

제9장 설계

소프트웨어 설계는 '고객이 무엇을 원하고 있는가?'

즉, 무엇을 개발해야 하는지 요구분석이 끝나면 다음 단계로 소프트웨어를 설계한다.

설계는 요구분석명세서에 기술된 기능이 수행되고, 공학적으로 의미 있도록 표현되어야 한다.

〈무엇을 설계하는가?〉

- 자료구조 설계 : 자료를 프로그래밍 구현에 적합한 자료구조로 만든다.
 - 컴포넌트 설계 : 컴포넌트는 재사용 가능하도록 설계한다.(컴포넌트 조립으로 시스템 개발)
 - 인터페이스 설계 : 사용자가 시스템을 쉽게 이용할 수 있도록 설계한다.
 - 시스템구조 설계 : 시스템의 각 요소들(서브시스템) 사이의 구조 관계를 설계한다.
-

1. 컴포넌트(component)

다음은 비주얼베이직 프로그램에서 사용되는 컨트롤이다. 컨트롤은 일반적인 컴포넌트이다.



- 비주얼베이직의 도구상자는 컨트롤이 모여 있는 상자이다.
- 각 컨트롤은 자신만의 독특한 기능을 가진다.
- 비주얼베이직 프로그래머는 각 컨트롤을 상황에 맞게 사용할 수 있어야 한다.
- 컨트롤은 비주얼베이직 프로그램 작성에 없어서는 안 될 핵심 요소이다.
- 컨트롤을 사용하려면 그 기능과 사용법을 알아야 한다.(사용자 인터페이스)
- 각 컨트롤을 이용하여 여러 개의 객체(인스턴스)를 생성할 수 있다.
- 객체는 속성, 메서드를 가지고 이벤트에 반응한다.

// 컨트롤(컴포넌트)

- **TextBox**는 텍스트, 수 등을 나타내거나 실행 중에 편집 가능하다.(입출력 가능)
- **CheckBox**는 같은 그룹에서 동시에 여러 개 선택 가능하다.
- **Shape**는 사각형, 원 등 여러 가지 도형을 제작할 수 있다.

// 한국 소프트웨어 컴포넌트 컨소시엄의 '알기쉬운 소프트웨어 컴포넌트'에서 발췌

① 컴포넌트는 독립적인 소프트웨어 모듈이다.

- 컴포넌트는 소프트웨어 시스템에서 독립적인 업무 또는 기능을 수행하는 모듈이다.
- 컴포넌트는 시스템을 유지보수할 때 **교체가 가능한 부품**이다.
- 소프트웨어 컴포넌트는 하드웨어의 그래픽카드와 같은 개념으로 독립적인 기능을 수행한다.

② 컴포넌트는 구현, 명세화, 패키징화, 그리고 배포될 수 있어야 한다.(표준화도 필요)

- 컴포넌트는 **실행코드** 기반으로 **재사용**할 수 있도록 구현(implementation)되어야 한다.
- 컴포넌트는 **원시코드(source code)**가 아닌 **실행코드(executable code)**이어야 한다.
- 컴포넌트 정보는 **명세화(specification)** 되어 있어야 한다.
- 컴포넌트의 용도, 유형, 기술표준, 인터페이스 등에 대한 정보들을 명세화한다.
- 컴포넌트는 **교체 가능**해야 한다.
- 교체 가능한 컴포넌트를 개발하기 위해서는 **표준(standard)**을 준수해야 한다.
- 컴포넌트는 **독립적으로 배포(deployment)** 가능해야 한다.
- 컴포넌트는 독립적인 업무 단위로 개발된 것이다.
- 컴포넌트가 개발 완료되어 배포될 때는 패키징화(packaging)되어 있어야 한다.
- 즉, 컴포넌트가 배포될 때는 관련 문서와 코드들이 독립적인 단위로 패키징화되어야 한다.
- 컴포넌트는 사용자가 필요한 기능만을 패키징한 컴포넌트를 재사용할 수 있어야 한다.

③ 하나의 컴포넌트는 하나 이상의 클래스들로 구성될 수 있다.

- 컴포넌트와 클래스는 **다른 개념**이며 컴포넌트와 객체 또한 다른 개념이다.
- 컴포넌트가 되기 위해서 반드시 클래스나 객체로 만들어야 하는 것은 아니다.
- 기존의 C나 COBOL로 개발된 모듈 또한 컴포넌트로 만들 수 있다.
- 하지만, 현재 컴포넌트 기술 표준과 핵심 개념은 객체지향에서 비롯된 것이 많다.
→ 해서, 컴포넌트 개발은 객체지향 기술을 바탕으로 하는 것이 보다 효과적일 수 있다.
- 하나의 컴포넌트는 **하나 이상의 클래스들**을 바탕으로 개발할 수 있다.
- 컴포넌트가 실행될 때 해당 컴포넌트의 인스턴스(instance)인 컴포넌트 객체가 수행된다.
→ 이때 컴포넌트 내부에서는
하나 이상의 클래스에 대응하는 하나 이상의 객체들이 수행되고 있는 것이다

④ 컴포넌트는 인터페이스를 통해서만 접근할 수 있다.

- 컴포넌트는 외부에서 컴포넌트에 접근할 수 있도록 인터페이스를 제공해야 한다.
 - 컴포넌트의 모든 내부 정보는 외부로부터 숨긴다.(인터페이스 통한 접근만 허용)
 - 인터페이스는 **컴포넌트가 제공하는 서비스를 정의한 것**이다.
 - 인터페이스는 컴포넌트가 외부와의 연결을 위해 반드시 존재해야만 한다.
 - 컴포넌트는 **조립 가능**해야 한다.(컴포넌트 기반 소프트웨어 개발)
 - 컴포넌트가 인터페이스만 제공함으로써 컴포넌트도 하드웨어처럼 조립 가능하도록
-



탐구

사용자 인터페이스(UI, user interface)란?

- 먼저, 영어 단어 **interface**의 사전적 의미는 **접속기**이다.
- 인터페이스(interface)는 서로 다른 사물들 사이에서 **상호작용(interaction)**을 매개하는 것이다.
- **UI**는 사람(사용자)과 앱, 컴퓨터 등 사이에서 일어나는 상호작용(interaction)을 매개하는 것이다.
- **UI**는 사람(사용자) 또는 사물들(컴퓨터, 기계, 시스템 등) 사이에서 **의사소통**할 수 있도록 일시적 또는 영구적인 접근을 목적으로 만들어진 물리적, 가상적 매개체이다.
- **UI**는 키보드, 마우스, 화면, 아이콘, 도움말 등 사용자들과 상호작용을 하도록 설계된 모든 고안품을 포함하며, 앱이나 컴퓨터 등이 상호작용을 초래하거나 그것에 반응하는 방법 등을 의미한다.

〈사용자 인터페이스 설계 지침〉

- ① **가시성의 원칙(visibility)**
 - 앱의 주요기능(feature)을 노출시켜 최대한 앱 조작이 쉽도록 구성
- ② **조작결과 예측의 원칙(natural mapping)**
 - 사용자가 앱을 조작하여 작동시킨 결과를 조작 부위만 보고도 미리 예측가능하게 설계
 - UI 구성 및 디자인을 통해 기능 및 실행결과가 예측 가능하도록 설계
- ③ **일관성의 원칙(consistency)**
 - 앱의 조작방식에 일관성을 제공함으로 사용자가 쉽게 기억하고 빠르게 적응할 수 있도록 설계
- ④ **단순성의 원칙(simplicity)**
 - 앱의 기능구조를 단순화시켜 조작에 요구되는 노력을 최소화하여 인지적 부담을 줄이게 한다.
- ⑤ **지식배분의 원칙(knowledge in world & head)**
 - 앱의 기능 조작에 요구되는 지식은 사용자의 지식과 기억구조에 적합하도록 설계
 - 해서, 사용자가 학습하기 쉽고 기억하기 쉽게 상호 보완적으로 분배
- ⑥ **조작오류의 원칙(design for error)**
 - 발생된 오류는 쉽게 발견될 수 있도록 설계한다.
 - 또한, 오류 수정이 최대한 쉽게 이루어질 수 있도록 설계
- ⑦ **제한사항 선택사용의 원칙(constraint)**
 - 앱의 조작 상의 제한사항을 이용하여 가능한 선택의 여지를 줄여 조작방법이 명확하도록 설계
- ⑧ **표준화의 원칙(standardization)**
 - 앱의 기능구조와 디자인을 표준화하여 한번 학습한 이후 효과적으로 사용할 수 있도록 설계
- ⑨ **행동유도성의 원칙(affordance)**
 - 사용자에게 앱의 기능을 어떻게 조작하면 될 것인가에 관한 단서를 제공
- ⑩ **접근성의 원칙(accessibility)**
 - 사용자의 성별, 연령, 인종 등에 따른 다양한 계층의 사용자를 수용할 수 있도록 고려

〈한국HCI연구회〉

한국 HCI 학회(The HCI Society of Korea)는 **인간과 컴퓨터의 상호작용**(HCI, human computer interaction)에 관한 이론과 응용에 관련된 학문을 연구하는 모임이다.

4 <http://cafe.daum.net/pass365>(홍재연)

// 인터페이스 종류

CLI	<ul style="list-style-type: none">• command-line interface• 명령어 줄 인터페이스• 사용자가 컴퓨터 자판 등을 이용해 명령어를 문자열 형태로 입력한다.
GUI	<ul style="list-style-type: none">• graphical user interface• 그래픽 사용자 인터페이스• 입출력 등의 기능을 알기 쉬운 아이콘 같은 그래픽으로 나타낸 것이다.• 사용자는 각 기능을 편리하게 사용할 수 있다.

기출문제 분석

1. 컴포넌트에 대한 설명으로 옳지 않은 것은? [2014년 국가 7급]

- ① 컴포넌트는 인스턴스를 정의하는 데 사용되는 템플릿으로서 타입을 정의한다.
- ② 컴포넌트는 배치 가능한 개체로서 실행 플랫폼에 직접 설치된다.
- ③ 컴포넌트를 정의하는 요소는 인터페이스이다.
- ④ 컴포넌트는 언어 독립적이고, 객체지향 프로그래밍 언어가 아닌 언어로도 구현할 수 있다.

☞ 컴포넌트(component)

-
- 컴포넌트는 인스턴스를 정의하는 데 사용되는 템플릿으로서 타입을 정의한다.(x)
→ 컴포넌트는 인터페이스를 정의해야 한다.(외부와의 연결을 위해서)
 - 인터페이스는 컴포넌트가 제공하는 서비스를 정의한 것이다.
 - 컴포넌트는 인터페이스를 통해서만 접근할 수 있다.
 - 인스턴스(객체) 타입 정의는 Class이다.
-

정답 : ①

2. 컴포넌트 기반 개발에 대한 설명으로 옳지 않은 것은? [2023년 국가 7급]

- ① 컴포넌트는 정적으로 바인딩되며 실행시간에는 인터페이스를 통해 접근할 수 없다.
- ② 재사용 가능한 컴포넌트를 기반으로 소프트웨어를 개발하는 방법론이다.
- ③ 컴포넌트는 독립적이고 교체 가능한 부품이다.
- ④ 컴포넌트는 인터페이스를 통해 다른 컴포넌트와 연결된다.

☞ 컴포넌트 기반 개발

-
- 컴포넌트는 정적으로 바인딩되며 실행시간에는 인터페이스를 통해 접근할 수 없다.(x)
→ 컴포넌트는 실행코드이다. 동적으로 바인딩된다.
-

정답 : ①

3. 사용자 인터페이스를 설계할 때 고려 사항이 아닌 것은? [2020년 국가 7급]

- ① 소프트웨어 내부 모듈 사이의 인터페이스를 정의하거나 외부시스템 사이의 인터페이스를 정의한다.
- ② 사용자가 어떤 액션을 취하였는지 아니면 오류가 발생했는지 알지 못할 때는 시스템이 적절히 지원하는 것은 중요하다.
- ③ 사용자 인터페이스는 일관성이 있어야 한다.
- ④ 사용자 인터페이스는 사용자로부터의 입력이 최소화되도록 해야 한다.

☞ 사용자 인터페이스

-
- 소프트웨어 내부 모듈 사이의 인터페이스를 정의하거나 외부시스템 사이의 인터페이스를 정의한다.(x)
 - 내부 모듈 사이의 인터페이스는 정의하지 않는다.
 - 인터페이스는 서로 다른 사물들 사이에서 상호작용(interaction)을 매개하는 것이다.

—————(사용자 인터페이스 설계 지침)—————

- ① 가시성의 원칙(visibility)
 - 앱의 주요기능(feature)을 노출시켜 최대한 앱 조작이 쉽도록 구성
- ② 조작결과 예측의 원칙(natural mapping)
 - 사용자가 앱을 조작하여 작동시킨 결과를 조작 부위만 보고도 미리 예측가능하게 설계
 - UI 구성 및 디자인을 통해 기능 및 실행결과가 예측 가능하도록 설계
- ③ 일관성의 원칙(consistency)
 - 앱의 조작방식에 일관성을 제공함으로 사용자가 쉽게 기억하고 빠르게 적응할 수 있도록 설계
- ④ 단순성의 원칙(simplicity)
 - 앱의 기능구조를 단순화시켜 조작에 요구되는 노력을 최소화하여 인지적 부담을 줄이게 한다.
- ⑤ 지식배분의 원칙(knowledge in world & head)
 - 앱의 기능 조작에 요구되는 지식은 사용자의 지식과 기억구조에 적합하도록 설계
 - 해서, 사용자가 학습하기 쉽고 기억하기 쉽게 상호 보완적으로 분배
- ⑥ 조작오류의 원칙(design for error)
 - 발생된 오류는 쉽게 발견될 수 있도록 설계한다.
 - 또한, 오류 수정이 최대한 쉽게 이루어질 수 있도록 설계
- ⑦ 제한사항 선택사용의 원칙(constraint)
 - 앱의 조작 상의 제한사항을 이용하여 가능한 선택의 여지를 줄여 조작방법이 명확하도록 설계
- ⑧ 표준화의 원칙(standardization)
 - 앱의 기능구조와 디자인을 표준화하여 한번 학습한 이후 효과적으로 사용할 수 있도록 설계
- ⑨ 행동유도성의 원칙(affordance)
 - 사용자에게 앱의 기능을 어떻게 조작하면 될 것인가에 관한 단서를 제공
- ⑩ 접근성의 원칙(accessibility)
 - 사용자의 성별, 연령, 인종 등에 따른 다양한 계층의 사용자를 수용할 수 있도록 고려

—————(한국HCI연구회)—————

정답 : ①