

2022학년도 1학기
JAVA 프로그래밍 I

514760
2022년 봄학기
5/10/2022
박경신

과제 Lab4 (Interface & Collection)

- Lab3 프로그램을 Interface와 Collection을 활용한다.
 - IHealthCalculator 인터페이스
 - *void calculate()* 추상메소드
 - *void getUserInput()* 추상 메소드
 - *void setDataInput(PersonData person)* 추상 메소드
 - HealthCalculatorFactory 클래스
 - **public static HealthCalculator getInstance(Mode mode);** // Mode에 따른 BMI Calculator, BFPCalculator, BMRCalculator, CalorieIntakeCalculator 객체 반환 메소드 추가

과제 Lab4 (Interface & Collection)

- IHealthCalculator 인터페이스를 상속받아 각 클래스마다 *calculate()*, *getUserInput()*, *setDataInput(PersonData person)* 추상메소드 오버라이딩한다.
 - BMICalculator
 - BFPCalculator
 - BMRCalculator
 - CalorieIntakeCalculator는 BMRCalculator를 상속받는다.

과제 Lab4 (Interface & Collection)

- HealthCalculatorListManager와 HealthCalculatorMapManager 클래스

```
public class HealthCalculatorListManager {
    private List<IHealthCalculator> list = null;
    public HealthCalculatorListManager () {
        this.list = new ArrayList<> ();
    }
    ...
}

public class HealthCalculatorMapManager {
    private Map<Integer, IHealthCalculator> map = null;
    public HealthCalculatorMapManager () {
        this.map = new HashMap<> ();
    }
    ...
}
```

과제 Lab4 (Interface & Collection)

□ HealthCalculatorListManager 클래스 멤버

- `public List<IHealthCalculator> getList() { .. }` // 리스트 반환
- `public void setList(List<IHealthCalculator> list) { .. }` // 새 리스트로 지정
- `public void add(IHealthCalculator c) { }` // 리스트에 새로 추가
- `public void insert(int index, IHealthCalculator c) { }` // 리스트에 index위치에 새로 추가
- `public void removeAll() { }` // 전체 리스트 삭제
- `public void remove(IHealthCalculator c) { }` // 리스트에서 해당 c 삭제
- `public void replace(int index, IHealthCalculator c) { }` // 리스트에서 해당 index를 c로 교체
- `public IHealthCalculator get(int index) { }` // 리스트에서 해당 index의 c 반환
- `public List<IHealthCalculator> find(Predicate<? Super IHealthCalculator> predicate) { }` // 리스트에서 해당 predicate(조건)의 리스트 반환
- `public IHealthCalculator getRandom() { }` // 리스트에서 랜덤하게 하나 선택해서 반환
- `public int size() { }` // 리스트의 크기 반환
- `public void sort() { }` // 리스트 정렬

과제 Lab7 (Interface & Collection)

□ HealthCalculatorMapManager 클래스 멤버

- `public Map<Integer, IHealthCalculator> getMap() { .. } // 맵 반환`
- `public void setMap(Map<Integer, IHealthCalculator> map) { .. } // 새 맵으로 지정`
- `public void add(int index, IHealthCalculator c) { } // 맵에 새로 추가`
- `public void removeAll() { } // 전체 맵 삭제`
- `public void remove(int index) { } // index로 맵에서 해당 c 삭제`
- `public void replace(int index, IHealthCalculator c) { } // 맵에서 해당 index를 c로 교체`
- `public IHealthCalculator get(int index) { } // 맵에서 해당 index의 c 반환`
- `public IHealthCalculator getRandom() { } // 맵에서 랜덤하게 하나 선택해서 반환`
- `public int size() { } // 맵의 크기 반환`
- `public List<IHealthCalculator> find(Predicate<? Super IHealthCalculator> predicate) { } // 맵에서 해당 predicate(조건)의 리스트 반환`

과제 제출

- Lab4와 보고서 전체를 묶어서 e-learning에 과제 제출 (due by 5/16)
 - `List<IHealthCalculator> calculators = new ArrayList<>();` 사용
 - `List<PersonData> data = new ArrayList<>();`는 Lab3 data 사용
 - 모든 calculators와 모든 data 를 사용해서, listManager와 mapManager에 add (추가)후 print
 - HealthCalculatorListManager/HealthCalculatorMapManager 의 모든 메소드 테스트
 - List와 map에서 랜덤 선택해서 print
 - List와 map의 sort, find 테스트
 - 그 외 나머지 모든 메소드 테스트
 - 보고서에 HealthCalculatorListManager/HealthCalculatorMapManager 를 비교분석
 - **UserInput과 HealthCalculatorFactory.getInstance(mode)를 사용하여 calculator 생성 및 계산**