

06

객체 지향과 클래스

- + 학습 목표**
 - 객체의 개념과 구성 요소를 설명할 수 있다.
 - 클래스와 인스턴스를 활용하여 프로그램을 작성할 수 있다.
- + 학습 요소**
 - 객체 지향, 클래스, 인스턴스

생각 깨우기

게임 속 오브젝트는 어떻게 만들어질까?

게임 속에는 다양한 오브젝트가 등장한다. 비슷한 종류의 많은 오브젝트들은 어떻게 생산되는 것일까?



게임 속 많은 오브젝트는 어떻게 하면 편하게 만들 수 있을까?

1 | 객체 지향의 이해

프로그램이 복잡하고 길어질수록 코드를 해석하는 데 어려움이 있다. 이러한 문제를 보완하기 위해 같은 기능이나 목적을 갖는 변수, 함수 등을 묶어서 객체로 만들어 객체들이 상호 작용하며 실행되도록 프로그램을 구성하는 것을 객체 지향이라고 한다.

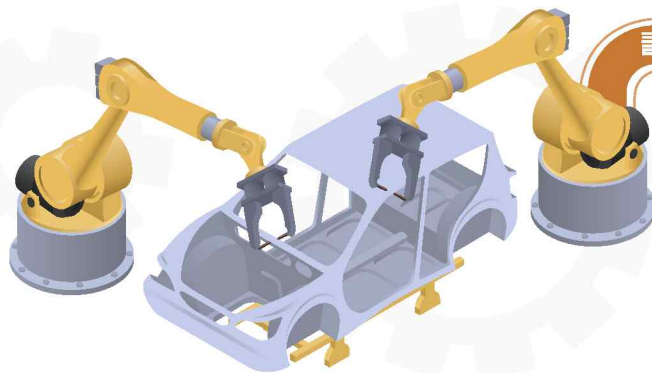
객체 지향 프로그래밍이란 프로그램 구현에 필요한 객체의 역할을 정의하고, 객체를 조합하여 프로그래밍하는 것을 의미한다. 예를 들어 자동차를 만드는 것을 생각해 보자. 자동차들은 각각의 객체로서 자동차 이름, 제조사, 배기량 등의 속성과 주행하기, 주차하기, 앞 차와 거리두기 등의 기능을 가지고 있으며, 다른 자동차의 주행 상황에 따라 동작을 제어할 수 있다.

객체 지향은 객체(Object), 클래스(Class), 인스턴스(Instance)로 구성된다. 객체는 동일한 목적, 기능을 가진 속성과 메서드를 하나로 묶은 것이다. 속성은 객체가 가지고 있는 자료이고, 메서드는 객체가 수행하는 기능이다. 프로그램은 어떤 객체가 메서드를 수행할지로 구성되어 있다. 클래스는 각각의 객체들이 갖는 속성과 메서드를 정의하고 있는 틀로서, 클래스를 활용하여 객체를 만든다. 인스턴스는 클래스를 바탕으로 소프트웨어에 구현된 구체적인 실체다.

예를 들어 아래 그림처럼 공장에서 자동차를 만들고자 한다. 자동차를 만들기 위한 틀을 클래스라 하고, 클래스에서 생성된 자동차를 객체라 한다. 자동차 객체에는 자동차의 종류, 색상, 생산 일자 등의 여러 속성과 자율주행하기, 차선 유지하기 등의 메서드가 할당될 수 있으며, 이를 인스턴스라고 부른다.

메서드

기존 프로그래밍에서의 함수가 객체 지향의 클래스에서 사용될 때 메서드라 한다.



클래스 | 자동차를 만들기 위한 설계

속성

자동차 종류, 색, 생산 일자

메서드

반 자율주행 기능, 차선 유지 기능, 자동 주차 기능

객체



속성

종류(승용차), 색(파란색), 생산 일자 (20xx. 11. 11.)

메서드

반 자율주행 기능, 차선 유지 기능

객체



속성

종류(SUV), 색(빨간색), 생산 일자 (20xx. 8. 15.)

메서드

차선 유지 기능, 자동 주차 기능

객체



속성

종류(승합차), 색(검정색), 생산 일자 (20xx. 3. 15.)

메서드

자동 주차 기능

❗ 객체는 반드시 속성과 메서드를 동시에 포함할 필요는 없다. 속성 혹은 메서드만으로도 구성될 수 있다.

객체 지향 프로그래밍은 객체를 생성하고 객체가 각자의 기능을 수행하면서 다른 객체와 상호 작용하도록 프로그래밍하여 실제 사람들이 사는 세상과 비슷한 형태로 모델링하는 특징이 있다.

객체가 클래스를 바탕으로 생성된 실체라면, 인스턴스는 객체가 소프트웨어에서 실체화된 것이다. 객체 지향 프로그램에서는 클래스를 바탕으로 여러 객체들을 생성하고, 객체가 어떠한 값을 받아서 실체화된 것을 인스턴스라고 한다.

❗ 인스턴스는 객체에 포함된다 고 볼 수 있으며, 객체가 실제로 시스템의 메모리에 할당되어 사용될 때 인스턴스라고 부른다.

같은 종류의 자동차는 설계된 내용에 따라 같은 공장에서 만드는 것이 효율적인 것처럼 객체 지향 프로그래밍은 반복적인 코드가 줄어들고, 클래스를 통해 객체를 한번에 수정할 수 있어 효율적이다.

해 보기 1 객체 만들기 연습하기

🗨️ 교사와 학생 객체를 만드는 클래스가 있다. 교사와 학생 클래스를 참고하여 객체를 만들어 보자.

교사 클래스

- 속성: 이름, 과목
- 메서드: 수업하기, 평가하기, 상담하기, 생활 지도하기 등

예 교사 ①

속성	속성	메서드	메서드
김교사	정보	평가하기	상담하기

교사 ②

속성	속성	메서드	메서드

학생 클래스

- 속성: 이름, 장래 희망
- 메서드: 공부하기, 운동하기, 독서하기, 수다떨기 등

학생 ①

속성	속성	메서드	메서드

학생 ②

속성	속성	메서드	메서드

2 | 객체 지향 실습

객체 지향의 클래스와 객체를 활용하여 프로그램을 작성해 보자.

먼저 객체를 효율적으로 생성하기 위하여 클래스를 만들어 보자. 클래스는 오른쪽과 같은 구문으로 생성할 수 있다.

```
class 클래스 이름:  
    클래스의 내용~
```

클래스를 생성하면 클래스의 이름과 같은 함수를 사용해서 객체를 만들 수 있다.

이때 사용하는 클래스 이름()은 생성자 함수라고도 한다. 이때 클래스를 기반으로 만들어진 객체를 인스턴스라고 한다.

```
객체(인스턴스) 이름 = 클래스 이름()
```

✔ 학생의 성적을 입력받고, 합계와 평균을 계산하여 출력하는 프로그램을 만들어 보자.

01 클래스 생성

실습 1

Students 클래스(속성: name, informatics, math / 메서드: sum, avg, string)를 생성하는 프로그램을 작성하고 실행해 보자.

프로그램

```
1 class Students:                                # Students 이름의 클래스 생성  
2     def __init__(self, name, informatics, math):  
3         self.name = name  
4         self.informatics = informatics  
5         self.math = math  
6  
7     def sum(self):  
8         return self.informatics + self.math  
9  
10    def avg(self):  
11        return self.sum() / 2  
12  
13    def string(self):  
14        return '{}\t{}'.format(self.name, self.avg()).format(self.name,  
15        self.avg())
```

클래스는 def _ _init_ _(self, 추가 속성...) 메서드로 시작한다. class Students:를 통해 Students라는 이름의 클래스를 생성했다.

❗ self는 다른 단어로 바꾸어도 코드상 오류는 없지만 관례적으로 self로 적는다. self가 가지고 있는 속성과 기능에 접근할 때에는 self. 식별자 형태로 접근할 수 있다.

❗ 일반적으로 클래스 이름은 알파벳 대문자로 시작한다. 클래스의 함수의 첫 번째 속성은 self를 사용한다.

예를 들어 “def _ _init_ _ (self, name, informatics, math):” 부분은 클래스의 속성이 name, informatics, math로 구성되어 있음을 의미한다. 이때 _ _init_ _ 메서드를 통해 입력 받은 name, informatics, math 속성의 값은 각 객체에 저장된다.

sum, avg, string 메서드는 합계, 평균, 출력을 위한 코드로, 각 함수에서 처리한 결과를 return을 통해 반환하도록 한다.

02 학생 리스트(객체) 생성

실습 2

학생 리스트(객체)를 생성하는 프로그램을 작성하고 실행해 보자.

프로그램

```
1 students = [
2     Students('김00', 88, 90),
3     Students('정00', 73, 65),
4     Students('한00', 63, 77)
5 ]
```

☑ 설명 students 리스트에 앞에서 생성한 Students 클래스를 활용하여 객체를 생성하도록 한다. 생성된 객체는 리스트 students[0], students[1], students[2]에 저장된다.

03 객체 값 출력

실습 3

김00의 정보 성적을 출력하는 프로그램을 작성하고 실행해 보자.

프로그램

```
1 print('김00의 정보 성적')
2 print(students[0].informatics)
```

실행 결과

```
김00의 정보 성적
88
```

☑ 설명 print(students[0].informatics)와 같이 입력하면 첫 번째 입력받았던 김00의 정보 성적을 확인할 수 있다. 만일 print(students[2].math)를 입력하면 한00의 수학 성적을 확인할 수 있다.

같은 메서드를 실행하는 객체를 여러 개 만들 때 클래스를 활용하여 생성하면 효율적이죠!



실습 4

모든 학생의 이름과 평균을 출력하는 프로그램을 작성하고 실행해 보자.

프로그램	실행 결과								
<pre> 1 print('이름', '평균', sep='\t') 2 for i in students: 3 print(i.string()) </pre>	<table border="1"> <thead> <tr> <th>이름</th> <th>평균</th> </tr> </thead> <tbody> <tr> <td>김00</td> <td>89.0</td> </tr> <tr> <td>정00</td> <td>69.0</td> </tr> <tr> <td>한00</td> <td>70.0</td> </tr> </tbody> </table>	이름	평균	김00	89.0	정00	69.0	한00	70.0
이름	평균								
김00	89.0								
정00	69.0								
한00	70.0								

설명 반복문을 통해 이름, 평균의 값을 하나씩 확인할 수 있다. 이와 같이 클래스를 활용해 객체를 선언하면 편리하게 값을 입력받고 클래스 내의 합계, 평균 등을 계산하는 메서드를 추가로 코드에 입력하지 않아도 결과를 확인할 수 있다.

이처럼 클래스를 활용하여 객체를 여러 개 생성할 수 있으며, 각 객체는 동일한 종류의 속성과 메서드를 가지게 되어 객체를 생성한 후 동일한 메서드를 실행하는 데 유용하다.

Students 클래스는 name, informatics, math라는 속성을 가지고 있으며, 총 3명에 대한 데이터를 가진 객체가 students[0]~students[2]까지 만들어진다. 이때 각 객체들은 Students 클래스의 인스턴스라고 부른다.

print 문에서 sep는 문자열 사이를 무엇으로 구분할지 정한다. 예를 들어 sep='\t'는 일정한 간격을 띄워 문자열을 구분하는 역할을 한다.

알고 가기 ▶ isinstance() 함수



isinstance() 함수는 만들어진 인스턴스*가 어떤 클래스에서 만들어졌는지 확인하는 함수다. isinstance(인스턴스, 클래스)와 같은 형태로 사용하며 첫 번째 매개 변수에는 인스턴스 이름을, 두 번째 인스턴스에는 클래스 이름을 입력한다. 이때 입력한 인스턴스가 입력한 클래스에서 만들어졌다면 True를, 그렇지 않다면 False를 반환한다.

아래 코드를 위에서 실습한 코드에 추가 입력하여 결과를 확인해 보면, students[0] 인스턴스는 Students 클래스에서 만들어졌으므로 결과로 True를 출력한다.

*인스턴스 확인 print('인스턴스 확인 결과: ', isinstance(students[0], Students))

소단원 1 요약

- 객체 지향:** 객체라는 개념을 기반으로 프로그래밍하는 것을 의미하며, 클래스, 객체, 인스턴스로 구성된다.
- 객체:** 동일한 목적, 기능을 가진 속성과 메서드를 하나로 묶은 것이다.
- 클래스:** 각각의 객체들이 갖는 속성과 메서드를 정의하고 있는 틀로, 클래스를 활용하여 객체를 생성하므로 객체를 만들기 위한 설계도라고 할 수 있다.
- 인스턴스:** 클래스를 바탕으로 소프트웨어에 구현된 구체적인 실체를 말한다.

소단원 자기 평가

평가 항목	평가 기준		
	잘함	보통	노력
1. [과정기능] 클래스와 인스턴스를 설명할 수 있다.			
2. [과정기능] 클래스를 정의하고 인스턴스를 생성할 수 있다.			
3. [과정기능] 문제 해결에 적합한 객체를 활용하여 프로그램을 작성할 수 있다.			



권장 칼로리 계산 프로그램 작성하기

건강 관리를 위한 권장 칼로리 계산 프로그램을 만들고자 한다. 키와 활동 지수 속성을 활용하여 권장 칼로리를 안내해 주는 프로그램을 작성해 보자.

- ※ 권장 칼로리 계산 $(키 - 100) \times 0.9 \times \text{활동 지수}$
- ※ 활동 지수: 적은 활동량 25, 보통의 활동량 33, 많은 활동량 40으로 한다.

1 다음 클래스를 생성하기 위해 빈칸을 채워 보자.

- 클래스의 이름은 Data, 속성명은 height(키), activity(활동 지수)이다.
- 메서드는 권장 칼로리를 계산하는 메서드(cal), 키, 활동 지수, 권장 칼로리를 반환하는 메서드(string)를 포함한다.

```

프로그램
1 class [빈칸]:
2     def __init__(self, [빈칸], [빈칸]):
3         self.height = height
4         self.activity = activity
5
6     def [빈칸](self):
7         return (self.height-100) * 0.9 * self.activity
8
9     def string(self):
10        return '{}\t{}\t{}'.format(self.height, self.activity, self.cal())
    
```

2 친구들의 데이터를 객체로 생성하고, 권장 칼로리를 출력하는 프로그램을 완성해 보자.

```

프로그램
[빈칸]
    
```

탐구 활동
자기 평가

평가 항목

평가 기준

잘함 보통 노력

1. 클래스를 생성할 수 있는가?
2. 객체를 활용하여 프로그램을 작성할 수 있는가?

• 잘함 (내용을 이해하고 설명함) • 보통 (내용을 이해함) • 노력 (내용을 부분적으로 이해함)