

Java Programming II

Lab4

514770-1

Fall 2026

9/29/2026

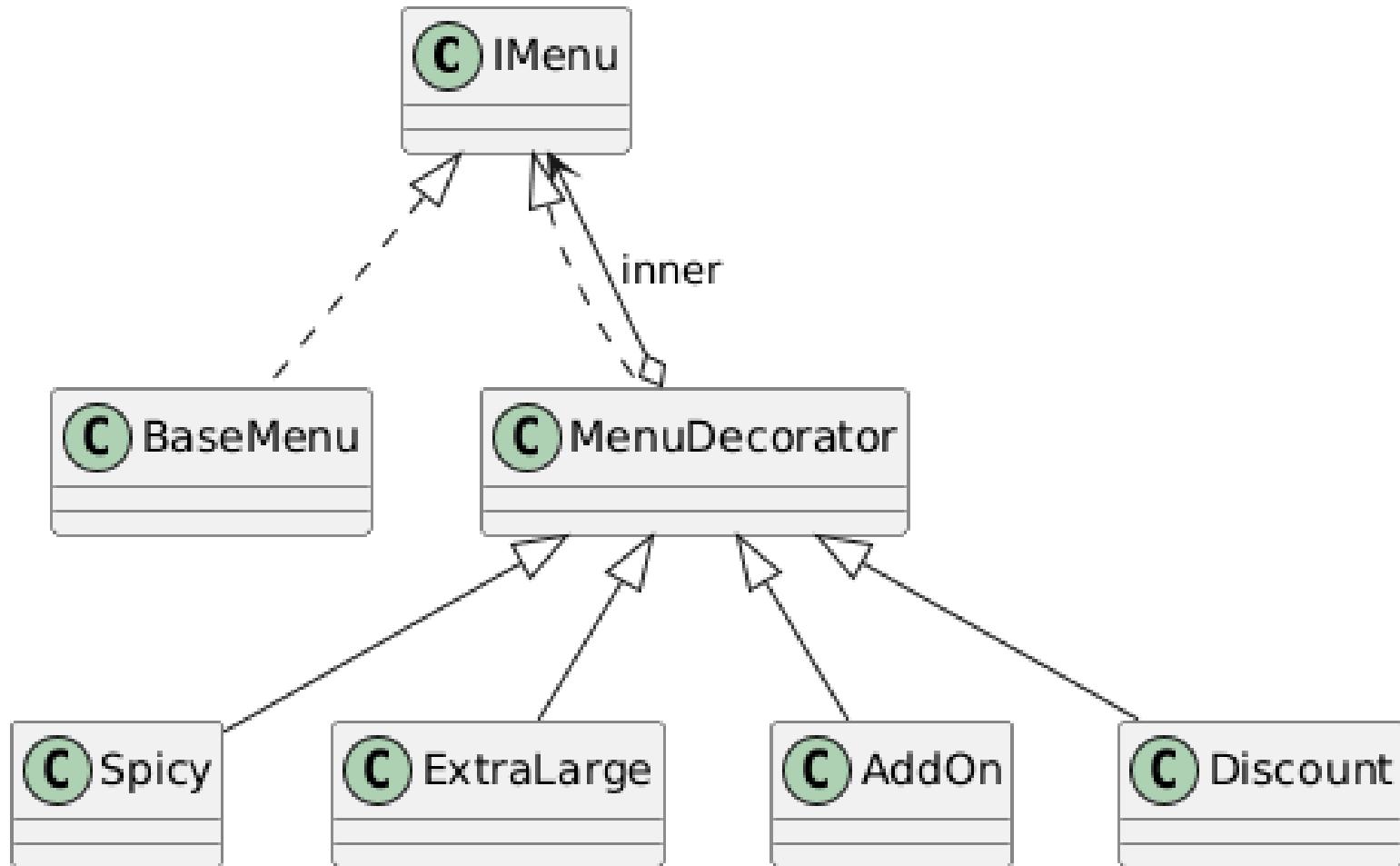
Kyoung Shin Park
Computer Engineering
Dankook University

Lab4

- ❑ Decorator 패턴을 사용하여 학생식당 메뉴 옵션과 할인을 데코레이더로 처리한다.
 - **IMenu** 인터페이스는 **int price()**와 **String description()**를 갖는다.
 - **MenuDecorator** 추상 클래스는 implements **IMenu** 하고 IMenu decoratedMenu 멤버를 갖는다.
 - **BaseMenu** 레코드는 implements **IMenu**를 하고 (String name, int price)로 기본 메뉴의 이름과 가격을 제공한다.
 - **Spicy, ExtraLarge, AddOn, Discount** 클래스는 extends **MenuDecorator** 하고 다음같이 처리한다.
 - ❑ int level(1~3)에 따른 매운맛 정보 추가 및 단계당 500원 가산
 - ❑ 곱빼기 정보 추가 및 1500원 가산
 - ❑ String name, int add에 따른 추가 메뉴 정보 및 가격 가산
 - ❑ int percent(0~100)에 따른 할인 정보 추가 및 가격 비율 할인

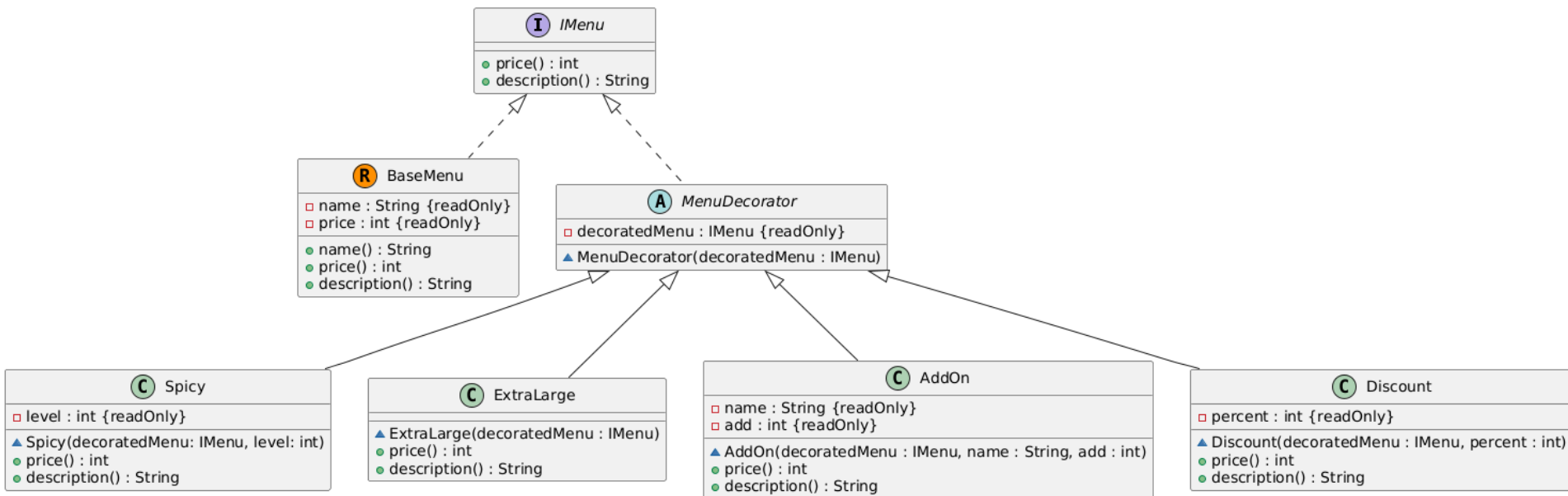
Lab4

▣ Class Diagram



Lab4

Class Diagram



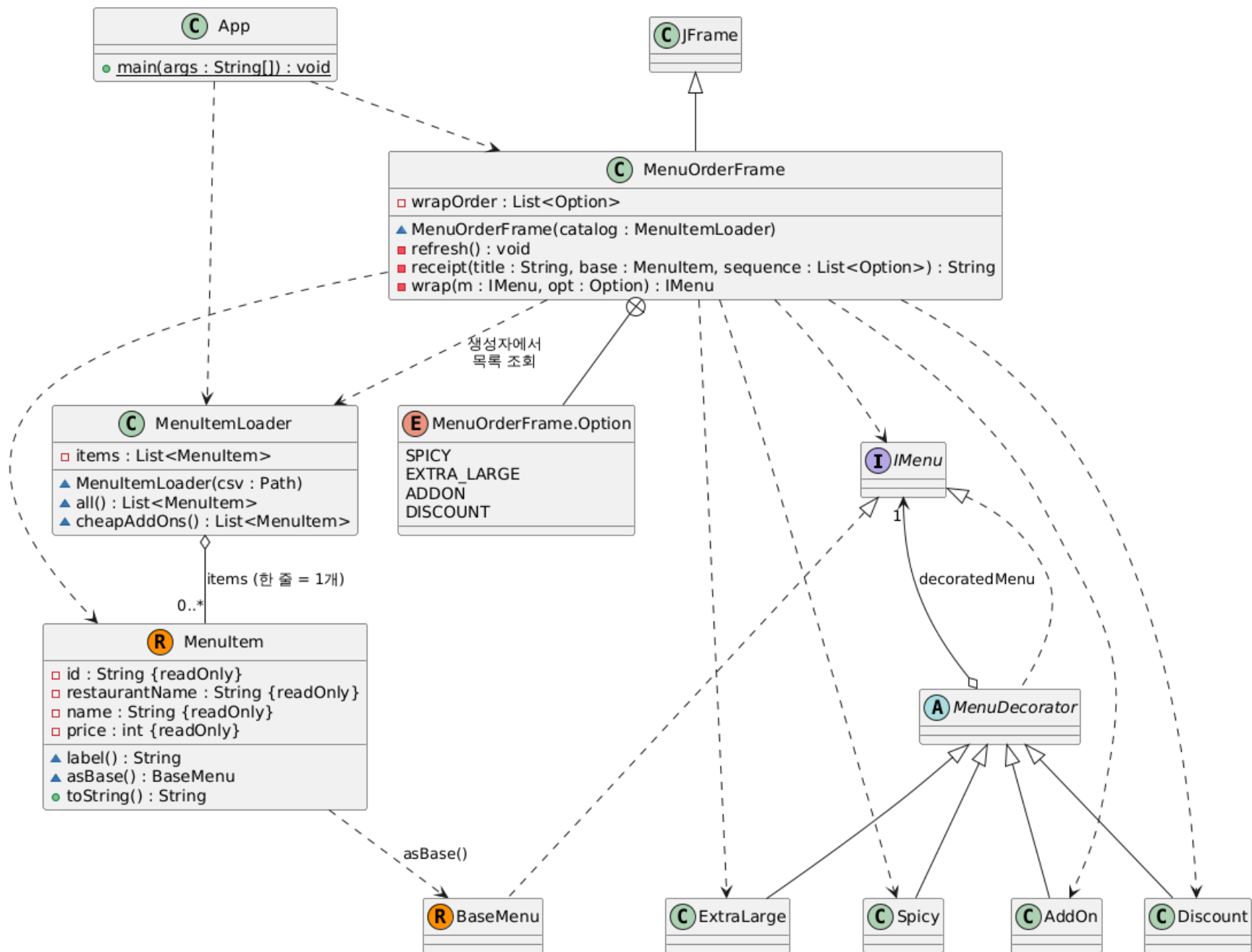
Lab4

□ MenuItem 레코드

- MenuItem 레코드는 (String id, String restaurantName, String name, int price)를 갖는다.
- **label()**은 "코너 / 메뉴 (가격원)" 형식의 표시 문자열을 제공한다.
- **asBase()**는 자신의 name과 price로 **BaseMenu**를 생성하여 데코레이터 체인의 시작점을 만든다.

□ MenuItemLoader 클래스

- 생성자에서 menus.csv를 한 줄씩 읽어, 헤더와 빈 줄을 건너뛰고 id, restaurantName, name, price 순서로 MenuItem 목록을 만든다.
- **all()**은 전체 메뉴를 **불변 리스트**로 제공한다.
- **cheapAddOns()**는 가격이 2500원 이하인 메뉴를 추가 메뉴(AddOn) 후보로 제공한다.



Lab4

□ App 클래스

- App은 main()에서 menus.csv를 읽어 MenuItemLoader를 만들고 MenuOrderFrame을 띄운다.

```
public class App {  
    public static void main(String[] args) throws Exception {  
        // list of MenuItem  
        Path csv = Path.of("menus.csv");  
        MenuItemLoader catalog = new MenuItemLoader(csv);  
        SwingUtilities.invokeLater(() -> new  
MenuOrderFrame(catalog).setVisible(true));  
    }  
}
```

Lab4

▣ MenuOrderFrame 클래스

- 버튼을 누를 때마다 현재 메뉴(IMenu current)를 해당 데코레이터로 한 겹씩 감싸고, 단계별 가격 변화와 최종 합계 및 설명을 출력한다.
- 누른 순서가 곧 감싸는 순서이며, 같은 옵션을 여러 번 적용할 수 있다. 할인을 먼저 적용한 경우와 나중에 적용한 경우의 합계를 비교한다.

```
class MenuOrderFrame extends JFrame {  
    private final JComboBox<MenuItem> baseBox = new JComboBox<>();  
    private final JSpinner spicyLv = new JSpinner(new  
SpinnerNumberModel(1, 1, 3, 1));  
    private final JComboBox<MenuItem> addOnBox = new  
JComboBox<>();  
    private final JSpinner discountPct = new JSpinner(new  
SpinnerNumberModel(10, 5, 30, 5));  
    private final JTextArea out = new JTextArea(14, 40);  
    private IMenu current; // 지금까지 감싼 체인의 가장 바깥 겹  
    private final StringBuilder steps = new StringBuilder(); //  
단계별 영수증 줄 기록 (단계 | 증감 | 누적)
```

Lab4

```
MenuOrderFrame(MenuItemLoader catalog) {  
    super("HW4 Decorator");  
    // 기본메뉴는 전체 메뉴를 baseBox에 채운다  
    catalog.all().forEach(baseBox::addItem);  
    // 추가 메뉴는 2500원 이하 메뉴를 addOnBox에 채운다  
    catalog.cheapAddOns().forEach(addOnBox::addItem);  
    // 버튼  
    JButton resetBtn = new JButton("초기화");  
    JButton spicyBtn = new JButton("매운맛 추가");  
    JButton xlBtn = new JButton("곱빼기 추가");  
    JButton addOnBtn = new JButton("추가 메뉴");  
    JButton discountBtn = new JButton("할인 적용");  
... // 패널 레이아웃  
  
    // 기본메뉴를 바꾸거나 초기화하면 체인을 BaseMenu부터 다시시작  
    baseBox.addActionListener(e -> reset());  
    resetBtn.addActionListener(e -> reset());  
}
```

Lab4

```
MenuOrderFrame(MenuItemLoader catalog) {  
    super("HW4 Decorator");  
    // 기본메뉴는 전체 메뉴를 baseBox에 채운다  
    catalog.all().forEach(baseBox::addItem);  
    // 추가 메뉴는 2500원 이하 메뉴를 addOnBox에 채운다  
    catalog.cheapAddOns().forEach(addOnBox::addItem);
```

```
    JButton resetBtn = new JButton("초기화");  
    JButton spicyBtn = new JButton("매운맛 추가");  
    JButton xlBtn = new JButton("곱빼기 추가");  
    JButton addOnBtn = new JButton("추가 메뉴");  
    JButton discountBtn = new JButton("할인 적용");  
    ... // 패널 레이아웃
```

```
// 기본 메뉴를 바꾸거나 초기화하면 체인을 BaseMenu부터 다시 시작  
baseBox.addActionListener(e -> reset());  
resetBtn.addActionListener(e -> reset());
```

Lab4

```
// 각 버튼은 current를 새 데코레이터로 한 겹 감싼다.  
// 값(단계, 할인율, 추가 메뉴)은 버튼을 누르는 순간의 값이 그  
겹에 고정된다.  
    spicyBtn.addActionListener(e -> onSpicy((Integer)  
spicyLv.getValue()));  
    xlBtn.addActionListener(e -> onExtraLarge());  
    addOnBtn.addActionListener(e -> {  
        MenuItem item = (MenuItem) addOnBox.getSelectedItem();  
        if (item != null) onAddOn(item);  
    });  
    discountBtn.addActionListener(e -> onDiscount((Integer)  
discountPct.getValue()));  
    reset();  
    pack();  
    setLocationRelativeTo(null);  
}  
// onReset, onSpicy, onExtraLarge, onAddOn, onDiscount 구현요망
```

Submit to e-learning

- ▣ Lab4 과제에 yourcode 를 추가 (yourcode 없을시 10점에
서 -1점 감점)
- ▣ **Java26-2-HW4-YourID-YourName.zip** 과제(보고서에
반드시 yourcode 설명 포함)를 e러닝에 제출 (**due by
10/14**).