



Google Developer Student Clubs
Hanyang University

App Front 2주차 과제 풀이

```
lookup.KeyValue  
f.constant(['en  
=tf.constant([G  
.lookup.StaticV  
_buckets=5)
```

과제 1

enum 의 장점은 무엇인가요?

enum 을 사용하는 이유는 무엇인가요? 어떤 장점이 있나요?

과제 1 - 답변

가독성 향상

코드를 더 읽기 쉽게 만든다. enum 을 사용하면 관련된 상수값들을 그룹화 할 수 있기 때문에, 코드에서 의미 있는 이름을 사용하여 상수값을 구성하면 식별하기가 쉬워진다.

유형 안정성

enum 는 컴파일러가 이해하고 검사할 수 있는 유형을 가지고 있어서, 올바르지 않은 유형의 값을 enum 변수에 할당하려고 시도하면 컴파일 오류를 발생시키기 때문에 프로그램 런타임 오류를 방지하는데 도움이 된다.

코드의 일관성

enum 으로 정의하면 모든 개발자가 동일한 값 집합을 공유하기 때문에 코드의 일관성이 유지된다.

요약하면 enum 코드의 가독성, 유지 관리성, 안전성을 향상시키고 개발자가 코드 작성 시 실수를 줄여주는 프로그래밍 도구이다!

과제 1 - 답변

코드 유지 관리 용이성

enum 을 사용하면 관련된 상수값을 하나의 위치에 정의하므로, 상수값이 변경되어야 할 때, 해당 위치만 수정하면 되므로 코드 유지 관리가 간편해진다.

자동 완성 및 IDE 지원

대부분의 IDE는 enum 을 지원하여 해당 enum 유형의 상수값을 자동으로 완성하거나 추천합니다. 이로 인해 개발자가 오타를 줄이고 빠르게 작업할 수 있다.

코드 의미 전달

enum 을 사용하면 상수값에 의미를 부여할 수 있기 때문에 의미 전달을 명확하게 할 수 있다. 예시로 색상을 나타내는 enum 을 사용하면 RED , BLUE , GREEN 과 같은 이름으로 색상을 명확하게 표현할 수 있다.

과제 2 - 답변

2번 문제

1. `People` 클래스는 이름(`name`)과 나이(`age`)를 가진 사람을 나타내는 기본 클래스입니다. `sayHello` 메서드는 이름과 나이를 출력합니다.
2. `Team` enum은 여러 팀을 나타내는 enum입니다. (Enum의 상수는 자유롭게 하시면 됩니다)
3. `Student` 클래스는 `People` 클래스를 상속합니다. 추가로 `team` 멤버 변수를 가지며, 생성자에서 `name`, `age`, `**team**`을 필수 매개변수로 받습니다. 부모 클래스인 `**People**`의 생성자를 호출하여 `**name**`과 `**age**`를 초기화합니다.
4. `Student` 클래스에서는 `sayHello` 메서드를 재정의하여 부모 클래스의 `**sayHello**`를 호출한 후 `team` 멤버 변수의 값을 출력합니다.
5. `main` 함수에서 `Student` 객체를 생성하고 `sayHello` 메서드를 호출합니다. 이로 인해 학생의 이름, 나이 및 팀이 출력됩니다.

<출력 예시>

(출력 예시는 예시일 뿐 값을 필요는 없습니다!!)

```
안녕하세요. 김지송, 21 살 입니다.  
파트는 app입니다
```

과제 2 - 답변

1. People 클래스는 이름(name)과 나이(age)를 가진 사람을 나타내는 기본 클래스입니다. sayHello 메서드는 이름과 나이를 출력합니다.

```
class People {  
    final String name;  
    int age;  
  
    People({required this.name, required this.age});  
  
    void sayHello() {  
        print('안녕하세요. $name, $age 살 입니다.');    }  
}
```

과제 2 - 답변

2. `Team` enum은 여러 팀을 나타내는 enum입니다. (Enum 의 상수는 자유롭게 하시면 됩니다)

```
enum Team {app, web, server, m1d1}
```

과제 2 - 답변

3. **Student** 클래스는 **People** 클래스를 상속합니다. 추가로 **team** 멤버 변수를 가지며, 생성자에서 **name**, **age**, **team**을 필수 매개변수로 받습니다. 부모 클래스인 **People**의 생성자를 호출하여 **name**과 **age**를 초기화합니다.

4. **Student** 클래스에서는 **sayHello** 메서드를 재정의하여 부모 클래스의 **sayHello**를 호출한 후 **team** 멤버 변수의 값을 출력합니다.

```
class Student extends People {
    Team team;

    Student({
        required this.team,
        required String name,
        required int age,
    }): super (
        name : name,
        age : age);

    @override
    void sayHello() {
        super.sayHello();
        print('파트는 ${team.name}입니다');
    }
}
```


과제 2 - 답변

5. **main** 함수에서 **Student** 객체를 생성하고 **sayHello** 메서드를 호출합니다. 이로 인해 학생의 이름, 나이 및 팀이 출력됩니다.

```
void main() {  
    var student = Student(  
        name : "김지송",  
        age : 22,  
        team : Team.app  
    );  
  
    student.sayHello();  
}
```