20220424

<1과목: 소프트웨어 설계>

1. UML 다이어그램 중 순차 다이어그램에 대한 설명

객체 간의 동적 상호작용을 시간 개념을 중심으로 모델링

일반적으로 다이어그램의 수직방향이 시간의 흐름을 나타냄

회귀메시지, 제어블록 등으로 구성

틀린 것: 주로 시스템의 정적 측면을 모델링하기 위해 사용

1. 메시지 지향 미들웨어(message-oriented middleware, MOM)에 대한 설명

독립적인 애플리케이션을 하나의 통합된 시스템으로 묶기 위한 역할

송신측과 수신측의 연결 시 메시지 큐를 활용하는 방법이 있다

상이한 애플리케이션 간 통신을 비동기 방식으로 지원

틀린 것: 느리고 안정적인 응답보다는 즉각적인 응답이 필요한 온라인 업무에 적합

1. 익스트림 프로그래밍에 대한 설명

소규모 개발조직이 불확실하고 변경이 많은 요구를 접하였을 때 적절

익스트림 프로그래밍을 구동시키는 원리는 상식적인 원리와 경험을 최대로 끌어올리는 것

구체적인 실천방법을 정의, 개발 문서보다 소스코드에 중점

틀린 것: 대표적인 구조적 방법론 중 하나

1. 유스케이스의 구성요소 간의 관계

연관

확장

일반화

틀린 것: 구체화

1. 요구사항 분석에서 비기능적 요구에 대한 설명

‘차량 대여 시스템이 제공하는 모든 화면이 3초 이내에 사용자에게 보여야 한다’는 비기능적

틀린 것:

시스템의 처리량(Throughput), 반응시간 등의 성능 요구나 품질 요구는 비기능적 x

시스템 구축과 관련된 안전, 보안에 대한 요구사항들은 비기능적 x

‘금융 시스템은 조회, 인출, 입금, 송금의 기능이 있어야 한다’는 비기능적 요구

1. 정보공학 방법론에서 데이터베이스 설계의 표현으로 사용하는 모델링 언어

Entity-Relationship Diagram

틀린 것:

Package Diagram

State Transition Diagram

Deployment Diagram

1. 미들웨어에 대한 설명

여러 운영체제에서 응용 프로그램들 사이에 위치한 소프트웨어

소프트웨어 컴포넌트를 연결하기 위한 준비된 인프라 구조를 제공

여러 컴포넌트를 1대 1, 1대 다, 다대 다 등 여러가지 형태로 연결이 가능

틀린 것: 미들웨어의 서비스 이용을 위해 사용자가 정보교환방법 등의 내부 동작을 쉽게 확인할 수 있어야 한다

1. UI의 설계지침

이해하기 편하고 쉽게 사용할 수 있는 환경을 제공해야 한다

주요 기능을 메인화면에 노출하여 조작이 쉽도록 하여야 한다

사용자의 직무, 연령, 성별 등 다양한 계층을 수용하여야 한다

틀린 것: 치명적인 오류에 대한 부정적인 사항은 사용자가 인지할 수 없도록 한다

1. 객체지향 개념에서 다형성(Polymorphism)과 관련된 설명

다형성은 현재코드를 변경하지 않고 새로운 클래스를 쉽게 추가할 수 있다

다형성이란 여러가지 형태를 가지고 있다는 의미로, 여러 형태를 받아들일 수 있는 특징

메소드 오버라이딩은 상위클래스에서 정의한 일반 메소드의 구현을 하위 클래스에서 무시하고 재정의 할 수 있다

틀린 것: 메소드 오버로딩의 경우 매개변수타입은 동일, 메소드명을 다르게 함으로써 구현, 구분할 수 있다

1. 소프트웨어 개발영역을 결정하는 요소

인터페이스:

소프트웨어에 의해 간접적으로 제어되는 장치와 소프트웨어를 실행하는 하드웨어

기존의 소프트웨어와 새로운 소프트웨어를 연결하는 소프트웨어

순서적 연산에 의해 소프트웨어를 실행하는 절차

1. 객체에 대한 설명

객체는 상태, 동작, 고유 식별자를 가진 모든것이라 할 수 있다

객체는 필요한 자료 구조와 이에 수행되는 함수들을 가진 하나의 독립된 존재

객체의 상태는 속성값에 의해 정의

틀린 것: 객체는 공통속성을 공유하는 클래스들의 집합

1. 속성과 관련된 연산(Operation)을 클래스 안에 묶어서 하나로 취급하는 것을 의미하는 객체지향 개념

Encapsulation

틀린 것:

Inheritance: 상속

Class: 클래스

Association: 연관

1. 애자일(Agile) 프로세스 모델에 대한 설명

프로세스와 도구 중심이 아닌 개개인과의 상호소통을 통해 의견을 수렴

협상과 계약보다는 고객과의 협력을 중시

문서 중심이 아닌 실행가능한 소프트웨어를 중시

틀린 것: 변화에 대한 대응보다는 자세한 계획을 중심으로 소프트웨어를 개발

1. 명백한 역할을 가지고 독립적으로 존재할 수 있는 시스템의 부분, 넓은 의미에서는 재사용되는 모든 단위, 인터페이스를 통해서만 접근 가능한 것

Component

틀린 것:

Model

Sheet

Cell

1. GoF(Gang of Four) 디자인 패턴을 생성, 구조, 행동 패턴의 세 그룹으로 분류할 때 구조 패턴

Adapter

Bridge

Proxy

틀린 것: Builder

1. UI와 관련된 기본 개념 중 하나로, 시스템의 상태와 사용자의 지시에 대한 효과를 보여주어 사용자가 명령에 대한 진행 상황과 표시된 내용을 해석할 수 있도록 도와주는 것

Feedback

틀린 것:

Posture

Hash

Module

1. UI의 종류로 멀티터치, 동작 인식 등 사용자의 자연스러운 움직임을 인식하여 서로 주고받는 정보를 제공하는 것

NUI(Natural User Interface)

틀린 것:

GUI(Graphical User Interface)

OUI(Organic User Interface)

CLI(Command Line Interface)

1. 소프트웨어 모델링과 관련된 설명

구조적 방법론에서는 DFD(Data Flow Diagram), DD(Data Dictionary) 등을 사용하여 요구사항의 결과를 표현

객체지향 방법론에서는 UML표기법을 사용

소프트웨어 모델을 사용할 경우 개발될 소프트웨어에 대한 이해도 및 이해 당사자 간의 의사소통향상에 도움

틀린 것: 모델링 작업의 결과물은 다른 모델링 작업에 영향을 줄 수 없다

1. 유스케이스 다이어그램에 관련된 내용

유스케이스는 사용자 측면에서의 요구사항으로, 사용자가 원하는 목표를 달성하기 위해 수행할 내용을 기술한다

시스템 액터는 다른 프로젝트에서 이미 개발, 사용되고 있으며, 본 시스템과 데이터를 주고받는 서로 연동될 시스템

액터가 인식할 수 없는 시스템 내부의 기능을 하나의 유스케이스로 파악해서는 안된다

틀린 것: 시스템과 상호작용하는 외부시스템은 액터로 파악해서는 안된다

1. 소프트웨어 아키텍쳐 모델 중 MVC(Model-View-Controller)와 관련된 설명

UI를 담당하는 계층의 응집도를 높일 수 있고, 여러 개의 다른 UI를 만들어 그 사이의 결합도를 낮출 수 있다

View는 Model에 있는 데이터를 사용자 인터페이스에 보이는 역할을 담당

Controller는 Model에 명령을 보냄으로써 Model의 상태를 변경할 수 있다

틀린 것: Model은 View와 Controller 사이에서 전달자 역할, View마다 Model 서브시스템이 각각 하나씩 연결

<2과목: 소프트웨어 개발>

통합 테스트(Integration Test)와 관련한 설명

시스템을 구성하는 모듈의 인터페이스와 결합을 테스트

하향식 통합테스트의 경우 넓이우선(Breadth First) 방식으로 테스트를 할 모듈을 선택할 수 있다

모듈 간의 인터페이스와 시스템의 동작이 정상적으로 잘되고 있는지를 빨리 파악하고자 할 때 상향식보다는 하향식 통합 테스트를 사용하는 것이 좋다

틀린 것: 상향식 통합 테스트의 경우 시스템 구조도의 최상위에 있는 모듈을 먼저 구현하고 테스트한다

이진검색방법

중간값 설정 – 찾고자 하는 값과 크기 비교 – 중간값이 새로운 최대값 or 최소값이 됨 – 반복

소프트웨어 공학에서 워크스루(Walkthrough)에 대한 설명

사용사례를 확장하거나 명세하거나 설계 다이어그램, 원시코드, 테스트 케이스 등에 적용할 수 있다

복잡한 알고리즘 또는 반복, 실시간 동작, 병행 처리와 같은 기능이나 동작을 이해하려고 할 때 유용

단순한 테스트 케이스를 이용하여 프로덕트를 수작업으로 수행해 보는 것이다

틀린 것: 인스펙션(Inspection)과 동일한 의미를 가진다

소프트웨어 개발과정에서 소프트웨어의 변경사항을 관리하기 위해 개발된 일련의 활동을 뜻하는 것

형상관리

틀린 것:

복호화

저작권

크랙

테스트 케이스와 관련한 설명

프로그램에 결함이 있더라도 입력에 대해 정상적인 결과를 낼 수 있기 때문에 결함을 검사할 수 있는 테스트 케이스를 찾는 것이 중요

개발된 서비스가 정의된 요구사항을 준수하는지 확인하기 위한 입력값과 실행조건, 예상결과의 집합으로 볼수있다

테스트 케이스 실행이 통과되었는지 실패하였는지 판단하기 위한 기준을 테스트 오라클(Test Oracle)이라고 한다

틀린 것: 테스트의 목표 밑 테스트 방법을 결정하기 전에 테스트 케이스를 작성해야 한다

객체지향 개념을 활용한 소프트웨어 구현과 관련한 설명

객체(Object)란 필요한 자료구조와 수행되는 함수들을 가진 하나의 독립된 존재

상속(Inheritance)은 개별 클래스를 상속관계로 묶음으로써 클래스 간의 체계화된 전체구조를 파악하기 쉽다는 장점

같은 클래스에 속하는 개개의 객체이자 하나의 클래스에서 생성된 객체를 인스턴스(Instance)라고 한다

틀린 것: JAVA에서 정보은닉(Information Hiding)을 표기할 때 private의 의미는 ‘공개’이다

DRM(Digital Rights Management)과 관련한 설명

디지털 콘텐츠와 디바이스의 사용을 제한하기 위해 하드웨어 제조업자, 저작권자, 출판업자 등이 사용할 수 있는 접근 제어 기술을 의미

디지털 미디어의 생명 주기 동안 발생하는 사용권한관리, 과금, 유통 단계를 관리하는 기술로도 볼 수 있다

클리어링 하우스(Clearing House)는 사용자에게 콘텐츠 라이선스를 발급하고 권한을 부여해주는 시스템을 말한다

틀린 것: 원본을 안전하게 유통하기 위한 전자적 보안은 고려하지 않기 때문에 불법 유통과 복제의 방지가 불가능

위험 모니터링의 의미

위험 요소 징후들에 대하여 계속적으로 인지하는 것

틀린 것:

위험을 이해하는 것

첫번째 조치로 위험을 피할 수 있도록 하는 것

위험 발생 후 즉시 조치하는 것

동시에 소스를 수정하는 것을 방지, 다른방향으로 진행된 개발결과를 합치거나 변경내용을 추적할 수 있는 소프트웨어 버전관리도구

RCS(Revision Control System)

틀린 것:

RTS(Reliable Transfer Service)

RPC(Remote Procedure Call)

RVS(Relative Version System)

화이트박스 테스트와 관련한 설명

화이트박스 테스트의 이해를 위해 논리흐름도(Logic-Flow Diagram)을 이용할 수 있다

테스트 데이터를 이용해 실제 프로그램을 실행함으로써 오류를 찾는 동적테스트(Dynamic Test)에 해당한다

테스트 데이터를 선택하기 위하여 검증기준(Test Coverage)를 정한다

틀린 것: 프로그램의 구조를 고려하지 않기 때문에 테스트케이스는 프로그램 또는 모듈의 요구, 명세를 기초로 결정

알고리즘에 관련한 설명

주어진 작업을 수행하는 컴퓨터 명령어를 순서대로 나열한 것

검색(Searching)은 정렬되지 않는 데이터 혹은 정렬이 된 데이터 중에서 키값에 해당되는 데이터를 찾는 알고리즘

정렬(Sorting)은 흩어져있는 데이터를 키값을 이용하여 순서대로 열거하는 알고리즘

틀린 것: 선형검색은 검색을 수행하기 전에 반드시 데이터의 집합이 정렬되어 있어야 한다

버블 정렬

처음부터 순서대로 각 값과 바로 인접한 다음 값 사이의 크기비교를 통해 정렬

인스펙션(Inspection) 과정

계획 – 사전교육 – 준비 – 인스펙션 회의 – 수정 – 후속조치 or 계획

소프트웨어를 보다 쉽게 이해할 수 있고 적은 비용으로 수정할 수 있도록 겉으로 보이는 동작의 변화 없이 내부구조를 변경하는 것

Refactoring

틀린 것:

Architecting

Specification

Renewal

단위 테스트(Unit Test)와 관련한 설명

구현 단계에서 각 모듈의 개발을 완료한 후 개발자가 명세서의 내용대로 정확히 구현되었는지 테스트

모듈 내부의 구조를 구체적으로 볼 수 있는 구조적 테스트를 주로 시행

테스트할 모듈을 호출하는 모듈도 있고, 테스트할 모듈이 호출하는 모듈도 있다

틀린 것: 필요데이터를 인자를 통해 넘겨주고, 테스트 완료 후 그 결과값을 받는 역할을 하는 가상의 모듈을 테스트 스텁(Stub)이라고 한다

IDE(Integrated Development Environment) 도구의 각 기능에 대한 설명

Coding – 프로그래밍 언어를 가지고 컴퓨터 프로그램을 작성할 수 있는 환경을 제공

Debugging – 프로그램에서 발견되는 버그를 찾아 수정할 수 있는 기능

Deployment – 소프트웨어를 최종 사용자에게 전달하기 위한 기능

틀린 것: Compile – 저급언어의 프로그램을 고급언어 프로그램으로 변환하는 기능

트리구조 순회

전위 순회(Preorder Traversal): root –> left –> right

중위 순회(Inorder Traversal): left –> root -> right

후위 순회(Postorder Traversal): left -> right -> root

인터페이스 구현 시 사용하는 기술, 속성-값 쌍(Attribute-Value Pairs)으로 이루어진 데이터 오브젝트를 전달하기 위해 사용하는 개방형 표준 포맷

JSON

틀린 것:

HTML

AVPN

DOF

순서가 있는 리스트에서 데이터의 삽입(Push), 삭제(Pop)가 한 쪽 끝에서 일어나며 LIFO(Last-In-First-Out)인 자료구조

Stack

틀린 것:

Tree

Graph

Queue

단위테스트 도구

CppUnit: C++

Junit: JAVA

HttpUnit: HTML., JAVA

틀린 것: IgpUnit

<3과목: 데이터베이스 구축>

데이터베이스 정규화

1차 정규화: 각 행마다 열의 값이 1개씩만 있어야함 – 원자값(Atomic value)를 갖는다 열의 값이 여러 개인 경우 여러 행으로 쪼갠다

2차 정규화: 모든 열이 완전 함수적 종속을 만족, 기본키 중 특정 열에만 종속된 열(부분적 종속)이 없어야 한다