

목차

1. 객체지향적 사고

2. 함수적 사고

3. 언어

A. 자연 언어

i. 구성 요소

1. 명사

2. 동사

B. 프로그래밍 언어

i. 객체지향 언어

1. ¹객체지향적 사고

객체지향적 사고에서는 아웃풋을 담당자로 만들고 메소드를 아웃풋 안으로 들여보낸다. 예를 들어 미리담기리스트를 추가하고 싶다는 것이 요구사항이라면 일단 담당자부터 추출한다. 이때 담당자는 데이터와 명사 중심으로 뽑는다. 따라서 리스트가 담당자가 되고 추가라는 메소드를 가지게 되는 것이다..

2. 함수적 사고

함수는 원소끼리 대응관계를 규칙을 만들어 놓은 것이다. 제약조건은 어떤 입력이 들어와도 확정적 결과를 가져와야 한다. 함수를 사용하는 이유는 예측하기 위해서다. 미리 어떤 행위가 일어날지 규칙을 파악하기 위해서다.

또한 함수라는 것은 데이터 관점으로 볼지 규칙 관점으로 볼지에 따라 다르다. 함수를 다른 말로 표현하면 데이터플로우다이아그램으로 표현할 수 있다. 규칙을 흐름으로 표현해 놓은 것이 데이터플로우다이아그램이다. 함수가 어떠한 인풋을 받고 어떤 아웃풋을 가지는 지를 표현하는 것이다.

3. 언어

언어라는 것은 기본적으로 사람의 생각과 다른 사람의 생각을 전달하기 위한 매개체다. 다른 사람의 생각이 내 머리에 들어오기 위해, 소통하기 위해 언어를 사용한다.

또한 언어는 세상이 오감을 통해 머리 속으로 들어와 생각으로 자리잡는데, 거기에도 규칙이 있다. 즉, 언어에도 규칙이 존재한다.

생각은 세상을 전부 기억할 수 없으니 추상화해서 기억하는 것을 말한다. 여기서 추상화란 공통성이나 특징, 부분을 뽑아서 기억하는 것이다.

¹ <https://velog.io/@majaeh43/절차지향-객체지향-함수형-프로그래밍>

추상화는 일종의 타입을 만들어내고 거기에 값을 채워 실체를 만들어내는 것이다.

A. 자연 언어

자연 언어는 사람 간의 언어를 뜻한다. 그리고 세상에 관한 표현이다. 따라서 세상을 표현하는 모든 재료가 다 들어있다.

자연어는 모든 영역을 일반적으로 잘 나타낼 수 있도록 되어있기 때문에 모호성을 가진다. 또한 세상을 구성하는 요소를 명사와 동사로 나눈다.

i. 구성 요소

1. 명사

명사는 세상과 생각을 만드는 재료로, 시간이 지나도 변하지 않는 것이다. 구조나 틀, 형태와 같다. 명사를 일반화시키면 일반명사고, 일반명사는 타입이 된다. 그리고 여기에 값을 넣으면 고유명사가 되고, 고유명사는 실체가 된다.

2. 동사

동사는 함수로, 시간이 지남에 따라 주체를 변화시키는 규칙에 관한 것이다. 행위에 관한 것이므로 주체가 반드시 있어야 한다.

B. 프로그래밍 언어

i. 객체지향 언어

프로그래밍 언어는 우리의 생각을 프로그램으로 옮길 때 프로그램이 알아들을 수 있도록 하기 위해 사용하는 언어다. 프로그래밍 언어의 가장 큰 특징은 모호성이 없다는 것이다. 왜냐하면 프로그래밍 언어는 컴퓨터라는 특정 영역을 잘 나타내는 구조를 가지기 때문에 이 영역의 특성을 포함하고 있다.

따라서 요구사항이 들어왔을 때 프로그래밍 언어로 바뀌야 하는데,
객체지향프로그래밍 언어에서는 일반명사는 클래스라 하고 클래스를
new 해서 만든 것을 오브젝트라고 한다. 여기서 클래스는 타입과
인스턴스를 의미하고, 메소드는 함수를 의미한다.

즉, 요구사항에서 구현까지 가는 것은 자연어를 클래스와 메소드로 바뀌서
자바로 표현하는 것이다.