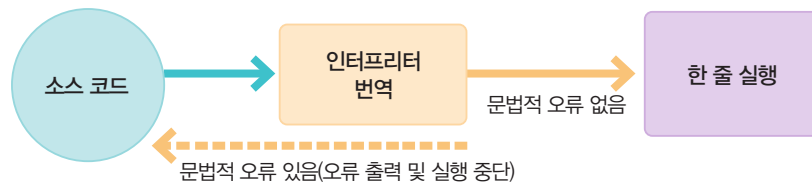


컴파일러(Compiler)는 프로그래밍 언어로 작성된 소스 코드를 컴퓨터에서 실행 가능한 형태의 프로그램으로 변환해 주는 프로그램이다. 컴파일러는 작성한 프로그램 코드를 한 번에 모두 읽어 들인 후 컴퓨터가 실행할 수 있는 어셈블리어나 기계어로 변환시키는 역할을 한다. 프로그램 코드에 문법적 오류가 있는 경우는 변환하지 못하고, 그 이유를 간단히 알려 준다.



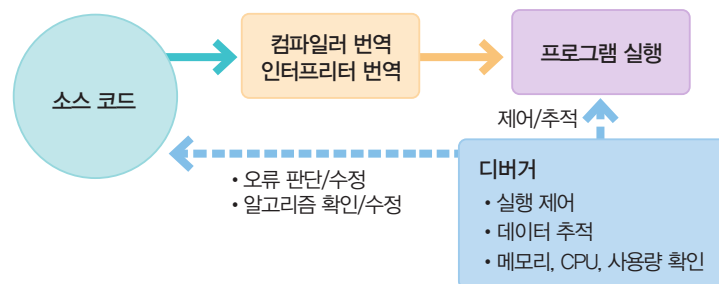
▲ [그림 I-12] 컴파일러의 프로그램 변환과 오류 처리

인터프리터(Interpreter)는 소스 코드를 한 번에 한 줄씩 읽고 해석하여 실행시키는 프로그램이다. 프로그램 실행 도중에 오류가 발생하면, 그 이유를 간단히 알려 주고 프로그램 실행을 중단한다.

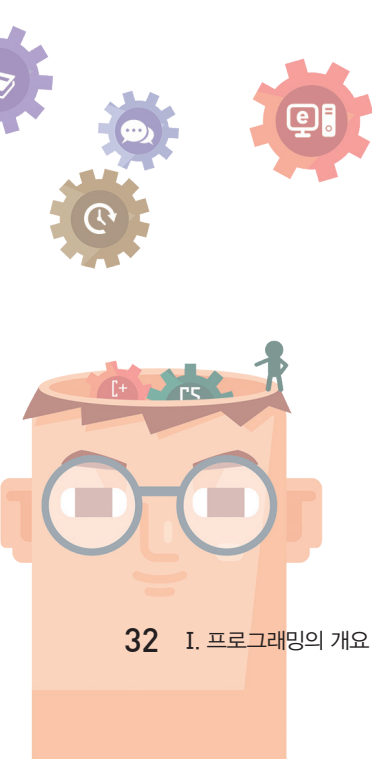


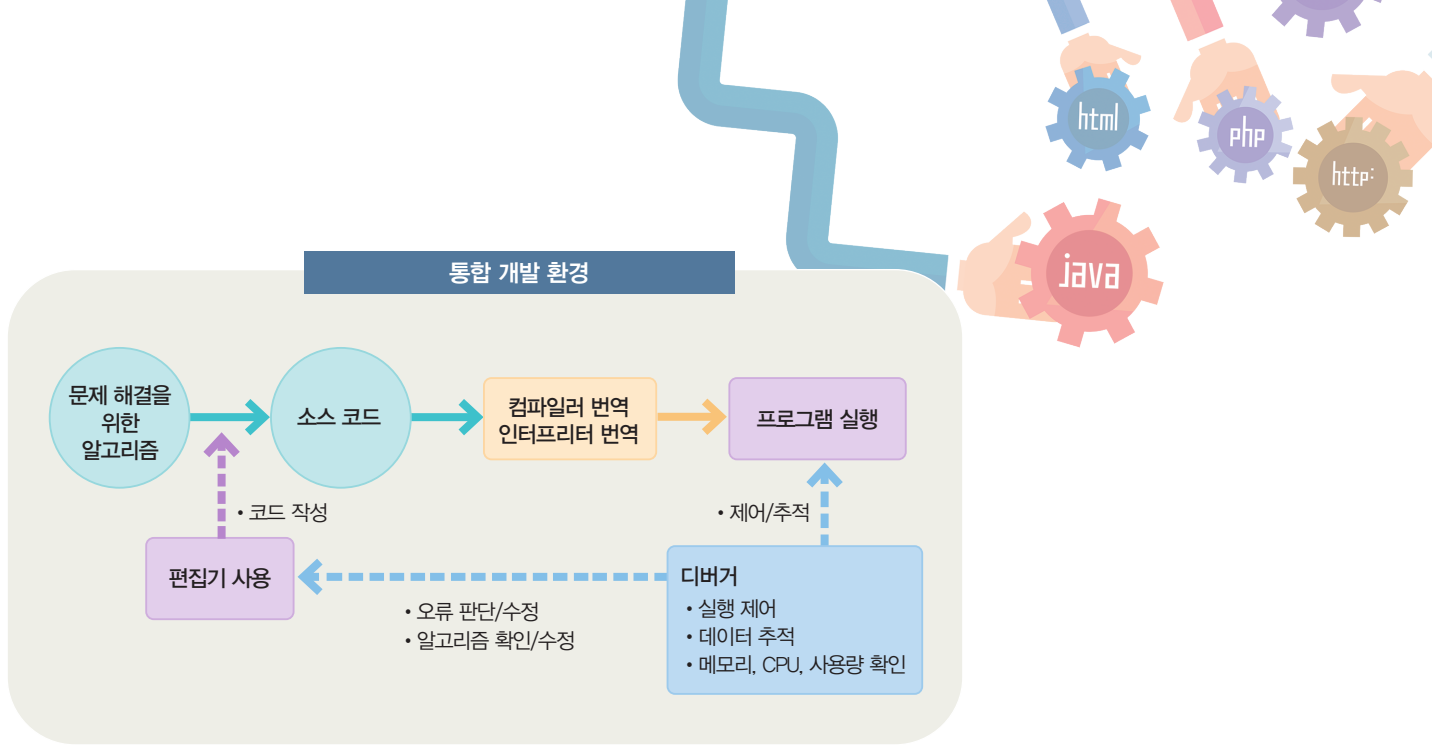
▲ [그림 I-13] 인터프리터의 해석과 실행 중단

디버거(Debugger)는 오류를 찾아내고 테스트하는 데 사용되는 프로그램이다. 프로그램의 실행을 제어하고 프로그램이 실행되는 과정에서 프로그램이 사용하는 여러 가지 값들과 프로그램의 상태 변화를 알 수 있도록 해 준다. 프로그램 실행 도중에 잠시 정지시키거나 원하는 단계까지만 선택적으로 실행시키는 등의 작업이 가능하다. 해당 프로그램이 사용하는 메모리나 CPU 사용량도 보여 주기 때문에 프로그램 실행 과정에서 발생하는 오류나 잘못된 동작을 찾고 프로그램을 수정할 수 있도록 도움을 준다.



▲ [그림 I-14] 디버거의 제어/추적 기능



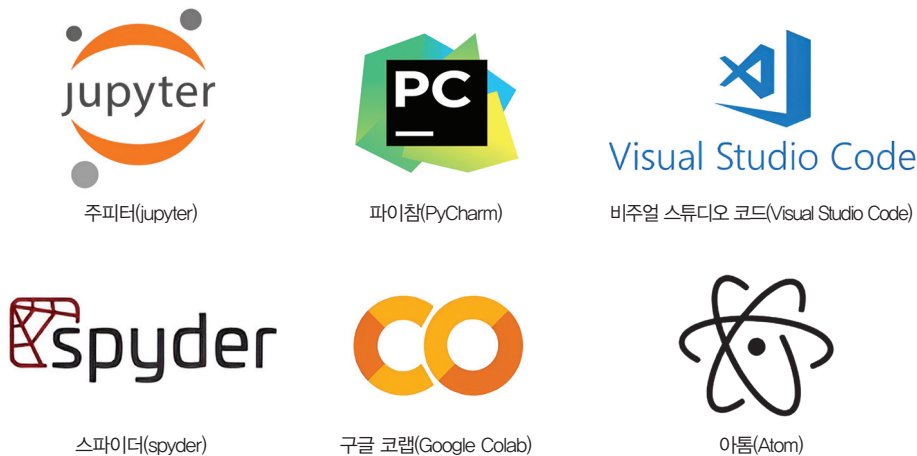


▲ [그림 I-15] 통합 개발 환경

## 2 통합 개발 환경의 종류

여러 가지 통합 개발 환경이 만들어지면서 사용자는 필요한 프로그래밍 언어를 자유롭게 선택하여 사용할 수 있게 되었다.

통합 개발 환경의 종류에 따라 지원하는 프로그래밍 언어, 컴파일러 및 인터프리터 포함 유무, 코드 강조색, 화면 배치, 클라우드 시스템 연동 여부 등이 다르다. 따라서 사용하기려는 프로그래밍 언어, 프로그램 개발 방식이나 협업 여부, 개인적 익숙함과 편리성 등을 고려하여 자유롭게 선택하여 사용할 수 있다.



▲ [그림 I-16] 통합 개발 환경의 종류

# 02

## 공동 개발 환경

### ▶ 학습 목표

- 프로그래밍 공동 개발 환경의 필요성을 설명할 수 있다.
- 공유 및 협업을 통한 프로그래밍을 경험할 수 있다.

### ▶ 학습 개념

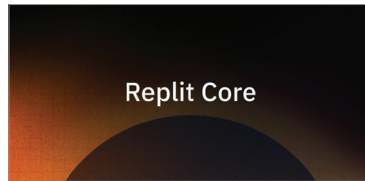
- 공동 개발 환경



### 1 공동 개발 환경의 개념

여러 사람이 협업하여 프로그램을 개발하도록 온라인으로 연결되거나 특별한 통합 개발 환경 소프트웨어를 설치하지 않고도 웹 브라우저 접속만으로도 프로그램을 개발할 수 있는 온라인 공동 개발 환경이 많이 개발되어 함께 사용되고 있다.

온라인 공동 개발 환경에서는 코드 작성 및 편집, 코드에 따른 글자색 표시 기능뿐만 아니라 여러 프로그램 개발자가 함께 프로젝트를 수행하고 협업하는 데 필요한 소스 코드 및 문서 공유 기능, 버전 관리 기능, 프로젝트 관리 기능, 메시지 및 커뮤니케이션 기능 등의 서비스와 도구를 함께 제공한다.



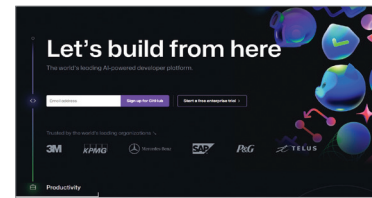
Replit



Eclipse Che

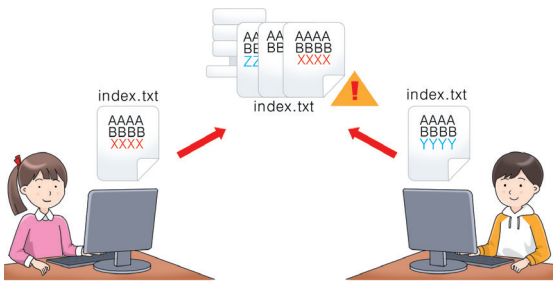


Project Jupyter



GitHub

▲ [그림 I-17] 다양한 온라인 통합 개발 환경(IDE) 및 공동 개발 환경



▲ [그림 I-18] 버전 관리 및 충돌 문제 해결

여러 명의 개발자가 동시에 소스 코드와 문서들을 온라인으로 공유하여 편집·저장하는 과정에서는 버전 관리 및 충돌 문제 발생의 가능성이 뒤따른다. 이를 해결하기 위하여 공동 개발 환경에서는 변경 사항 추적, 코드 비교, 코드 병합, 히스토리 및 복구 기능을 제공하고, 여러 개발자 사이의 원활한 협력 및 의사소통을 위한 자체 메시징 및 커뮤니케이션 서비스도 함께 제공한다.

## 2 공동 개발 환경의 장점

프로젝트 일정, 작업 분배 및 할당, 작업 상태 추적 등을 관리하는 프로젝트 관리 도구를 통하여 여러 개발자의 진행 상황을 파악하며 업무를 조절할 수 있다. 또한, 시공간의 제약 없이 여러 개발자가 협력하여 효율적으로 프로그램을 개발하는 데 도움을 줄 수 있다.



▲ [그림 I-19] 협업 프로젝트 개발

이러한 공동 개발 환경은 개발자들이 팀으로 협업하여 프로젝트를 개발하는 데 필요한 도구와 서비스를 통합하여 제공한다. 이를 통하여 개발자들은 효율적으로 협업하고 코드의 품질을 향상시키며, 프로젝트를 성공적으로 완료할 수 있다.

공동 개발 환경은 개발 효율성을 높이는 것뿐만 아니라, 전 세계 팀원들과 온라인을 바탕으로 협업할 수 있는 환경을 제공한다. 이를 통하여 새로운 비즈니스 기회를 제공하고, 전 세계에서 지식을 수집할 수 있도록 지원하여 글로벌 규모의 비즈니스를 가능케 한다.

### 요약 정리

- 통합 개발 환경은 프로그램 개발과 관련한 여러 가지 도구 프로그램들을 함께 모아 제공하는 소프트웨어이다.
- 통합 개발 환경에는 편집기, 디버거, 컴파일러, 인터프리터, 자동화 도구 등의 프로그램이 포함된다.
- 편집기는 프로그램 소스 코드를 작성하고 수정하는 데 사용된다.
- 인터프리터, 컴파일러는 프로그램 소스 코드를 변환하여 실행하는 데 사용된다.
- 디버거는 프로그램의 실행을 제어하고 오류를 수정할 수 있도록 도움을 준다.
- 여러 사람이 협업하여 프로그램을 개발하는 온라인 통합 개발 환경이나 공동 개발 환경이 함께 사용되고 있다.
- 온라인 공동 개발 환경에서는 협업에 필요한 공유 기능, 버전 관리 기능, 메시지 기능 등의 서비스와 도구가 함께 제공된다.

### 자기 평가

범주	평가 항목	성취도		
		😊	🙂	☹️
지식·이해	통합 개발 환경과 공동 개발 환경의 필요성을 설명할 수 있다.			
과정·기능	통합 개발 환경과 공동 개발 환경을 통하여 협업 프로그래밍 과정을 경험할 수 있다.			
가치·태도	디지털 기술을 통한 공유 및 협업의 중요성을 느끼고 디지털 의사소통 및 협력 역량 강화를 위해 노력할 수 있다.			



# 깃허브(GitHub)와 텐서플로(TensorFlow)

깃허브는 깃(Git, 버전 관리 시스템)을 사용하여 소프트웨어 개발 및 버전 제어를 하기 위한 플랫폼 및 클라우드 기반 서비스로 개발자가 코드를 저장하고 관리할 수 있다. 즉 깃허브는 깃의 분산 버전 제어와 액세스 제어, 버그 추적, 소프트웨어 기능 요청, 작업 관리, 지속적인 통합 및 모든 프로젝트에 대한 위키를 제공한다. 일반적으로 오픈 소스 소프트웨어 개발 프로젝트를 호스팅하는 데 사용된다. 주요 기능과 특징은 다음과 같다.

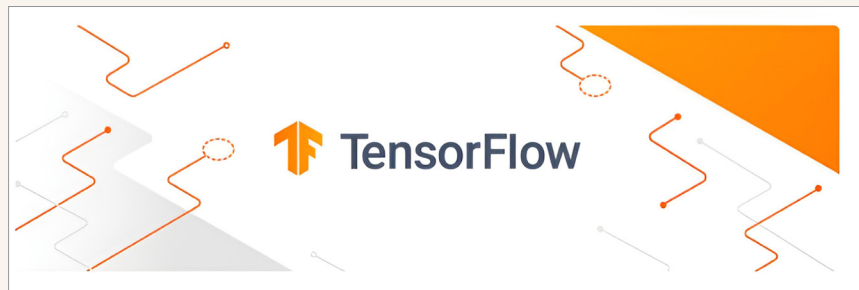
■ 위키(Wiki): 문서의 편집 권한이 모든 사람에게 부여된 웹 사이트 또는 시스템

- 1 버전 관리: Git을 이용해 소스 코드의 변경 사항을 추적하고, 여러 사용자가 동시에 동일한 프로젝트에서 작업하는 것을 지원한다.
- 2 공동 작업: 사용자들이 다른 개발자의 프로젝트를 복제하고 자신의 변경 사항을 제안할 수 있다. 이는 오픈 소스 프로젝트에서 특히 유용하다.
- 3 이슈 트래킹: 문제점을 추적하고, 토론하고, 해결하기 위한 기능이 내장되어 있다.
- 4 액션: 지속적인 통합 및 배포와 같은 자동화 작업을 수행할 수 있다.
- 5 문서화: 프로젝트에 대한 정보를 손쉽게 공유하고 업데이트할 수 있다.

전 세계의 개발자들이 깃허브(GitHub)에서 텐서플로(TensorFlow)라는 프로젝트에 참여하여 이를 발전시키고 있다. 텐서플로는 딥 러닝과 머신 러닝 알고리즘을 수행하는 오픈 소스 라이브러리이다. 이와 같은 공동 개발 환경에서 개발자들은 코드 공유, 문제점 보고 및 논의, 프로젝트 개선 및 확장 등 다양한 활동을 펼치면서 프로젝트를 끊임없이 발전시키고 있다.



▲ [그림 I-20] 깃허브



▲ [그림 I-21] 텐서플로



협업 도구를 이용하여 프로그래밍 언어의 특징을 설명하는 프로그램을 함께 만들어 보자.

- 1 모둠원들과 의사소통을 위하여 어떤 협업 도구를 사용할지 결정해 보자.

( 예 0000 메신저 )

- 2 프로젝트를 관리를 위하여 어떤 협업 도구를 사용할지 결정해 보자.

( 예 0000 일정 )

- 3 모둠원들과 각자 소개할 프로그래밍 언어를 나눈 후 협업 도구를 이용하여 프로그래밍 언어의 특징을 소개하는 프로그램을 만들어 보자.

예

```
[ ] 1 def c():
    2     print("C의 특징")
    3     #C 특징 작성

[ ] 1 def cpp():
    2     print("C++의 특징")
    3     #C++ 특징 작성

[ ] 1 def java():
    2     print("Java의 특징")
    3     #Java 특징 작성

[ ] 1 def python():
    2     print("Python의 특징")
    3     #Python 특징 작성

[ ] 1 print("특징을 알고 싶은 프로그래밍 언어를 선택하세요.")
    2 print("1. C")
    3 print("2. C++")
    4 print("3. Java")
    5 print("4. Python")
    6
    7 n = int(input())
    8
    9 if n==1 :
   10     c()
   11 elif n==2 :
   12     cpp()
   13 elif n==3 :
   14     java()
   15 elif n==4 :
   16     python()
   17 else :
   18     print("잘못 선택했습니다.")
```

**01** 프로그래밍 언어를 이용하여 컴퓨터가 사람의 의도에 따라 작업을 수행할 수 있도록 명확하고 구체적인 실행 명령을 순서대로 나열한 명령어의 집합은?

- ① 네트워킹                      ② 프로그램
- ③ 정보 검색                    ④ 시스템 분석
- ⑤ 컴퓨터 시스템

**02** 프로그램 코드의 설계 및 작성 방법으로, 필요한 명령이나 함수를 순서대로 호출하여 실행하는 형태는?

- ① 절차적 프로그래밍          ② 함수형 프로그래밍
- ③ 객체 지향 프로그래밍      ④ 선언적 프로그래밍
- ⑤ 이벤트 기반 프로그래밍

**03** 1970년대에 개발된 포인터로 메모리 접근이 가능한 프로그래밍 언어는?

- ① C                                ② C++
- ③ C#                              ④ Java
- ⑤ Python

**04** 프로그램 개발 과정에서 요구 사항을 수집, 분류, 조직화하여 명확히 기술하고 검증하는 단계는?

- ① 설계                            ② 구현
- ③ 테스트                        ④ 유지·보수
- ⑤ 요구 사항 분석

**05** 9세기 페르시아 수학자 알콰리즈미의 이름에서 유래한 용어로, 어떤 문제를 해결하기 위하여 계산을 수행하는 엄격한 절차나 작업 과정은?

- ① 함수                            ② 컴파일러
- ③ 알고리즘                      ④ 프로그래밍
- ⑤ 데이터베이스

**06** 알고리즘의 특징 중 구체적이고 명확한 계산이나 처리 방법에 따라 명령을 실행하는 것은?

- ① 추상화                        ② 명확성
- ③ 효율성                       ④ 유한성
- ⑤ 입력과 출력

**07** 특별한 작업이나 처리를 의미하는 여러 가지 도형과 흐름선을 사용하여 알고리즘을 표현한 것은?

- ① 순서도                        ② 의사 코드
- ③ 객체 지향 모델              ④ 프로그래밍 언어
- ⑤ 데이터 다이어그램

**08** 프로그램 개발과 관련한 여러 가지 도구 프로그램들을 함께 모아 제공하는 소프트웨어는?

- ① 운영 체제                    ② 웹 브라우저
- ③ 통합 개발 환경              ④ 애플리케이션 소프트웨어
- ⑤ 데이터베이스 관리 시스템

**09** 프로그램 소스 코드에 작성된 명령을 한 번에 한 줄씩 해석하여 실행시키고, 실행 도중 오류가 발생하면 그 이유를 알려 주며 실행을 중단하는 프로그램은?

- ① 로더                            ② 링커
- ③ 어셈블러                      ④ 컴파일러
- ⑤ 인터프리터

**10** 소스 코드를 해석하여 실행 가능한 형태의 프로그램으로 변환하는 프로그램은?

- ① 에디터                        ② 컨버터
- ③ 디버거                        ④ 테스터
- ⑤ 컴파일러



11 빈칸에 들어갈 알맞은 말을 <보기>에서 찾아 채우시오.

컴퓨터 프로그램을 만드는 작업을 프로그래밍이라고 하고, 프로그래밍에 사용하는 언어를 ( )이라고 한다. 이것은 사람이 컴퓨터를 활용하여 원하는 작업을 실행할 때 프로그램 작성의 수단이 된다.

<보기>

- ㉠ 컴퓨터                      ㉡ 프로그램
- ㉢ 프로그래머                ㉣ 프로그래밍 언어

12 빈칸에 들어갈 알맞은 말을 <보기>에서 찾아 채우시오.

( )은 어떤 데이터와 그 데이터에 접근할 수 있는 코드를 포함하는 객체를 중심으로 프로그램의 코드를 설계하고 작성하는 방법이다.

<보기>

- ㉠ 프로그램                    ㉡ 소프트웨어
- ㉢ 절차적 프로그래밍        ㉣ 객체 지향 프로그래밍

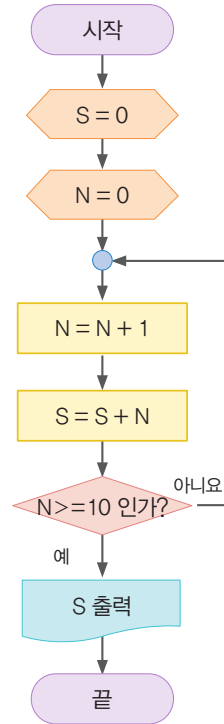
13 <보기>의 프로그래밍의 절차를 순서대로 나열하시오.

<보기>

- ① 구현                        ② 설계                      ③ 테스트
- ④ 유지 보수                ⑤ 요구 사항 분석

( ) → ( ) → ( ) → ( ) → ( )

14 다음 순서도의 실행 과정과 결과를 쓰시오.



(1) 아래 표에 실행 과정에 따른 값을 채우시오.

N	0	1								
S	0	1								

(2) 순서도의 실행 결과값을 쓰시오.

( )