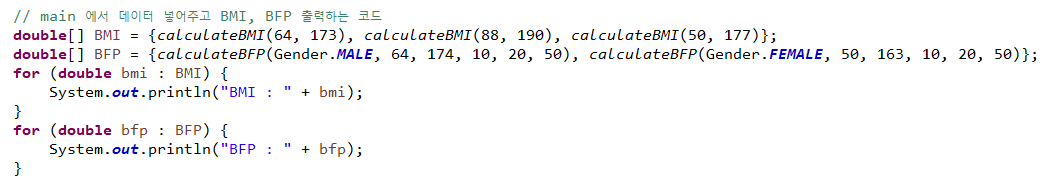
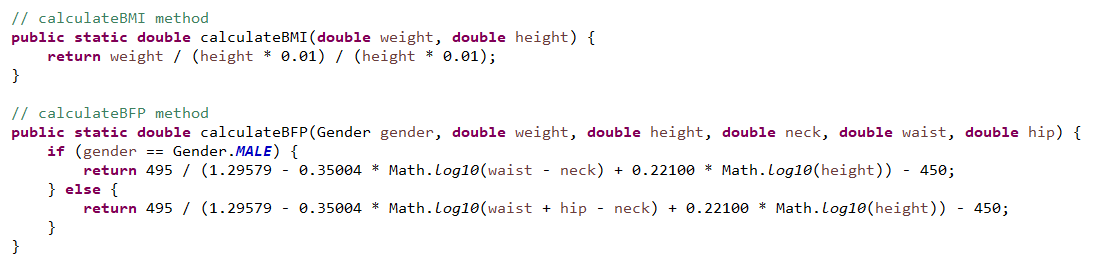
<Lab2 보고서>

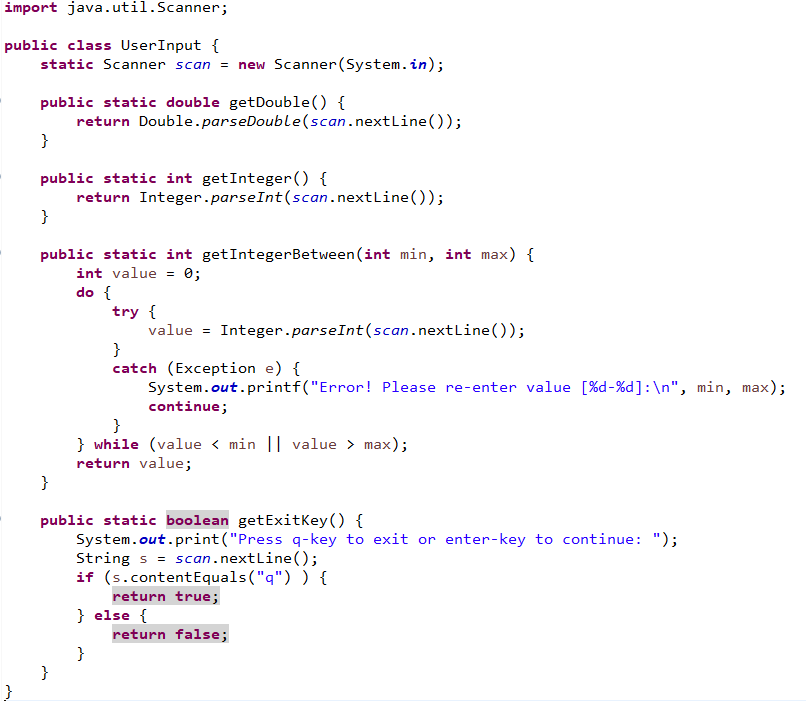
Lab2 / 3분반 / 2022-04-01 / 32191597 / 박민규

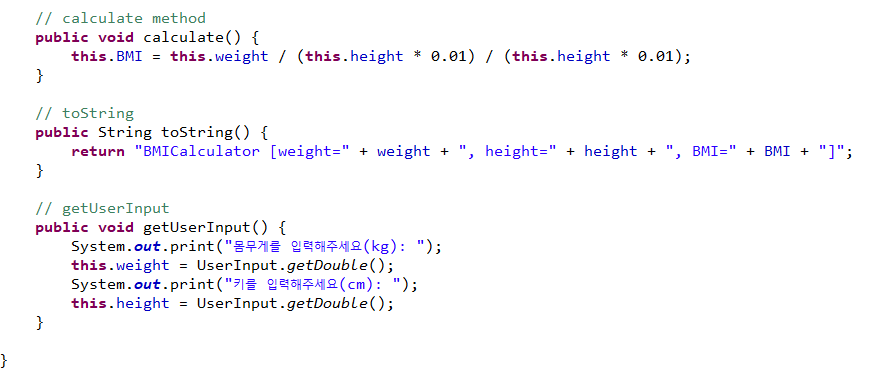
1. **주요 소스 코드 및 설명**
   1. **BMI, BFP method and execute**

****처음으로, Lab2 class 안에 BMI 와 BFP를 계산하는 method를 만들어 줬다. 나중에 이 method를 이용해 계산한 것과 class로 object를 만들어 계산하는 것과의 사용 방식에 대한 비교분석을 할 것이다.

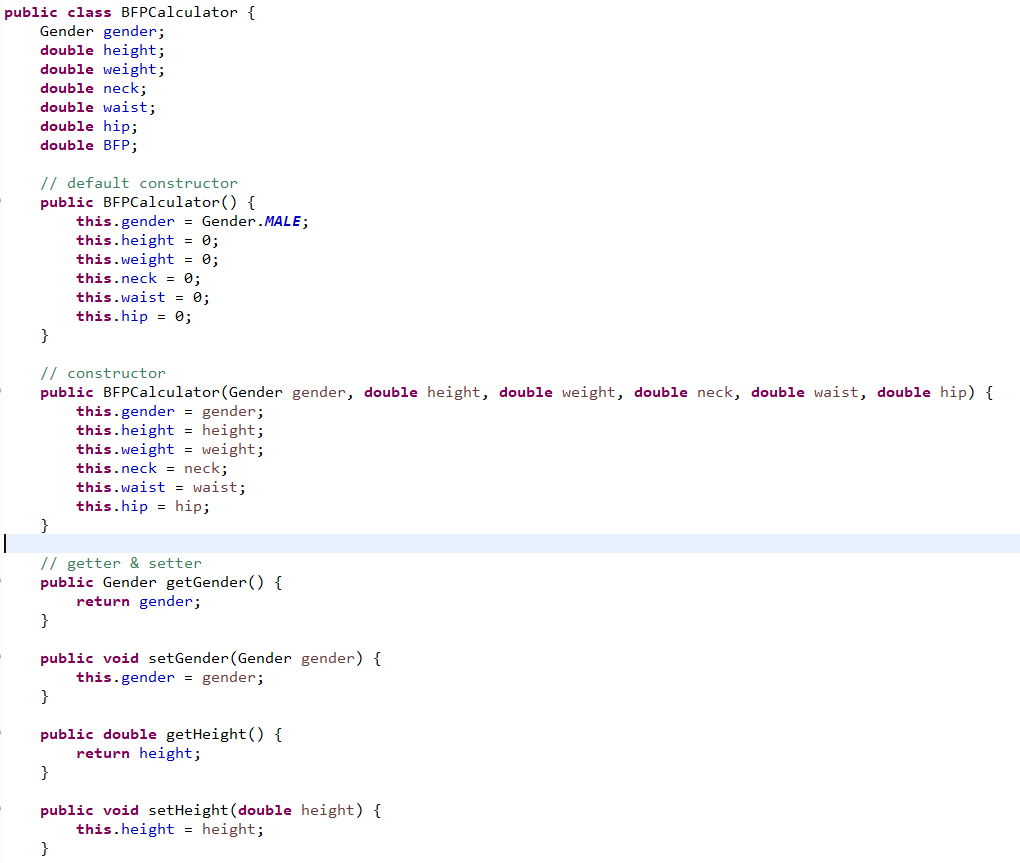
Main class 안에서 BMI 라는 double 형 배열과 BFP라는 double 형 배열을 생성해 줬고, 배열들 안에 method를 이용한 계산 값들을 넣어 주었다. 그 후 foreach 문을 사용하여 배열에 들어가 있는 값들을 출력하도록 했다.

* 1. **UserInput Class**

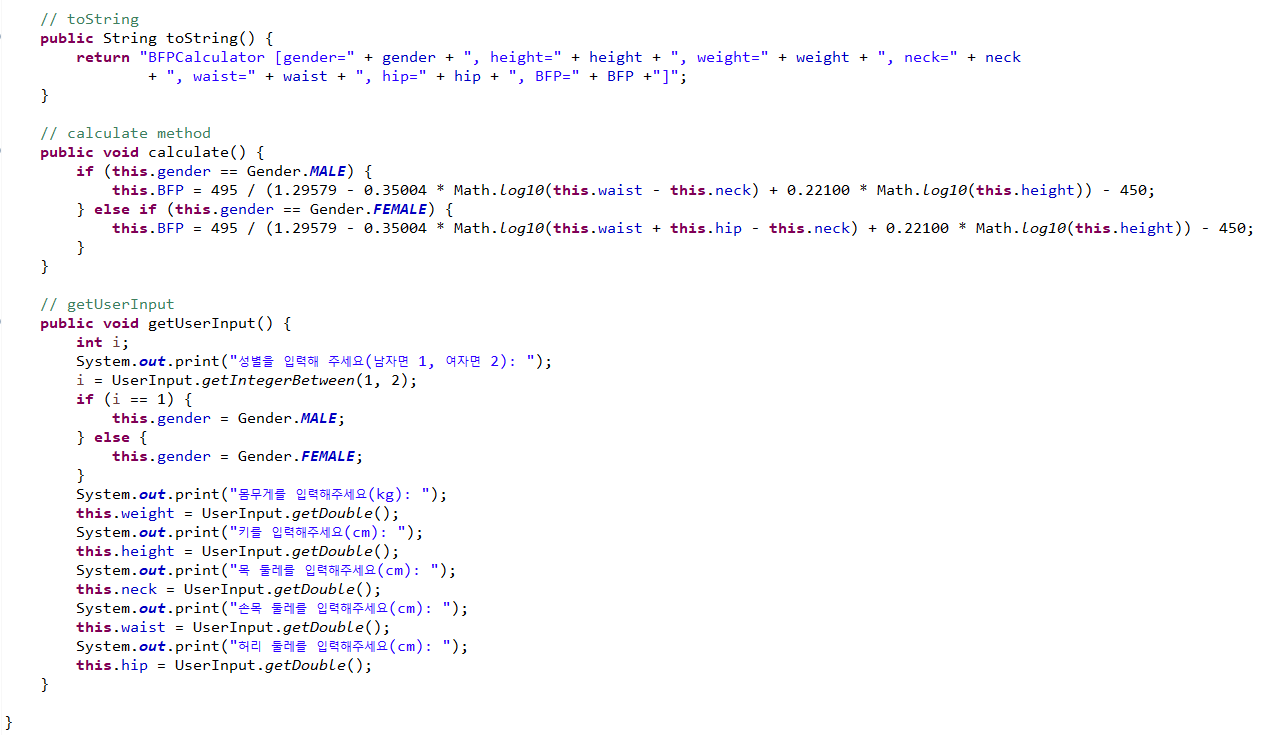
다음으로 살펴볼 것은 “UserInput class” 이다. 이 class에서는 사용자에게 입력을 받는 역할 뿐만 아니라, 설정한 범위 사이의 값이 아닌 값을 입력하면 error를 출력하는 method와, “q”를 입력하면 프로그램이 true를 return 하고 “enter” 을 입력하면 false 를 return하는 method가 포함되어 있다. 이 method 들은 나중에 반복문을 탈출하거나 원하는 명령을 실행하고 싶을 때 사용될 것이다.

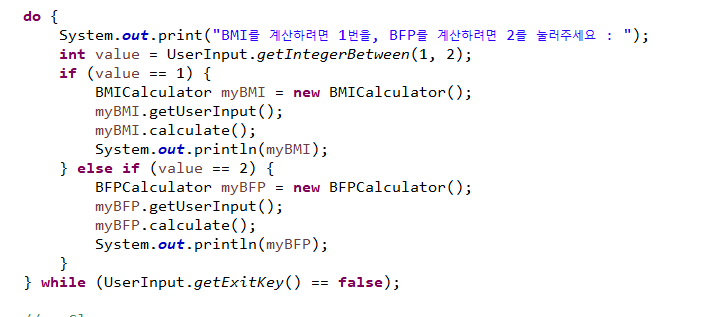
* 1. **BMI calculator class**

다음은 BMI를 계산하기 위해서 BMIcalculator 라는 이름의 class를 생성했다. 이 class 에는 double 형 변수 weight, height, BMI 가 있다. weight, height에 대한 constructor 와 default constructor를 생성해 줬고, getter 와 setter를 생성해 주었다. 이후에 키와 몸무게를 이용한 공식을 통해 계산한 값을 BMI에 할당해 주는 method와, 사용자에게 몸무게와 키의 값을 받아오는 method를 생성해 주었다.

* 1. **BFP calculator class**

****

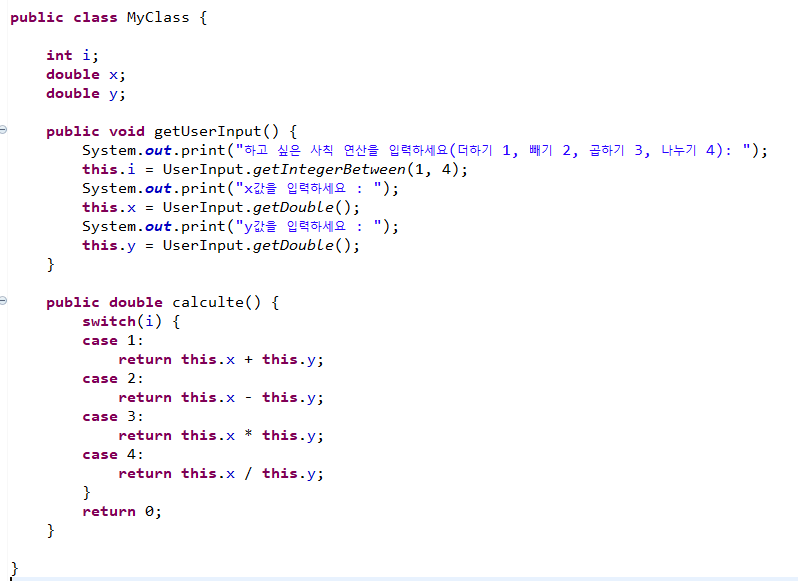
위의**** class는 BFP를 계산하기 위해서 만든 class 이다. BMIcalculator과 달라진 점이라면, 몇 개의 변수가 추가되었다는 점과 gender를 enum형으로 처리한 것이다. 또한 gender에 따라서 계산하는 공식이 달라지기 때문에, calculate method에서 성별에 따라 다른 공식을 이용해 처리한 것을 볼 수 있다. 또한 사용자에게 입력을 받을 때 남자면 1을, 여자면 2를 입력하도록 처리하였고, 이 과정에서 아까 본 UserInput class에 getIntegerBetween method를 사용한 것을 볼 수 있다. 나머지 constructor 와 getter, setter 부분은 BMIcalculator class와 유사하다.

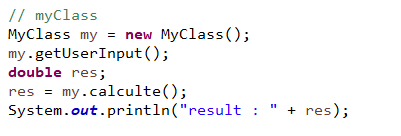
* 1. **Class execute code**

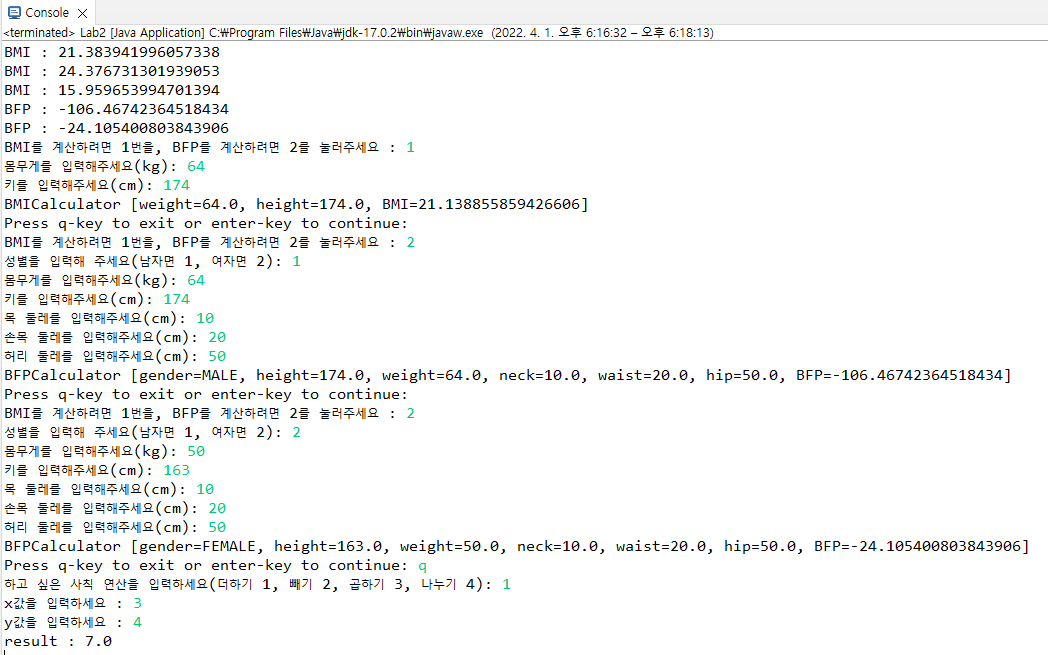
앞서 작성한 BMIcalculator, BFPCalculator 를 실행하는 코드다. 우선 do-while 문을 사용해 UserInput class에 getExitKey method를 통해 받아온 값이 false 일 때, 즉 사용자가 “enter”를 누르는 동안 계속 반복될 수 있도록 설계가 되었고 BMI를 계산하기 위해서는 1번, BFP를 계산하기 위해서는 2번을 누르도록 하였다. 이 과정 역시 UserInput class에 getIntegerBetween method를 사용하여 구현하였다. 이후에 Class를 사용하기 위해 object를 만들었고. 사용자에게 입력을 받아 결과를 출력하는 과정이 반복된다. 사용자가 “enter” 키가 아닌 “q”를 누르게 되면 반복문을 탈출하게 된다.

* 1. **Method vs Class**

Method 와 Class의 사용 방식의 차이점은 method는 적절한 매개변수를 넣어서 바로 사용할 수 있다는 점과, class는 object를 생성한 후에 사용을 해야 한다는 점이다. 사실 java 프로그램 에서는 모든 코드들을 class 안에 작성을 해야 하기 때문에 method라도 결국 어떤 class 안에 들어 있다. 하지만 특정 method와 비슷한 기능을 하는 class를 만든다면, 코드의 재사용성과 더불어 더 다양하고 멋진 기능들을 추가해서 유용하게 사용할 수 있다는 장점이 있을 수 있다. 또한 class를 만든다면 상속, 확장성 면에서 압도적으로 유리하기 때문에 자주 사용할 코드라면 따로 class를 만들어 사용하는 것이 좋을 것 같다.

* 1. **My class**

내가 추가로 작성한 class는 간단한 사칙연산 계산기이다. 코드를 짧게 설명해 보자면 사용자에게 1부터 4까지의 숫자 중 하나의 숫자를 받아와서 어떤 사칙연산을 할 것인지 정하는 method와, 그 값에 따라 계산한 값을 return 해주는 method가 있다. 이 class를 실행하기 위해서 다음과 같이 object를 만든 후, class 안의 method를 사용해서 실행시키면 된다.

1. **실행 화면**

다음은 실행 화면으로, 처음에는 method를 이용한 BMI, BFP 계산 값들이 나오고 다음으로 do-while 문을 통해 class를 이용한 BMI, BFP 계산을 하게 된다. 사용자가 “enter”를 누르면 계속 반복문이 돌고, “q”를 누르면 반복문이 종료되어 밑에 있는 코드가 실행된 것을 볼 수 있다.

1. **결론**

이번 과제를 통해서 객체 지향 프로그램이 어떤 것이구나 하는 것을 확실히 알게 되었고 그것의 장점도 확실히 와 닿게 되었던 과제였던 것 같다. 이제 class를 사용하는 과정이 두렵지 않을 것 같고 java의 재밌는 점을 점점 알게 되어 가는 과제였던 것 같다.