<Lab4 보고서>

32191597 박민규

**1. 과제 및 프로그램 개요:** 이번 과제는 Lab3에서 만든 클래스들을 활용하고, abstract class를 interface로 바꾸어 적용하는 과제다. 또한 데이터를 저장하는 과정에서 collection인 List를 사용하고, IHealthCalculator interface를 가지고 있는 BMICalculator, BFPCalculator, BMRCalculator, CalorieIntakeCalculator class를 List와 Map collection의 형태로 저장하고, class로 따로 만들어 준다. 이 클래스는 add, remove, sort, predicate 등 다양한 역할을 수행한다. collection을 활용하여 data들의 normal한 출력과, sort된 출력, predicate된 출력을 정상적으로 실행하는 것이 이번 과제의 최종 목표이다.

**2. 주요 코드 캡쳐 및 설명**



텍스트이(가) 표시된 사진

자동 생성된 설명먼저 IHealthCalculator 라는 interface를 만들어 주었기 때문에, 각 클래스에 implements 해 준다.

IHealthCalculator interface의 구성은 위 그림과 같다.

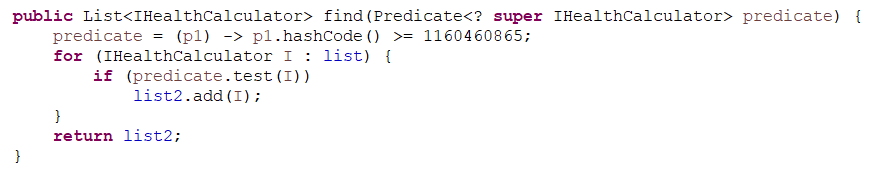
Mode에 따른 BMICalculator, BFPCalculator, BMRCalculator, CalorieIntakeCalculator 객체 반환 메소텍스트이(가) 표시된 사진

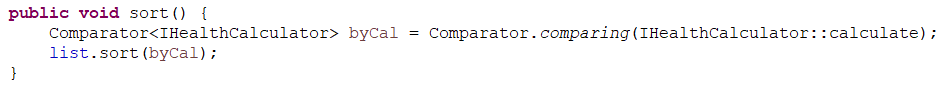
자동 생성된 설명드를 가진 HealthCalculatorFactory class를 만들었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Person data를 클래스의 멤버 필드에 저장하는 setDataInput method를 위 그림과 같이 각 class에 override하여 재정의해 사용하였다.

다음은 HealthCalculatorListManager의 find method다 이 코드에선 리스트의 각 object의 hashcode에 따라 조건에 맞는 object만 새로운 리스트에 add하여 return 하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명 위 그림은 getRandom method의 구현이다.

위 그림은 sort list by calculate를 해주기 위한 sort method다.

HealthCalculatorMapManager 에서도텍스트이(가) 표시된 사진

자동 생성된 설명 마찬가지로 hashCode에 따라 predicate할 수 있도록 코드를 작성했다.

HealthCalculatorListManager와 HealthCalculatorMapManager는 둘 다 IHealthCalculator를 가지고 있는 collection이지만, HealthCalculatorListManager는 List interface를 상속받은 ArrayList 자료형으로 이루어졌고, HealthCalculatorMapManager는 map interface를 상속받은 HashMap 자료형으로 이루어졌다. List는 순서가 있는 데이터의 집합으로 데이터의 중복을 허용한다. 반면 Map은 키(Key), 값(Value)의 쌍으로 이루어진 데이터의 집합으로, 순서는 유지되지 않으며 키(Key)의 중복을 허용하지 않으나 값(Value)의 중복은 허용한다.

**3. 메인 method 설명 및 실행화면**

먼저**텍스트, 테이블이(가) 표시된 사진

자동 생성된 설명** data를 ArrayList에 저장한다.

HealthCalculatorListManager**텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명**, HealthCalculatorMapManager에 HealthCalculatorFactory에 저장된 object들을 저장한다. 저장이 끝난 후, 순차적으로 저장되어 있는 값들을 출력한다. 출력 순서는 normal calculate with HealthCalculatorListManager, sort calculate with HealthCalculatorListManager, random calculate with HealthCalculatorListManager, normal calculate with HealthCalculatorMapManager, random calculate with HealthCalculatorMapManager, predict calculate with list, predict calculate with map, user input calculate의 순서로 실행된다. 기본적인 계산 후 실행, 계산된 값들의 정렬 후 출력, 랜덤으로 계산 후 출력, 조건에 맞는 출력 등 과제에서 요구하는 모든 것들을 출력할 수 있도록 했다.

텍스트이(가) 표시된 사진

자동 생성된 설명<sort calculate>

텍스트이(가) 표시된 사진

자동 생성된 설명<random calculate>

텍스트이(가) 표시된 사진

자동 생성된 설명<predict calculate>

**4. 느낀 점**

이번 과제가 개인적으론 꽤나 까다로운 과제였다. Interface의 적용까진 무리 없이 빨리 진행할 수 있었으나, JAVA에서 처음 배우는 collection이라는 개념 때문에 생소한 개념을 코드에 적용시키기에 어려움이 있었던 것 같다. 하지만 성공적으로 교수님의 요구 조건을 모두 만족하는 코드를 작성하는데 성공했다. 과제를 하면서 collection에 대해 더 개인적으로 공부하다 보니, 이제 collection을 사용하는 것이 생각보다 어렵지 않고 적응만 되면 굉장히 편리할 것 같다고 생각한다.