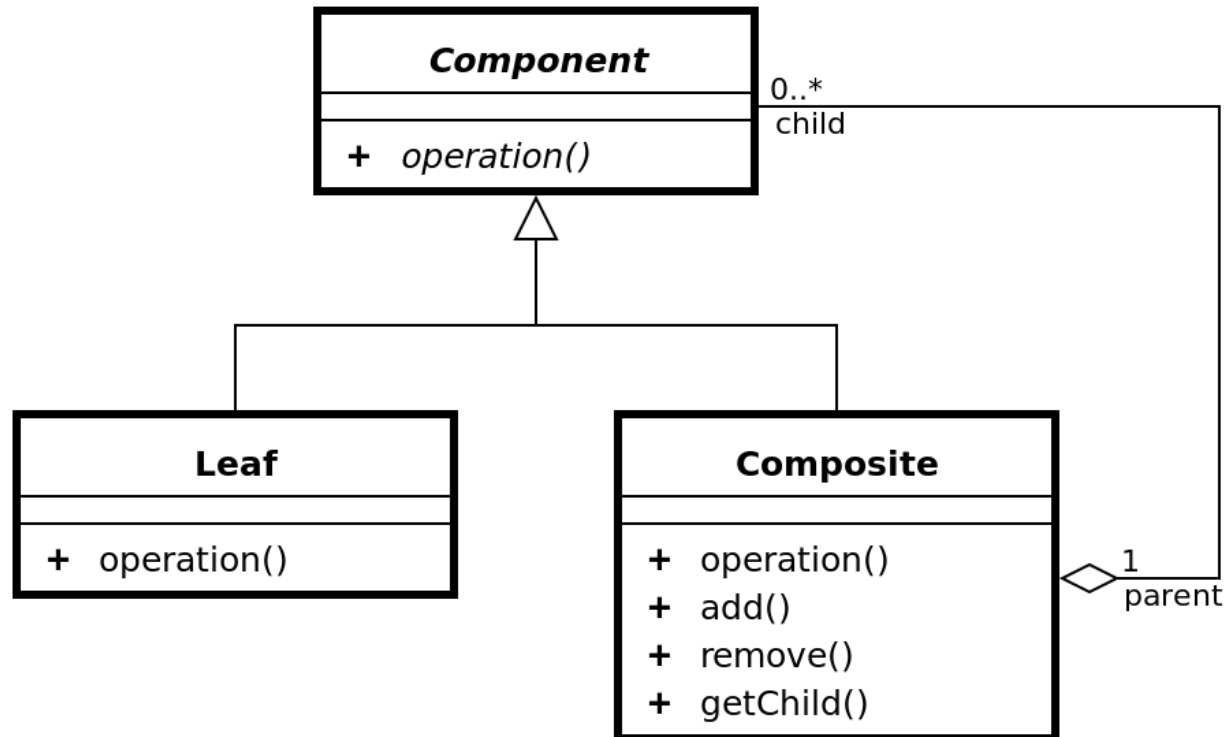


Composite 패턴, 메뉴

Composite 패턴

❖ Composite 패턴

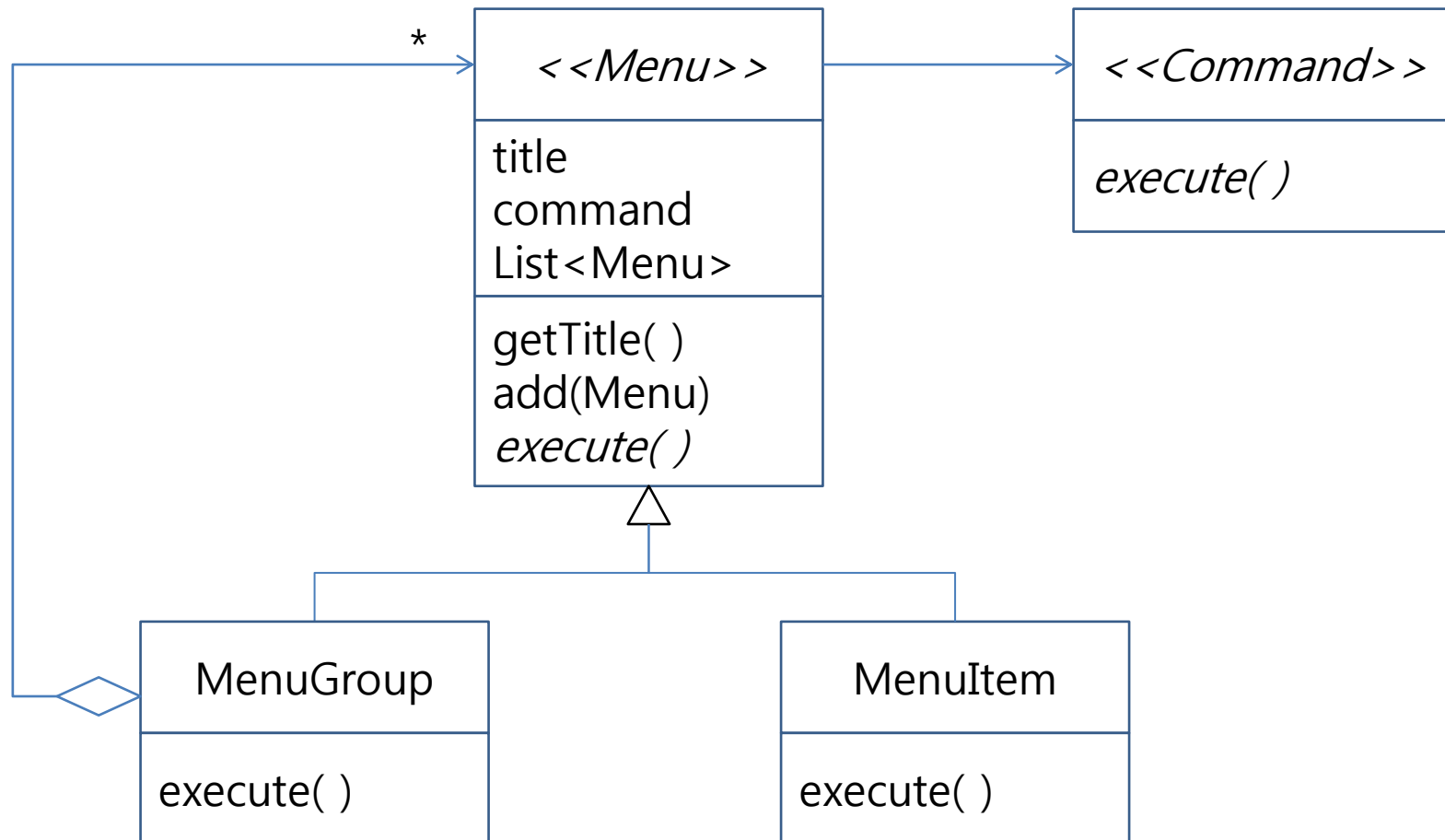
- 객체들의 관계를 트리 구조로 구성 부분-전체 계층을 표현하는 패턴
- 사용자가 단일 객체와 복합 객체 모두 동일하게 처리
- 대표적인 예 : 파일, 디렉토리 → 파일로 처리



메뉴

❖ 메뉴

- 메뉴아이템과 메뉴그룹을 메뉴로 동일하게 처리



❖ Command 인터페이스

```
public interface Command {  
    void execute() throw Exception;  
}
```

메뉴

❖ Menu 추상 클래스

```
@Data
@NoArgsConstructor
public abstract class Menu {
    String title;
    List<Menu> menuList;

    public Menu(String title) {
        this.title = title;
    }

    public void add(Menu menu) {
        menuList.add(menu);
    }

    public abstract void execute() throws Exception;
}
```

❖ MenuItem : 단일 메뉴 항목

```
public class MenuItem extends Menu {  
  
    Command command;    // 관련 명령  
  
    public MenuItem(String title, Command command) {  
        super(title);  
        this.command = command;  
    }  
  
    @Override  
    public void execute() throws Exception {  
        if(command != null) {  
            command.execute();  
        }  
    }  
}
```

❖ MenuGroup 메뉴 그룹

```
public class MenuGroup extends Menu {  
  
    public MenuGroup() {  
        this(null);  
    }  
  
    public MenuGroup(String title) {  
        super(title);  
        menuList = new ArrayList<Menu>();  
    }  
}
```

❖ MenuGroup 메뉴 그룹

```
@Override
public void execute() throws Exception {
    if(title!=null) {
        System.out.printf("[%s] ", title);
    }
    for(int i=0; i<menuList.size(); i++) {
        Menu menu = menuList.get(i);
        if(menu.getMenuList() != null) {    // 서브 메뉴
            System.out.printf("[%d) %s] ", i, menu.getTitle());
        } else {    // 다일 메뉴
            System.out.printf("%d) %s ", i, menu.getTitle());
        }
    }
    System.out.println();
    Prompt prompt = new Prompt();
    int select = prompt.getInt("선택");
    if(select >=0 && select<menuList.size()) {
        menuList.get(select).execute();
    }
}
}
```


❖ Application

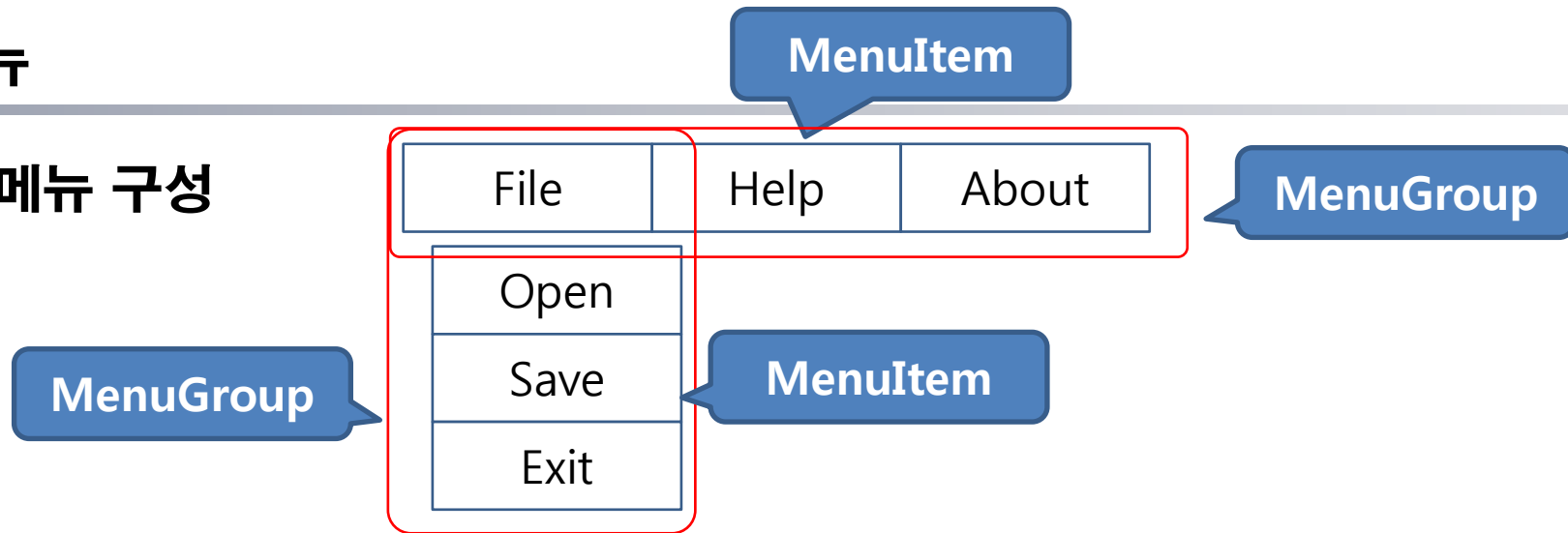
```
public abstract class Application {  
    Menu menu;  
  
    public Application() {  
        menu = new MenuGroup();  
        createMenu(menu);  
    }  
  
    abstract public void createMenu(Menu menu);  
  
    public void init() {    }
```

❖ Application

```
public void run() {  
    init();  
    while(true) {  
        try {  
            menu.execute();  
        } catch (Exception e) {  
            System.out.println("명령 실행 에러 : " +  
                                e.getMessage());  
        }  
    }  
}
```

메뉴

❖ 메뉴 구성



```
public void createMenu(Menu menu) {  
  
    MenuGroup fileMenu = new MenuGroup("File");  
    fileMenu.add(new MenuItem("Open", null);  
    fileMenu.add(new MenuItem("Save", null));  
    fileMenu.add(new MenuItem("Exit", ...));  
  
    menu.add(fileMenu);  
    menu.add(new MenuItem("Help", null));  
    menu.add(new MenuItem("About", null));  
}
```