7/6

1. 함수형 프로그래밍, 절차지향 프로그래밍, 객체지향프로그래밍이란 무엇인가

절차지향 프로그래밍 : 프로그램 전체가 하나의 덩어리로 동작하고, 함수 밖에 있는 변수, 즉 전역 변수는 프로그램 전체에서 공유하여 사용하는 방식의 프로그래밍.

(ex) CountCow = 1 (소 머릿수를 저장하는 함수)

addCow = function(){} (새로운 소 추가하는 함수)

-🡪 말이나 양이 새로 추가 되면 변수가 중복되는 것을 막기 위해서 또다른 변수/함수를 추가해야하 함.

객체지향 프로그래밍 : 변수와 함수를 class 단위로 묶고, class를 통해 새로 객체들을 생성하여 같은 기능의 변수와 함수를 재활용할 수 있게 하는 프로그래밍 방법.

(ex) class Farm {

변수와 함수를 class 단위로 묶어서 이 변수와 함수를 공통으로 사용하는 개별적인 객체를 만들어 냄. 객체마다 별개의 영역에서 count변수를 가질 수 있다.

Count = 1;

add(newCows);

}

CowFarm = new Farm();

CowFarm.count = 1;

HorseFarm = new Farm();

HorseFarm.count = 1;

함수형 프로그래밍 : 함수를 변수로, 값으로 바라보고 변하지 않는 값과 함수로만 코딩을 하는 프로그래밍 방식이다(변수선언 금지).

새로운 데이터가 추가/수정될 때마다 변수를 사용하여 이를 관리하는 대신, 함수 내부에서 추가된 데이터가 반영된 새로운 객체를 만들어 반환하는 방식을 사용한다. 함수형 프로그래밍에서는 함수를 인자로 받을 수 있고, 반환할 수 있도록 모든 함수가 일급객체로 동작하며, 이런 일급객체를 사용할 수 있는 함수를 ‘고차 함수’라고 부른다.

\*객체를 처음 생성할 때 고정값을 인자로 준다

\*새로운 값/수정되는 값을 새로 생성되는 객체의 인자값으로 넣고, 새로운 객체를 반환한다.

(ex) class Farm(count){

add(newCows);

}

cowFarm = new Farm(1);

add(2){return new Farm(count+2);}

1. 함수형 프로그래밍이 갖는 장점과 단점은 무엇인가.

장점 : 객체 내부에서 상태가 변하는 변수가 없기 때문에, 변수 하나를 여러 루틴에서 수정할 때 발생하는 오동작을 막을 수 있다.

단점 : 순수한 함수형 언어로만 코딩하면 값을 변경할 때마다 새로운 객체를 만들어내기 때문에 객체지향 언어보다는 속도가 느리다.

1. 순수함수란 무엇인가.

입력되는 파라미터(인자)로만 동작하며, 함수 외부의 값을 사용하지 않고, 함수 내부에서 임의의 타 객체 생성 또한 금지하는 함수. (변수가 없다)