

# Design Pattern

## 최종 보고서

프로젝트명: 운동 계획 관리 프로그램 (농구)  
학번: 201901208  
이름: 김현동

2023-1 학기 디자인패턴

2023. mm. dd

# 1. 프로젝트 개요

## 1) 프로젝트 소개

이 보고서는 "운동 계획 관리 프로그램" 프로젝트를 주제로 합니다. 개인이 운동 계획을 세우기란 쉽지 않습니다. 특히 이번 프로젝트는 농구를 중심으로 두며 대상은 이렇습니다. 농구를 처음 시작하는 사람. 포지션 변경을 원하는 사람. 현재 자신의 포지션과 몸의 상태를 확인하고 싶은 사람. 변화를 원하지만 방향성을 정하지 못한 점에서 공통점을 가집니다. 이러한 대상에게 방향성을 제시하는 프로그램입니다.

# 2. 목표 및 내용

## 1) 최종 목표

기능

1. 사용자의 정보를 수집합니다. (키,몸무게,근육량,포지션,부상,공 친화도)
2. 사용자의 포지션에 이상적인 몸의 상태를 계산합니다.
  - 3-1. 사용자의 몸의 상태가 이상적이지 않다면 웨이트를 일정에 추가합니다.
  - 3-2. 사용자의 몸의 상태가 이상적이지만 공 친화도가 높지 않다면 볼 컨트롤 훈련을 추가합니다.
  - 3-3. 사용자의 몸의 상태가 이상적이고 공 친화도 또한 높다면 기본 훈련만 진행합니다.
4. 몸의 상태, 공 친화도를 높이기 위한 훈련은 여러가지입니다. 변화가 없는동안 여러부위의 웨이트, 여러 종류의 볼 컨트롤 훈련을 진행합니다.
5. `observer` 가 사용자 상태 변화를 기다립니다. 변화가 생기면 2~4 번을 반복하여 필요하면(신체 상태가 변경되면) 계획을 다시 세워줍니다.

## 2) 적용된 패턴의 종류 및 방법

### 1. builder pattern:

- `PositionSpacInfo`클래스에서 `GoalBuilder`를 사용하여 복잡한 객체 생성 과정을 단순화하고 있습니다.
- `GoalBuilder`는 `setHeight()`, `setWeight()`, `setMuscleMass()`, `setPosition()`등의 메서드를 제공하여 필요한 속성을 설정한 뒤, `buildGoal()`메서드를 호출하여 `PositionSpacInfo`객체를 반환합니다.

### 2. template method pattern:

- `BasicPlan`클래스가 템플릿 역할을 수행합니다.
- `exercisePlan()`메서드는 `warmUp()`, `mainEx()`, `coolDown()`, `addPlan()`등의 단계로 나누어진 일련의 작업을 수행합니다. 각 단계는 서브클래스에서 오버라이드할 수 있습니다.
- `BodyPlan`, `SkillPlan`, `injuryPlan`클래스들은 `BasicPlan`을 상속받아 필요한 메서드를 오버라이드하여 구체적인 동작을 정의합니다.

### 3. Observer 패턴:

- `Subject`와 `Observer`클래스를 정의하고, `Player`클래스가 `Subject`를 상속받아 구현되어 있습니다.
- `Player`클래스는 `attach()`, `detach()`, `notify()`메서드를 구현하여 옵저버(`Observer`) 객체들을

추가하고 제거하며, 변경 사항이 있을 때 옵저버들에게 알립니다.

- Observer인터페이스를 구현한 클래스들은 update()메서드를 구현하여 옵저버 패턴의 구독자 역할을 수행합니다.

#### 4.Facade Pattern

- SetPlan클래스가 퍼사드 역할을 수행합니다.

SetPlan은 controlAddPlan()메서드를 통해 Player와 Goal객체를 인자로 받고, 복잡한 계산과 로직을 감추고 간단한 인터페이스를 제공합니다.

### 3) 주요 코드 및 설명

1. Player class: 사용자의 정보를 저장하고, 정보 변경 시 옵저버에게 알림을 보내는 역할을 합니다. 사용자의 키, 몸무게, 근육량, 포지션, 부상 여부, 공 친화도 등의 정보를 저장하며, attach, detach, notify메서드를 구현하여 옵저버 패턴을 사용합니다. set 함수로 이후 사용자의 정보 변경시 반영합니다.
2. PositionSpacInfo class: 사용자의 키, 몸무게, 근육량, 포지션 정보를 받아 목표 몸무게와 목표 근육량을 계산하고, 현재 몸무게와 근육량과의 차이를 계산합니다.
3. GoalBuilder class: PositionSpacInfo클래스의 인스턴스를 생성하기 위한 빌더 패턴을 구현합니다. 사용자의 목표 키, 몸무게, 근육량을 포지션에 맞게 설정해줍니다.
4. BasicPlan class: 기본 운동 계획을 생성하는 템플릿 패턴을 구현합니다. 워밍업, 메인 운동, 쿨다운 운동 등의 기본 운동 리스트를 저장하고, 이를 조합하여 운동 계획을 생성합니다.
5. BodyPlan, SkillPlan, InjuryPlan class: BasicPlan클래스를 상속받아 특정 운동 계획을 생성하는 클래스입니다. 각각 전신 운동, 스킬 트레이닝, 부상 관리 계획을 생성합니다.
6. SetPlan class: Player의 정보와 PositionSpacInfo의 계산 결과를 기반으로 적절한 운동 계획을 선택합니다. 신체 조건을 만족해야 ball controll 훈련을 시작합니다.  
Injury > Body Condition > Ball Controll
7. Service class: 사용자와 운동 계획 간의 중개 역할을 합니다. 사용자 정보 변경 시 Player객체를 수정하고, PositionSpacInfo객체를 생성하여 운동 계획을 설정합니다. 또한, 옵저버로 등록하여 사용자 정보 변경 시 운동 계획을 업데이트합니다.
8. UI: tkinter를 사용하여 사용자 인터페이스를 제공합니다. player 생성에 필요한 정보를 수집하고, 목표 몸 상태, 운동 계획을 사용자에게 보여주는 공간입니다.

## 3. 결과 및 느낀점

### 1) 결과 화면 및 설명

정보에 따른 계획 수립	<div> <div> <div>운동 계획 관리 프로그램 (농구)</div> <div> <div>Height 180</div> <div>Weight 70</div> <div>Muscle Mass 33</div> <div>Position Guard</div> <div>Ball Control <input type="checkbox"/></div> <div>Injury <input type="checkbox"/></div> <div>Goal</div> <div>Next Day(Condition No change)</div> </div> </div> <div> <div>Guard's Goal Condition</div> <div>Goal Weight: Gap with Best Spac of Guard : weight '63.36' muscleMass '36.54'</div> <div>-6.64kg 만큼 체중감량이 필요합니다. 3.54kg만큼 근육을 키워야합니다.</div> </div> <div> <div>Exercise Plan:</div> <div> <div>Warm-up start</div> <div>1. High knees</div> <div>2. Butt kicks</div> <div>3. Lunges Leg Swings</div> <div>4. Jogging in Place</div> <div>Main exercise start</div> <div>1. Lay up 20/20</div> <div>2. Three steps</div> <div>3. Sprint in Place</div> <div>4. Dribbel drill</div> <div>5. strategy scrimmage</div> <div>Cool-down start</div> <div>1. Hamstring Stretches</div> <div>2. Calf Stretches</div> <div>3. Hip Flexor Stretches</div> <div>4. Static Stretches</div> <div>Plus Plan start</div> <div>1. Squat</div> <div>2. Bench press</div> <div>3. Shoulder press</div> <div>4. Deadlift</div> </div> </div> </div>
--------------	---

신체 조건 만족 불훈련	<div>운동 계획 관리 프로그램 (농구)</div> <div> <div> Height: 180  Weight: 65  Muscle Mass: 36  Position: Guard  Ball Control: <input type="checkbox"/>  Injury: <input type="checkbox"/>  Goal  Next Day(Condition No change) </div> <div> ('Guard', 180.0, 65.0, 36.0, False, False)'s Goal Condition  Guard's Goal Condition  Goal Weight: Gap with Best Spac of Guard : weight '63.36' muscleMass '36.54' </div> <div> Exercise Plan:  Warm-up start  1. High knees  2. Butt kicks  3. Lunges Leg Swings  4. Jogging in Place  Main exercise start  1. Lay up 20/20  2. Three steps  3. Sprint in Place  4. Dribbel drill  5. strategy scrimmage  Cool-down start  1. Hamstring Stretches  2. Calf Stretches  3. Hip Flexor Stretches  4. Static Stretches  Plus Plan start  1. Cone Dribble and Finish  2. Two-Ball Dribble  3. 1-on-1 Dribble Attack  4. Dribble Combo Moves </div> </div>
몸, 볼 컨트롤 충족	<div>운동 계획 관리 프로그램 (농구)</div> <div> <div> Height: 180  Weight: 65  Muscle Mass: 35  Position: Guard  Ball Control: <input checked="" type="checkbox"/>  Injury: <input type="checkbox"/>  Goal  Next Day(Condition No change) </div> <div> ('Guard', 180.0, 65.0, 35.0, False, True)'s Goal Condition  Guard's Goal Condition  Goal Weight: Gap with Best Spac of Guard : weight '63.36' muscleMass '36.54' </div> <div> Exercise Plan:  Warm-up start  1. High knees  2. Butt kicks  3. Lunges Leg Swings  4. Jogging in Place  Main exercise start  1. Lay up 20/20  2. Three steps  3. Sprint in Place  4. Dribbel drill  5. strategy scrimmage  Cool-down start  1. Hamstring Stretches  2. Calf Stretches  3. Hip Flexor Stretches  4. Static Stretches </div> </div>
부상	<div>운동 계획 관리 프로그램 (농구)</div> <div> <div> Height: 180  Weight: 65  Muscle Mass: 34  Position: Guard  Ball Control: <input checked="" type="checkbox"/>  Injury: <input checked="" type="checkbox"/>  Goal  Next Day(Condition No change) </div> <div> ('Guard', 180.0, 65.0, 35.0, True, True)'s Goal Condition  Guard's Goal Condition  Goal Weight: Gap with Best Spac of Guard : weight '63.36' muscleMass '36.54' </div> <div> Exercise Plan:  focus on your recovery  go to hospital and get professional treatment  The most important thing is not to be injured </div> </div>

## 2) 느낀 점

코드를 작성할 때 가장 중요하게 고려한 점은 "해당 기능에 적절한 설계 패턴을 활용했는가"였습니다. 이전에는 각 기능을 개별적으로 구현하는 코딩은 많이 해보았지만, 모든 코드가 조화롭게 작동해야 하는 코드를 작성하는 것은 처음이었기 때문에 어려움을 많이 겪었습니다. 각 클래스의 기능이 잘 작동하더라도 연동하는 과정에서 많은 오류가 발생했습니다. 그러나

문제를 해결하면서 설계 패턴을 활용한 효과를 많이 경험했습니다. 패턴을 사용하면 각 클래스의 기능을 확실하게 정의할 수 있다는 것을 느꼈기 때문입니다. 그 결과, 코드의 명확성을 높일 수 있었습니다.

하지만 다양한 설계 패턴을 적용하려다 보니 오히려 코드가 더 복잡해지는 경우도 있었습니다. 그럴 때는 결단을 내리고 패턴을 제거하여 구현하려던 기능을 방해하지 않도록 했습니다. 이러한 조정 과정을 통해 코드를 더욱 단순화할 수 있었습니다.

학기 초에 다루었던 내용 중 단일 책임 원칙도 실감하게 되었습니다. 객체 지향 프로그래밍이 서툴러서 한 함수에 모든 것을 해결하려는 습관이 있었는데, 이로 인해 수정하는 데 많은 시간과 노력을 투자해야 했습니다. 이런 경험을 통해 저는 단일 책임 원칙을 잘 준수하여 코드를 작성하는 것의 중요성을 깨달았습니다. 또한, 이러한 관점에서 facade 패턴을 활용할 수 있는 기회를 가졌습니다.

이 프로젝트는 내가 작성한 코드의 품질을 평가하고 개선하는 기회였습니다. 긴 코드를 작성하면서 각 기능을 명확하고 단순하게 유지하는 필요성을 더욱 느낄 수 있었습니다. 나 자신의 나쁜 습관이 있는지, 더 나은 코드란 무엇인지 고민하는 계기가 되어 매우 유익한 경험이었습니다.