

Java Programming II

Lab5

514770-1

Fall 2024

10/6/2026

Kyoung Shin Park
Computer Engineering
Dankook University

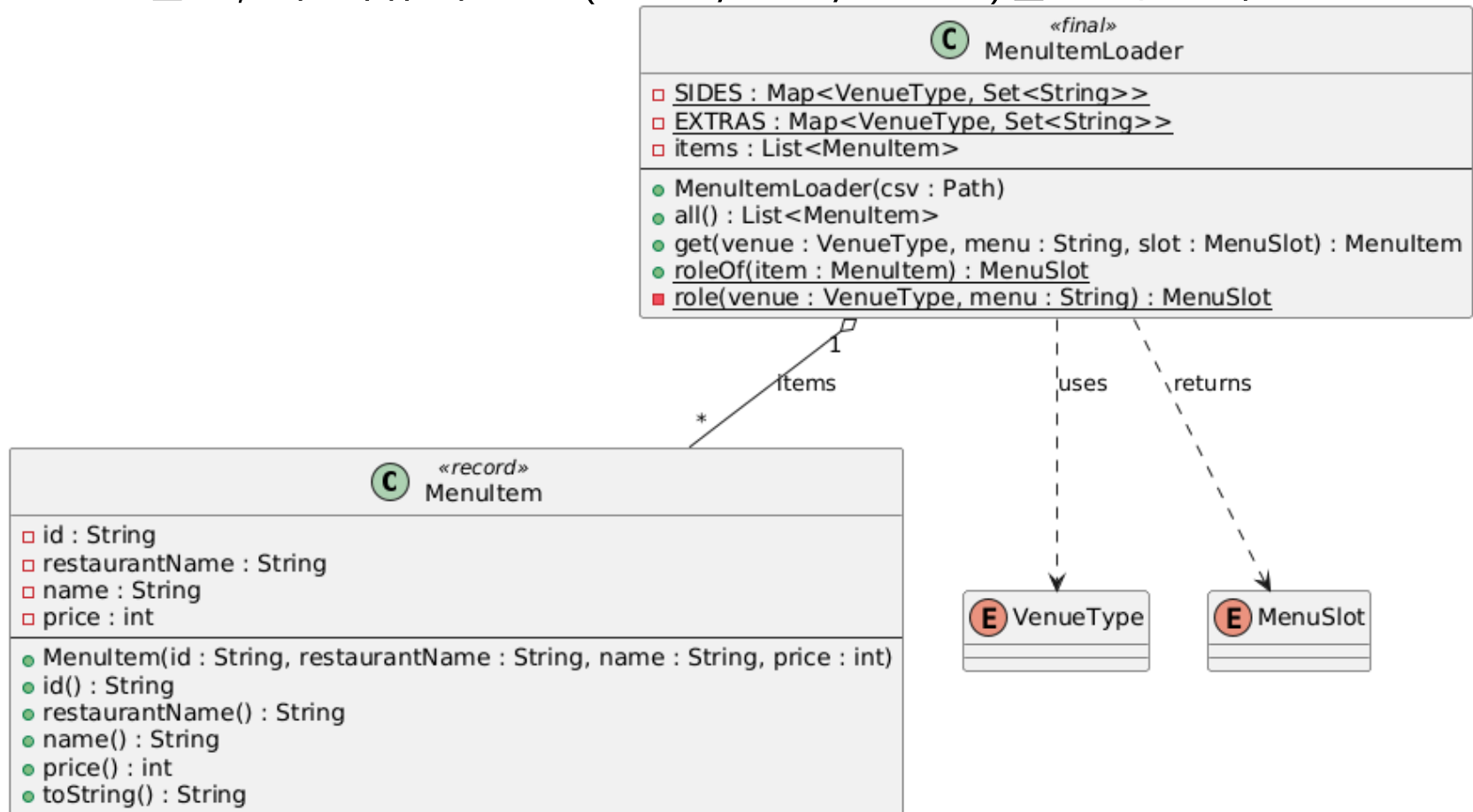
Lab5

- ▣ Factory & Builder 패턴을 사용하여 학생식당의 국밥(51장국밥), 덮밥(바비든든), 분식(폭풍분식), 중식(중식대장) 코너별 한끼 세트와 주간 식단을 생성하는 프로그램을 작성하라.
- ▣ **menus.csv 파일 사용**

Lab5

□ MenuItem & MenuItemLoader 클래스

- MenuItemLoader 클래스는 menus.csv을 읽어 MenuItem 목록을 만들고, 각 메뉴의 슬롯(MAIN/SIDE/EXTRA)를 판정한다.



Lab5

```
public enum MenuSlot {
    MAIN, SIDE, EXTRA
}

public enum VenueType {
    GUKBAP("51장국밥"),
    COVER("바비든든"),
    BUNSIK("폭풍분식"),
    CHINESE("중식대장");
    final String name;
    VenueType(String name) {        this.name = name;    }
    static VenueType ofRestaurant(String restaurantName) {
        for (VenueType v : values()) {
            if (v.name.equals(restaurantName)) return v;
        }
        return null;
    }
}
```

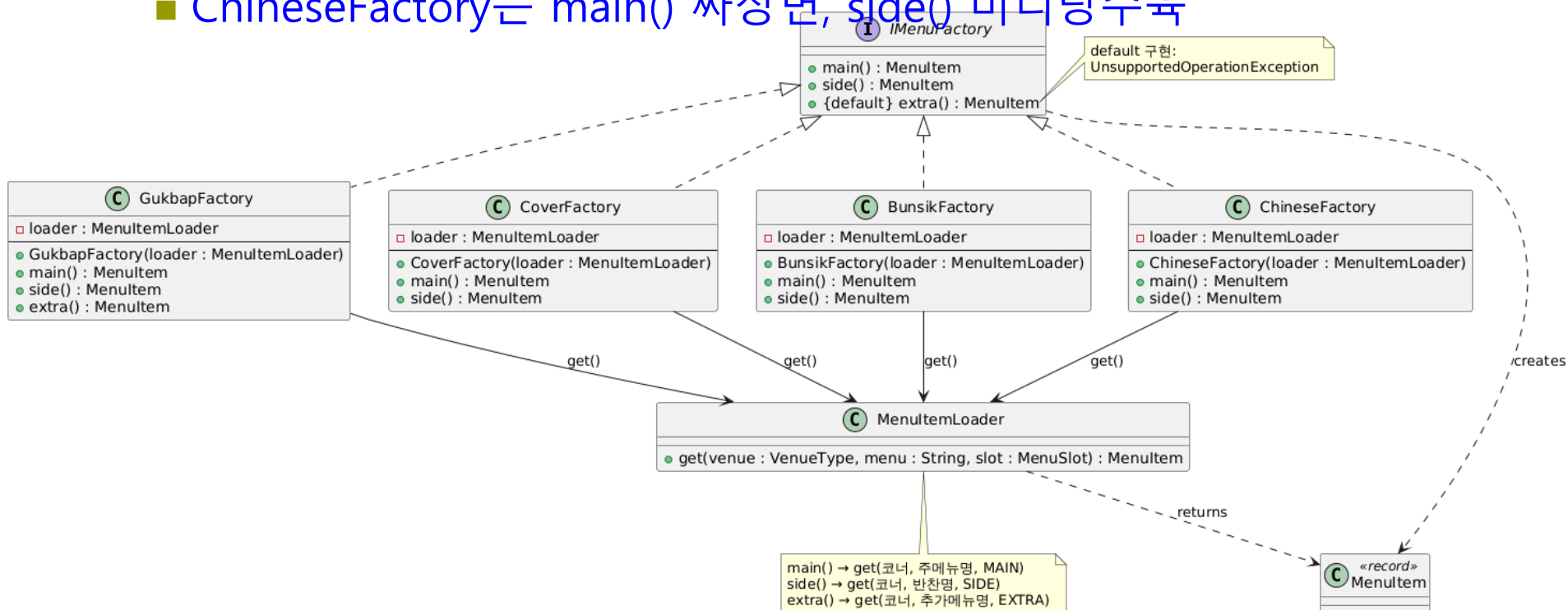
Lab5

□ MenuItemLoader 클래스

- **생성자**는 CSV 각 행을 읽어 MenuItem으로 만들고 items에 저장한다.
- **all()**은 전체 메뉴를 수정할 수 없는 목록으로 반환한다.
- **get(venue, menu, slot)**는 **Factory가 메뉴를 꺼낼 때**, 매장 이름과 메뉴 이름으로 항목을 찾은 뒤, 실제 역할이 요청한 슬롯과 같은지 확인한다. 메뉴가 없거나 슬롯이 다르면 예외를 던진다.
- **roleOf(item)**는 **Builder가 받은 메뉴를 검사할 때**, csv에 없는 역할 정보를 반환한다. MenuItem의 매장 이름으로 VenueType을 찾고, 메뉴 이름에 따라 역할을 판정한다.
- **role(venue, menu)**는 역할 판정 규칙. SIDES에 등록된 메뉴는 SIDE, EXTRAS에 등록된 메뉴는 EXTRA, 나머지는 MAIN으로 본다.

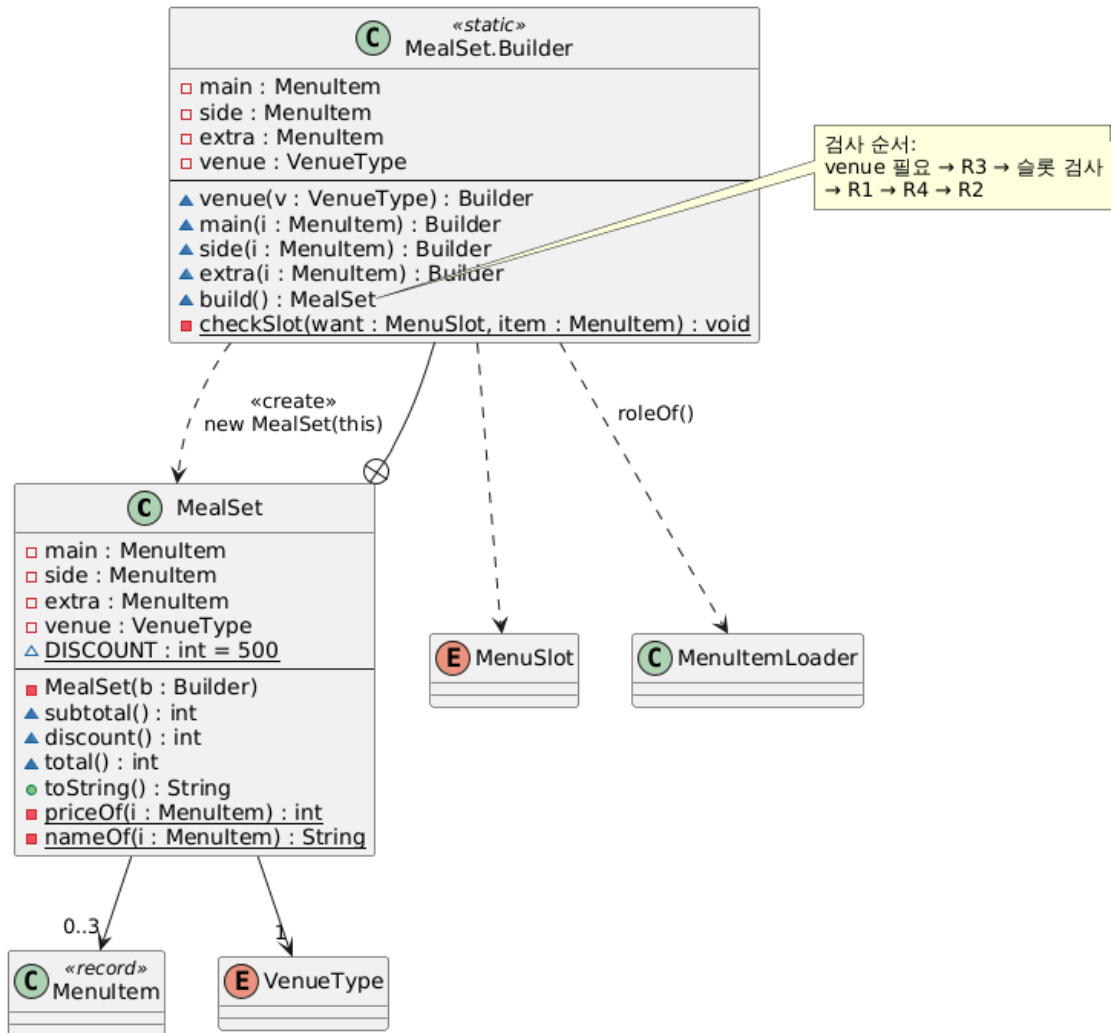
Lab5

- Factory 클래스들은 코너별 "대표 한 끼" 메뉴 묶음 생성
 - GukbapFactory는 main() 닭곰탕(밥 포함), side() 왕만두, extra() 편육한접시
 - CoverFactory는 main() 제육덮밥, side() 고기든든
 - BunsikFactory는 main() 폭풍라면, side() 라볶이
 - ChineseFactory는 main() 짜장면, side() 미니탕수육



Lab5

MenuSet & MenuSet.Builder 클래스



Lab5

□ MealSet – 완성된 한끼 세트

- **생성자**는 주메뉴, 반찬, 추가 메뉴, 매장을 받아 객체의 필드에 저장한다.
- **subtotal()**은 main, side, extra 가격을 더한다.
- **discount()**은 주메뉴와 반찬이 모두 있을 때 DISCOUNT 금액을 반환하고, 둘 중 하나라도 없으면 0을 반환한다.
- **total()**은 subtotal()에서 discount()를 뺀 최종 금액을 반환한다.
- **toString()**은 할인 여부에 따라 금액 표시 형식이 달라진다.
 - 할인이 있으면 소계 - 할인 = 합계원으로 계산 과정을 보여 주고, 없으면 합계만 보여준다.

Lab5

□ MealSet.Builder – MealSet 규칙 검사 담당

- **venue(), main(), side(), extra()**는 값을 저장하고 this를 반환

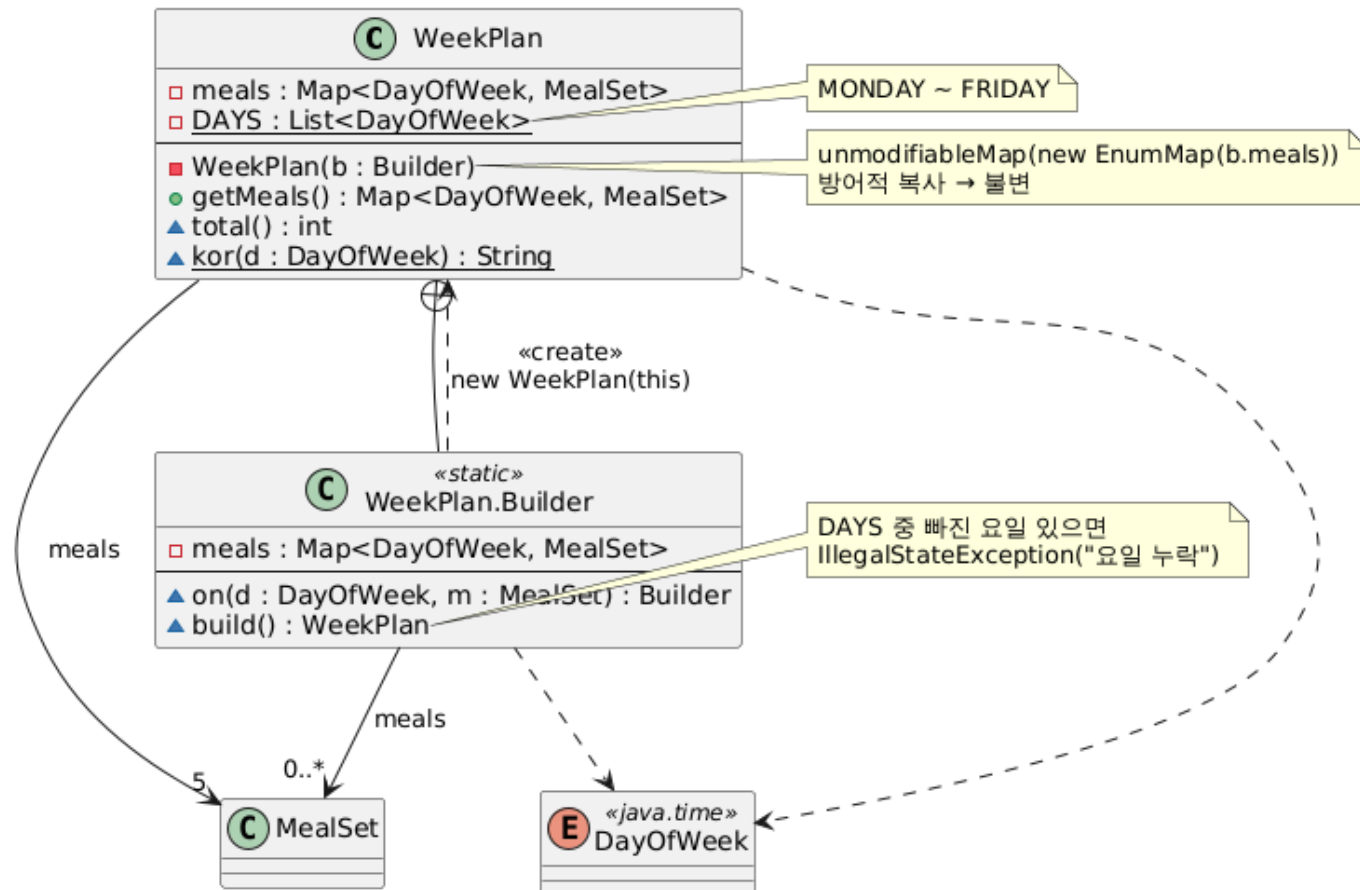
- **build()**은 다음 순서대로 규칙 검사를 진행한다.

1. venue 필수: venue가 없으면 이후의 어떤 규칙도 판단할 수 없으므로 가장 먼저 확인
2. R3 (바비든든 EXTRA 금지): 코너가 COVER인데 extra가 채워져 있으면 곧바로 거부
3. slot 검사: checkSlot()이 각 자리에 놓인 메뉴의 실제 역할을 MenuItemLoader.roleOf()로 확인
4. R1 (최소 구성): 주메뉴가 있어야 하고, 반찬이나 추가메뉴 중 하나 이상이 있어야 함.
5. R4 (코너 일치): 모든 메뉴의 restaurantName이 해당 코너 이름과 같아야 함.
6. R2 (국밥 8,000원 이하): new MealSet(this)로 객체를 실제로 만든 뒤에 검사. 할인이 반영된 total()이 있어야 판단할 수 있기 때문. 규칙을 어기면 만든 객체를 반환하지 않고 예외를 던지므로, 규칙 위반 세트는 밖으로 나가지 않음.

Lab5

□ WeekPlan & WeekPlan.Builder 클래스

- 완성된 주간(월~금) 식단, 이때 빌더는 요일 검사 담당 (월~금 중 빠진 요일이 있으면 생성 불가)



Lab5

```
public class WeekPlan {
    private final Map<DayOfWeek, MealSet> meals;
    private static final List<DayOfWeek> DAYS = List.of(
        DayOfWeek.MONDAY, DayOfWeek.TUESDAY, DayOfWeek.WEDNESDAY,
        DayOfWeek.THURSDAY, DayOfWeek.FRIDAY);
    private WeekPlan(Builder b) {
        // 방어적 복사 - build() 이후 Builder를 계속 써도 WeekPlan은 불변
        this.meals = Collections.unmodifiableMap(new
EnumMap<>(b.meals));
    }
    public Map<DayOfWeek, MealSet> getMeals() { return meals; }
    int total() { return
meals.values().stream().mapToInt(MealSet::total).sum(); }
    static String toKorean(DayOfWeek d) {
        return switch (d) {
            case MONDAY -> "월"; ...                default -> d.name();
        };
    }
}
```

Lab5

```
// 1) 코너별 IMenuFactory가 대표 메뉴(MAIN / SIDE / EXTRA)를 공
급
// 2) MealSet.Builder가 R1~R4 규칙을 검사해 한 끼 세트를 조립
// 3) WeekPlan.Builder가 월~금 누락을 검사해 주간 식단을 조립
public class App {
    public static void main(String[] args) {
        // 모든 메뉴 데이터의 출처
        MenuItemLoader loader = new
MenuItemLoader(Path.of("menus.csv"));
// 1) Factory
// EnumMap → VenueType 선언 순서(국밥, 덮밥, 분식, 중식)로 순회
Map<VenueType, IMenuFactory> fac = new EnumMap<>(Map.of(
    VenueType.GUKBAP, new GukbapFactory(loader),
    VenueType.COVER, new CoverFactory(loader),
    VenueType.BUNSIK, new BunsikFactory(loader),
    VenueType.CHINESE, new ChineseFactory(loader)));
```

Lab5

```
IMenuFactory guk = fac.get(VenueType.GUKBAP),  
    cover = fac.get(VenueType.COVER),  
    bunsik = fac.get(VenueType.BUNSIK),  
    chinese = fac.get(VenueType.CHINESE);
```

```
// 코너별 대표 메뉴 출력
```

```
// 2. Builder로 세트 조립 (성공)
```

```
// 국밥 주메뉴 + 반찬 → 500원 할인, R2(≤8000) 통과
```

```
MealSet lunch = set(VenueType.GUKBAP, guk.main(),  
guk.side(), null);
```

```
System.out.println("\n[성공] " + lunch);
```

```
// 2. MealSet.Builder로 세트 조립 (실패)
```

```
// R3 위조 EXTRA - 바비든든 주+반 세트에 가짜 EXTRA 추가
```

```
// R2 초과 - 주+반+추 구성으로 국밥 세트 8천원 초과
```

```
// R1 부족 - 주메뉴만 있고 반찬, 추가 모두 없음
```

```
// R4 매장 불일치 - 중식세트에 국밥 코너 반찬 넣음
```

```
// slot 불일치 - 반찬을 주메뉴 자리에 배치
```

Lab5

```
// 3. WeekPlan.Builder 주간식단 조립 (성공)
// 요일별 세트는 각각 MealSet.Builder 검사를 이미 통과한 상태
WeekPlan week = new WeekPlan.Builder()
    .on(MONDAY, lunch)
    .on(TUESDAY, set(VenueType.BUNSIK,
bunsik.main(), bunsik.side(), null))
    .on(WEDNESDAY, set(VenueType.COVER,
cover.main(), cover.side(), null))
    .on(THURSDAY, set(VenueType.CHINESE,
chinese.main(), chinese.side(), null))
    .on(FRIDAY, set(VenueType.GUKBAP,
        loader.get(VenueType.GUKBAP, "닭칼국
수", MenuSlot.MAIN), guk.side(), null))
    .build();
// 3. WeekPlan 검사: 월요일만 넣으면 화~금 누락 build() 실패
...
}
```

Submit to e-learning

- ▣ Lab5 과제에 yourcode를 추가 (yourcode 없을시 10점에
서 -1점 감점)
- ▣ **Java26-5-HW5-YourID-YourName.zip** 과제(보고서에
반드시 yourcode 설명 포함)를 e러닝에 제출 (**due by
10/12**).