

60200307 우연정

목차

1. Data Structure

1-1. Aggregation Hierarchy

1-2. Inheritance Hierarchy

1. Data Structure

1-1. Aggregation Hierarchy

집합 계층 구조(트리 구조)는 문법적으로 노드와 링크로 구성된다. 노드의 링크의 방향성이 존재하면 한쪽을 부모, 한쪽을 자식이라고 정의한다. 여기에는 제약 조건이 있다. 제약조건은 하나의 자식은 하나의 부모만을 가질 수 있다는 것이다. 거꾸로 말하자면, 부모는 여러 개의 자식을 가질 수 있어서 1:n의 관계인 것이다. 그렇기 때문에 트리구조가 만들어진다.

언어가 되려면, 문법과 의미가 있어야 한다. 단어의 구성요소들이 의미를 가진다. aggregation은 트리가 가지는 의미를 말한다. aggregation은 life cycle을 관리한다. 프로그램을 종료하면 자식이 먼저 죽고 그다음 부모가 죽는다.

1-2. Inheritance Hierarchy

Inheritance Hierarchy 상속 구조는 문법적으로 트리 구조를 가지고 있다. generalization과 specialization이 있다. specialization(특수화)는 특성이 추가되는 것이다. 자바에서는 extends(확장)라는 명령어를 쓴다.

확장 말고도, 특정 값을 세팅하는 것만으로도 specialization(특수화)되는 것이다.

메뉴에는 메뉴바가 있고, 글자가 버튼이고, 메뉴가 있고, 그 메뉴를 열면 메뉴 아이템이 있다. 메뉴바와 패널을 만들 때는 영역의 위치와 크기를 조정해야 한다. 이를 '레이아웃'이라고 한다. 레이아웃을 정하는 방법은 두 가지가 있다. 먼저, 내가 절대 좌표를 써서 조정하는 방법이다. 하지만 이 방법은 윈도우 크기를 조절할 때마다 안에 있는 구성 요소들의 위치가 그대로인 문제가 발생한다. 동적으로 사이즈를 조절하는 방법을 쓰면 이를 해결할 수 있다. 이를 레이아웃 매니저라고 한다. 레이아웃 매니저는 자식의 크기와 위치를 관리해주는 관리자이다.

외부에 노출되는 것은 내가 명시적으로 정리해서 지속적으로 그 이름을 여러 군데에서 써야 할 부분만 쓰는 것이다. 그런데 레이아웃은 중간에 바뀌는 경우가 거의 없기 때문에 한번 만들고 나서는 끝이라고 할 수 있다.

레이아웃 매니저에는 여러 종류가 있는데, 화면을 동, 서, 남, 북으로 나누어 배치해주는 Border Layout, 요소들을 일렬로 이어지게 배치해주는 Flow Layout이 있다.