

PRESENTATION

19주차 주제

프로그래밍 패러다임

함수형 프로그래밍

by mun



프로그래밍 패러다임이란

프로그래머에게 프로그래밍의 관점을 갖게 해 주고, 결정하는 역할
개발자가 프로그래밍을 위해 어떠한 '관점'을 가지고 개발을 할지에 대해서
결정을 하는데 도움을 주는 것을 의미

프로그래밍

하나 이상의 관련된 추상 알고리즘을 특정한 프로그래밍 언어를 이용해
구체적인 컴퓨터 프로그램으로 구현하는 기술

패러다임

어떤 한 시대 사람들의 견해나 사고를 근본적으로 규정하고 있는 테두리로서의 인식의 체계,
또는 사물에 대한 이론적인 틀이나 체계를 의미하는 개념을 의미

명령형 vs 선언형

무엇(What)을 할 것인지 나타내기보다
어떻게(How) 할 건지를 설명하는 방식

어떻게 할건지(How)를 나타내기보다
무엇(What)을 할 건지를 설명하는 방식

절차지향 프로그래밍

수행되어야 할 순차적인 처리 과정을 포함하는 방식
(C, C++)

객체지향 프로그래밍

객체들의 집합으로 프로그램의 상호작용을 표현
(C++, Java, C#)

논리형 프로그래밍

어떤 일에 대해서 해결하는 '속성들'에 주목하는
프로그래밍 기법

함수형 프로그래밍

순수 함수를 조합하고 소프트웨어를 만드는 방식
(클로저, 하스켈, 리스프)

명령형 vs 선언형

무엇(What)을 할 것인지 나타내기보다
어떻게(How) 할 건지를 설명하는 방식

어떻게 할건지(How)를 나타내기보다
무엇(What)을 할 건지를 설명하는 방식

절차지향 프로그래밍

수행되어야 할 순차적인 처리 과정을 포함하는 방식
(C, C++)

객체지향 프로그래밍

객체들의 집합으로 프로그램의 상호작용을 표현
(C++, Java, C#)

논리형 프로그래밍

어떤 일에 대해서 해결하는 '속성들'에 주목하는
프로그래밍 기법

함수형 프로그래밍

순수 함수를 조합하고 소프트웨어를 만드는 방식
(클로저, 하스켈, 리스프)

함수형 프로그래밍

자료 처리를 수학적 함수의 계산으로 취급하고
상태와 가변 데이터를 멀리하는 프로그래밍 패러다임의 하나

부수 효과가 없는 순수 함수를 1급 객체로 간주하여
파라미터나 반환값으로 사용할 수 있으며, 참조 투명성을 지킬 수 있다

함수형 프로그래밍

자료 처리를 수학적 함수의 계산으로 취급하고
상태와 가변 데이터를 멀리하는 프로그래밍 패러다임의 하나

부수 효과가 없는 순수 함수를 1급 객체로 간주하여
파라미터나 반환값으로 사용할 수 있으며, 참조 투명성을 지킬 수 있다

부수 효과 (Side Effect)

- 이벤트 발생 시 이벤트를 전달 받을 변수의 값이 변경됨
- 자료 구조를 제자리에서 수정함
- 객체의 필드값을 설정함
- 예외나 오류가 발생하며 실행이 중단됨
- 콘솔 또는 파일 I/O가 발생함

함수형 프로그래밍

자료 처리를 수학적 함수의 계산으로 취급하고
상태와 가변 데이터를 멀리하는 프로그래밍 패러다임의 하나

부수 효과가 없는 순수 함수를 **1급 객체**로 간주하여
파라미터나 반환값으로 사용할 수 있으며, 참조 투명성을 지킬 수 있다

1급 객체 (First-Class Object)

다음과 같은 것들이 가능한 객체를 의미한다

- 변수나 데이터 구조 안에 담을 수 있다
- 파라미터로 전달 할 수 있다
- 반환값으로 사용할 수 있다
- 할당에 사용된 이름과 무관하게 고유한 구별이 가능하다

함수형 프로그래밍

자료 처리를 수학적 함수의 계산으로 취급하고
상태와 가변 데이터를 멀리하는 프로그래밍 패러다임의 하나

부수 효과가 없는 순수 함수를 1급 객체로 간주하여
파라미터나 반환값으로 사용할 수 있으며, **참조 투명성**을 지킬 수 있다

참조 투명성(Referential Transparency)

- 동일한 인자에 대해 항상 동일한 결과를 반환해야 한다
- 참조 투명성을 통해 기존의 값은 변경되지 않고 유지된다

(Immutable Data)

함수를 실행하여도 어떠한 상태의 변화 없이 항상 동일한 결과를 반환하여
항상 동일하게(투명하게) 실행 결과를 참조(예측)할 수 있다는 것을 의미

순수 함수

부수효과(side-effect)가 없는 함수, 함수의 실행이 외부에 영향을 끼치지 않는 함수
입력으로 들어오는 파라미터에만 의존한다

```
static int number = 100
```

```
static int getNumber() {  
    return number;  
}
```

외부 변수에 의존 -> 순수 함수 X

```
static int getNumber(int num) {  
    return number;  
}
```

입력에만 의존 -> 순수 함수 O

```
static int plusNumber(int a, int b) {  
    return a + b;  
}
```

입력에만 의존 -> 순수 함수 O

스레드 안전하고
병렬적인 계산이 가능하다

익명 함수

함수형 언어에서 작성 가능한 이름이 없는 형태의 함수

```
int fun(int x) {return x*x}
```

fun 라는 이름을 가진 함수

```
[](int x) -> int {return x*x};
```

익명 함수로 작성

자바에서는 함수형 기반의
Stream API 로 사용 가능

고계 함수 (고차 함수)

함수를 다루는 함수

정수를 함수의 인수로 전달할 수 있듯이 어떤 함수도 다른 함수의 인수로 전달할 수 있다
(함수형 언어에서는 함수도 값(value)로 취급한다)

```
@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");
}
```

함수의 파라미터로 함수가 전달

자바에서 함수형 프로그래밍을 위한
많은 구현체를 제공 중

자세한건 StreamAPI를 더 공부해야...

**다양한 프로그래밍 패러다임을 이해하고,
어떤 관점으로 프로그래밍 할지
결정해 보아요**

감사합니다



참고 :

https://ko.wikipedia.org/wiki/%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D_%ED%8C%A8%EB%9F%AC%EB%8B%A4%EC%9E%84 , <https://adjh54.tistory.com/97> ,
<https://mangkyu.tistory.com/11>