

3.20 월요일

네트워크 이슈로 인해 프로그램 설치가 늦어져서 최초 커밋 진행하였습니다.

3.21 .화요일

variable(변수) 변수를 만드는 법으로 시작하여 변수의 데이터 타입에 관하여 배웠습니다.

int: 정수형 long : 비슷함 float: 실수형 double: 조금 더 정밀한 소수점

변수의 이름은 직관적인 이름, 명시성을 가지는 것이 팀원들에게 잘 전달된다.

변수 사용의 이점은 3.3이라는 값을 사용해야 하는 곳에 ABS라는 변수를 만들어 사용하면 만약에 3.3이라는 값이 바뀌었을 때 ABS의 값만 변경하면 전체적으로 바꿀 수 있기 때문이다. 또한 final은 상수로 고정하여 사용할 때 사용한다. final의 이점은 동작의 무결성을 보장하는 것에 있다. 그냥 변수로 지정할 경우 코드의 진행 과정에서 값이 변할 수 있기 때문이다..

스캐너

스캐너의 경우에

for 문 (초기화;조건;증감){}

-위의 조건식이 만족하는 동안 반복된다. 증감의 경우는 없어도 상관없다. 중괄호 내에 for문을 활용하여 반복할 내용이 들어간다.

if 문

if를 사용하여 다음 식이 참일 경우 그 아래의 문장을 진행하며 아닐 경우 else나 else if를 사용하여 그 외의 경우에 어떻게 진행할지 서술한다. else가 많아지는 경우 코드 파악의 문제가 생기는 경우가 있어 if문을 반복하는 것을 추천한다.

while()

while 뒤의 괄호 안의 값이 참일 경우 반복된다.

소괄호 내의 식이 참일 경우에 반복되지만 그 값을 벗어나게 되면 멈춘다.

switch ()

여러 케이스를 만들어 각 케이스의 경우에 해당하는 코드가 실행되는 조건문이다.

default의 경우 조건에 맞는 케이스가 없는 경우 진행되는 코드이다.

break;를 사용하지 않으면 다음 케이스로 진행되기 때문에 break 사용해야 한다.

ascii코드 - 문자를 출력하는 숫자로 나올 경우가 있는데 이것은 데이터 형이 정수형으로 되어있을 때 아스키 코드값이 출력되기 때문이다.

3.22 배열과 클래스

배열 -

1. 데이터 타입을 적고 대괄호를 적어 만들 배열이 어떤 데이터 타입을 갖고 있는지를 알려준다.
2. 변수와 마찬가지로 이름을 정한다. 명시적이어야 한다.
3. 필요하다면 초기화를 진행한다.
 - new를 붙여서 생성하면 heap 메모리 할당을 강제할 수 있다.

배열은 메모리에 순차적으로 배치된다. 5개로 이루어진 배열의 경우 0~4의 공간에 위치한다.

힙과 스택

스택은 지역 변수에 설정하는 것이고 new의 경우 heap이라는 공간에 할당되는 것이다.

-스택과 달리 힙은 메모리에 상주하기 때문에 어디서든 접근할 수 있다.

foreach 배열의 모든 값을 대입하여 아래의 코드를 진행한다.

-초기화나 조건식등이 향후 변경에 유리하다.

-for(ini...)의 경우 계속 조건식을 검사해야 함

이 두가지 관점이 있지만 그때 그때 사용할 코드를 정해야 할 듯 하다.

객체의 상수화와 객체 내부의 상수화 하는 것은 다른 것이다.

클래스

생성자 생성 규칙

1. public을 적고 클래스 이름을 적은 후 소괄호 중괄호 작성한다.
2. 값을 외부에서 받을 것이면 소괄호에 입력 받을 형태를 작성한다.
3. 실제 클래스가 구동될 때 하고 싶은 작업을 중괄호 내에 작성한다.

매서드-클래스 내부에 기능을 수행하는 집합들을 매서드라고 부른다.

1. public을 작성하고 소괄호를 작성하고 중괄호를 작성한다.
2. 리턴 타입을 public 옆에 작성한다.
3. 매서드의 이름을 그 옆에 적는다- 이름은 역시나 이 작업이 무엇을 하는지 명시적으로 알려줄 수 있는 형태
4. 중괄호 내에서는 실제 매서드 이름에 해당하는 작업을 진행하면 된다..
 - 이 때 단순히 작업만 하고 정보 반환이 없다면 리턴 타입은 void 입니다.
 - 참 거짓을 반환한다면 boolean, 숫자를 반환하면 long 이나 int
 - 무엇을 반환 하느냐에 따라 적절한 형태를 적어주면 된다.

전구는 스스로를 킬 수 없는 존재이기 때문에 다른 객체의 도움을 받아야 한다. 그 개념의 일

환으로 탄생한 것이 domain service 개념으로 비즈니스 관점을 더 명확하게 만들어준다는
이점을 가지고 있다.

객체가 스스로 하기 힘든 것 들을 모두 도메인 서비스로 재배치한다. 이를 통해 가독성과 유
지보수성의 향상을 가져온다.

객체 지향 프로그래밍 OOP

모든 정보를 객체화하여 레고처럼 무엇이든 만들 수 있는 것을 목표로 하는 개발 방법이다.

객체가 너무 커지면 객체 하나하나가 너무 복잡해지는 문제가 있을 수 있고 너무 작을 경우에
는 서로 연결이 너무 복잡해지는 경우가 생기는 이것을 주의해야 한다.

domain driven development -study material 살펴보기

랜덤 다이스 1,2,3에 관하여

1,2번 문제는 깔끔하지 않아도 결과가 맞게 만들 수 있었는데 3번의 경우에 앞이 깜깜해지는
것을 느꼈다. 비록 복수 전공이긴 했지만 컴공 공부를 했었는데 선생님이 코드 짜는 것을 보
면서 클론 코딩을 하는 것이 다였다. 무엇을 클래스로 만들어야 할지 어디서부터 어디까지 같
은 공간에 뒀야 하는지에 답이 안나와 답답해지는 시간이었다. 역치를 키운다고 하셨지만 충
격이 큰 것은 어쩔수 없었다.

주말간에 지방에 다녀와 세세하지 않은점 죄송합니다.