7월 6일 면접문제

1. 함수형 프로그래밍, 절차지향 프로그래밍, 객체지향프로그래밍이란 무엇인가

① 절차지향 프로그래밍(Procedural Programming)

: 일이 진행되는 순서대로 프로그래밍을 하는 방법으로, 프로그램 전체가 유기적으로 연결되도록 만드는 프로그래밍 기법이다. 컴퓨터의 작업 처리 방식과 유사하다. ex. C언어

* 장점: 코드를 따라가며 읽기가 쉬우며, 작성하기 쉽다. 컴퓨터의 처리 구조와 비슷해 실행 속도가 빠르다.
* 단점: 각 코드가 순서에 민감하게 연결되어 있어 유지보수가 어렵다. 프로그램을 분석하기 어렵다. 실행 순서가 정해져 있으므로 코드의 순서가 바뀌면 동일한 결과를 보장하기 어렵다. 디버깅이 어렵다.

② 객체지향 프로그래밍(Object Oriented Programming)

: 실제 세계를 모델링하여 소프트웨어를 개발하는 방법이다. 객체를 먼저 작성하고 함수를 작성하는 형태이다(객체간의 상호작용). 데이터(객체)를 먼저 디자인하고 데이터에 맞는 메소드(함수)를 구현하는 방식이다(상태를 가지고 있고, 상태를 바꾼다. 데이터 형에 메소드가 종속된다). 비순수 함수들을 순수 객체로 묶는다.

다음과 같은 세 가지 특성을 가진다.

ㄱ. 캡슐화: 관련된 데이터와 알고리즘(코드)이 하나의 묶음으로 정리된 것으로써, 관련된 코드와 데이터가 묶여 있고, 오류가 없어 사용이 편리하다. 데이터를 감추고 메소드를 통해 상호작용하는 방법인데, 라이브러리로 만들어 업그레이드하면 쉽게 바꿀 수 있다.

* 메소드: 메시지에 따라 실행시킬 프로시저로써 객체지향 언어에서 사용되는 것.

객체지향 언어에서는 메시지를 보내 메소드를 수행함으로써 통신을 수행한다.

ㄴ. 상속: 작성된 클래스를 이어 받아서 새로운 클래스를 생성하는 기법으로, 기존 코드를 재활용해서 사용하는 것을 의미한다.

ㄷ. 다형성: 하나의 방법으로 많은 상황에 대처하는 기법이다. 개념적으로 동일한 작업을 하는 함수들에 똑같은 이름을 부여할 수 있으므로 코드가 더 간단해지는 효과가 있습니다.

* 장점: 사람이 생각하는 방식과 닮아서 분석과 설계의 전환이 쉽다(‘객체’는 ‘메소드’다). 코드의 재사용이 가능하다. 코딩이 절차지향에 비해 간편하고 디버깅이 쉽다.
* 단점: 상대적으로 처리 속도가 느리다. 설계에 많은 시간이 소요되며 문제가 발견되면 처음으로 돌아가 다시 짜야하는 잠재적 위험이 있다. 테스트가 어렵다. 메소드를 통해서만 접근이 가능하여 절차지향 처럼 특정 함수에 접근할 수 없다.

③ 함수형 프로그래밍(Functional Programming)

: 함수를 먼저 작성하는 프로그래밍 방식이다. 함수를 먼저 만들고 함수에 맞는 데이터를 세팅한다(상태가 없다). 자료 처리를 수학적인 함수의 계산으로 취급하고 상태와 가변적인 데이터를 배제하는 프로그래밍이다. 실행 순서를 지정할 필요가 없기 때문에 비절차형 언어이다.

* 장점: 변경 가능한 상태를 갖지 않고, 불변상태로 만들어 사이드 이펙트를 미연에 방지한다. 객체지향보다 코드가 간결하다. 비절차형이라 평가 시점이 중요하지 않다. 테스트가 쉬워서 1회 테스트만으로 신뢰성이 보장된다. 데이터형에 구애받지 않는다.
* 단점: 상태를 조작할 수 없다.

<https://susu91.tistory.com/105>

<https://brownbears.tistory.com/407>

<https://madplay.github.io/post/functional-programming-object-oriented-programming>

<https://medium.com/@lazysoul/%ED%95%A8%EC%88%98%ED%98%95-%ED%94%84%EB%A1%9C%EA%B7%B8%EB%9E%98%EB%B0%8D%EC%9D%B4%EB%9E%80-d881230f2a5e>

2. 함수형 프로그래밍이 갖는 장점과 단점은 무엇인가.

(1번 문제의 답에 포함)

3. 순수함수란 무엇인가.

순수 함수란 동일한 인자를 넣었을 때 어디에서 호출이 되든 항상 동일한 반환값을 갖으며, 외부와 영향을 주고 받지 않는 함수이다. 이와 같은 특성으로 인해 모듈화가 가능하도록 함수형 프로그래밍의 로직을 효율적으로 구성할 수 있다.

<https://seungwoohong.tistory.com/14>

<https://jeong-pro.tistory.com/23>