

선언형과 함수형 프로그래밍

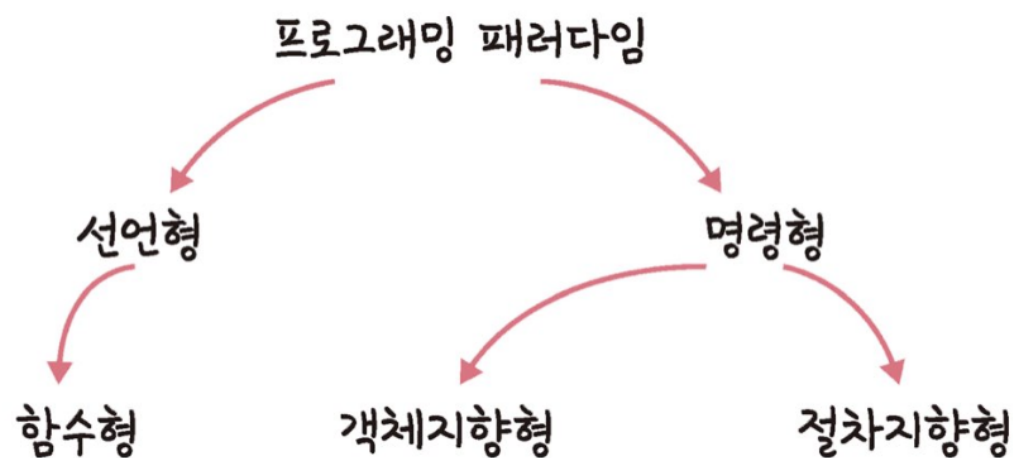
Tags

프로그래밍 패러다임

프로그래밍 패러다임

프로그래밍 패러다임(Programming Paradigm)은 프로그래머에게 프로그래밍의 관점을 갖게 하고 코드를 어떻게 작성할 지 결정하는 역할을 한다. 새로운 프로그래밍 패러다임을 통해서는 새로운 방식으로 생각하는 법을 배우게 되고, 이를 바탕으로 코드를 작성하게 된다.

프로그래밍 패러다임은 크게 선언형, 명령형으로 나누며, 선언형은 함수형이라는 하위 집합을 갖는다. 또한, 명령형은 다시 객체지향, 절차지향으로 나눈다.



선언형과 함수형 프로그래밍

- 선언형 프로그래밍(declarative programming)이란
 - ‘무엇을’ 풀어내는가에 집중하는 패러다임
 - “프로그램은 함수로 이루어진 것이다”라는 명제가 담겨 있는 패러다임
- 함수형 프로그래밍(functional programming)은 선언형 패러다임의 일종

함수형 프로그래밍

정의

함수형 프로그래밍은 작은 ‘순수 함수’(ex. `reduce()`)들을 블록처럼 쌓아 로직을 구현하고 ‘고차 함수’를 통해 재사용성을 높인 프로그래밍 패러다임

등장 과정

명령형 프로그래밍을 기반으로 개발했던 개발자들은 개발하는 소프트웨어의 크기가 커짐에 따라, 복잡하게 엉켜있는 스파게티 코드를 유지보수하는 것이 매우 힘들다는 것을 깨닫게 되었다.

그리고 이를 해결하기 위해 함수형 프로그래밍이라는 프로그래밍 패러다임에 관심을 갖게 되었다. 함수형 프로그래밍은 거의 모든 것을 순수 함수로 나누어 문제를 해결하는 기법으로, 작은 문제를 해결하기 위한 함수를 작성하여 **가독성을 높이고 유지보수를 용이하게** 해준다.

순수 함수

출력이 입력에만 의존하는 것을 의미

고차 함수

함수가 함수를 값처럼 매개변수로 받아 로직을 생성할 수 있는 것

일급 객체

이 때 고차 함수를 쓰기 위해서는 해당 언어가 일급 객체라는 특징을 가져야 하며 특징은 다음과 같다.

- 변수나 메서드에 함수를 할당할 수 있다.
- 함수 안에 함수를 매개변수로 담을 수 있다.
- 함수가 함수를 반환할 수 있다.

자바의 경우

jdk 1.8 이전의 자바는 객체지향 프로그래밍을 지원하는 반면에, jdk 1.8부터 함수형 프로그래밍 패러다임을 지원한다.

- 랴다식
- 생성자 레퍼런스
- 메서드 레퍼런스
- 스트림(stream)과 같은 표준 API

자바를 이용한 함수형 프로그래밍(Functional Programming) 예시

어떤 단어의 리스트가 있을 때 다음과 같은 문제가 주어졌다고 하자.

1. 단어의 크기가 2 이상인 경우를 필터링한다.
2. 모든 단어를 대문자로 변환한다.
3. 모든 단어를 앞글자만 잘라내어 변환한다.
4. 모든 단어를 스페이스로 구분한 하나의 문자열로 합친다.

이러한 요구 사항을 함수형 프로그래밍 없이 개발한다면 코드가 상당히 길어지고, 복잡해질 것이다. 하지만 함수형 프로그래밍을 적용한다면 해당 요구사항들 역시 비교적 간단하게 처리할 수 있다. 해당 문제를 해결하는 코드는 다음과 같다.

```
public class WordProcessTest {

    private final List<String> words = Arrays.asList
        ("TONY", "a", "hULK", "B", "america", "X", "nebula", "Korea");

    @Test
    void wordProcessTest() {
        String result = words.stream()
            .filter(w -> w.length() > 1)
            .map(String::toUpperCase)
            .map(w -> w.substring(0, 1))
            .collect(Collectors.joining(" "));

        assertThat(result).isEqualTo("T H A N K");
    }
}
```

처음 Stream API를 접하면 오히려 복잡하게 느낄수 있으므로 위의 코드를 전혀하지 못해도 크게 문제가 없다. 우리가 처음 객체지향 프로그래밍을 접했을 때 처럼 함수형 프로그래밍도 새로운 프로그래밍 패러다임인 만큼 학습과 적응이 필요하기 때문이다.

우리가 주목해야 하는 부분은 위의 wordProcess 메소드 내부에서 filter나 map, collect 등과 같은 함수의 파라미터로 함수가 전달된다는 것이다. 각각의 함수로 넘어간 파라미터 함수는 다음의 역할을 한다.

- filter 파라미터로 넘어간 함수: 주어진 문자열의 길이가 1인지를 판별함
- 첫 번째 map 파라미터로 넘어간 함수: 주어진 문자열을 대문자로 변환함
- 두 번째 map 파라미터로 넘어간 함수: 주어진 문자열을 첫 단어만 잘라냄
- collect 파라미터로 넘어간 함수: 주어진 문자열을 스페이스 간격으로 이어붙임

추가 질문

- 함수형 프로그래밍의 장단점은 무엇일까?

Reference

<https://mangkyu.tistory.com/111>

<https://jongminfire.dev/함수형-프로그래밍이란>