

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 07 : 26

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

2020–21년 Associate Architect 양성과정 추가 선발 평가(4월 2차)

OOAD 기본

1. 다음의 Object-Oriented Principles에 대한 설명 중 가장 옳바르지 않은 것은?

- Abstraction : 관심 있는 속성은 남겨두고, 나머지 속성들은 버린다.
- Encapsulation : 내부적으로 어떻게 동작하는지 몰라도 사용하기 쉽게 한다.
- Inheritance : 다양한 동작(Operations)을 하는 클래스들을 정의하기 쉽게 한다.
- Polymorphism : 하나의 Interface에서 상황에 따라 다르게 동작하게 할 수 있다.

2. OOA (Object-Oriented Analysis)에 대한 다음의 설명 중 가장 옳바른 것은?

- 해당 도메인에서 사용되는, 개발할 시스템 내부의 Concepts/Objects를 찾는 단계이다.
- 개발할 Software Object를 정확하게 정의하는 단계이다.
- 사용하는 개발프로세스에 따라 한 번에 수행될 수도 있고, 여러 번 Increment하게 진행될 수도 있다.
- OOA 단계에서는 Class Diagram의 일종인 Domain Model을 개발한다.

3. OOD (Object-Oriented Design)에 대한 다음의 설명 중 가장 옳바른 것은?

- 해당 도메인에서 사용되는 Concepts/Objects를 찾는 단계이다.
- 한 번에 정확하고 완벽하게 진행해야 다음 단계인 OOI (OO Implementation)로 쉽게 진행할 수 있다.
- 시스템 내부의 Object들 간의 Collaboration을 정확하게 표현하는데 집중해야 한다.
- 시스템 내부 Objects와 외부의 Objects 간의 Collaboration을 정확하게 표현하는데 집중해야 한다.

4. 다음은 UML을 구성하는 다양한 다이어그램을 Class Diagram을 이용해서 정리한 그림이다. 이를 중 Iterative한 개발 초기에는 정의되지 않고, 중반 이후에만 새로 정의되어 사용되는 Diagram은 무엇인가?

```

graph TD
    Diagram[Diagram] --> StructureDiagram[Structure Diagram]
    Diagram --> BehaviourDiagram[Behaviour Diagram]
    StructureDiagram --> ClassDiagram[Class Diagram]
    StructureDiagram --> ComponentDiagram[Component Diagram]
    StructureDiagram --> ObjectDiagram[Object Diagram]
    BehaviourDiagram --> ActivityDiagram[Activity Diagram]
    BehaviourDiagram --> UseCaseDiagram[Use Case Diagram]
  
```

답안저장 제출 및 평가종료

SAMSUNG

검색하려면 여기에 입력하십시오.

Windows Taskbar icons: Start, Task View, Edge, File Explorer, Mail, Internet Explorer, Taskbar search bar, Date/Time (2021-04-21), Battery icon.

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 07 : 14

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.
연락처 : 최강원
(kangwon.choi@samsung.com)

Diagram

```

graph TD
    Diagram[Diagram] --> StructureDiagram[Structure Diagram]
    Diagram --> BehaviourDiagram[Behaviour Diagram]
    StructureDiagram --> ClassDiagram[Class Diagram]
    StructureDiagram --> ComponentDiagram[Component Diagram]
    StructureDiagram --> ObjectDiagram[Object Diagram]
    StructureDiagram --> ProfileDiagram[Profile Diagram]
    StructureDiagram --> CompositeStructureDiagram[Composite Structure Diagram]
    StructureDiagram --> DeploymentDiagram[Deployment Diagram]
    StructureDiagram --> PackageDiagram[Package Diagram]
    BehaviourDiagram --> ActivityDiagram[Activity Diagram]
    BehaviourDiagram --> UseCaseDiagram[Use Case Diagram]
    BehaviourDiagram --> InteractionDiagram[Interaction Diagram]
    BehaviourDiagram --> StateMachineDiagram[State Machine Diagram]
    InteractionDiagram --> SequenceDiagram[Sequence Diagram]
    InteractionDiagram --> CommunicationDiagram[Communication Diagram]
    InteractionDiagram --> InteractionOverviewDiagram[Interaction Overview Diagram]
    InteractionDiagram --> TimingDiagram[Timing Diagram]
  
```

Notation: UML

① 한 번에 정확하고 완벽하게 진행해야 다음 단계인 OOI (OO Implementation)로 쉽게 진행할 수 있다.

② 시스템 내부의 Objects들 간의 Collaboration을 정확하게 표현하는데 집중해야 한다.

③ 시스템 내부 Objects와 외부의 Objects 간의 Collaboration을 정확하게 표현하는데 집중해야 한다.

④ 다음은 UML을 구성하는 다양한 다이어그램을 Class Diagram을 이용해서 정리한 그림이다. 이들 중 Iterative한 개발 초기에는 정의되지 않고, 충분 이후에만 새로 정의되어 사용되는 Diagram은 무엇인가?

⑤ Use Case Diagram – Sequence Diagram – Package Diagram

⑥ Package Diagram – Component Diagram – Deployment Diagram

⑦ Use Case Diagram – Component Diagram – Deployment Diagram

⑧ Class Diagram – Package Diagram – Component Diagram

5. UP 기반 OOAD 방법론에 대한 다음의 설명 중 가장 올바르지 않은 것은?

(a) 3~4주 단위로 짧게 반복되는 Mini-Waterfall Iteration으로 진행되므로, Transition Phase 직전까지 Use Cases를 수정해서 User Requirements를 쉽게 변경할 수 있다.

(b) Client가 특별히 중요하게 생각하는 요구사항은 Elaboration Phase에서 명확하게 확정 지어야 한다.

(c) Elaboration Phase에 만든 Software는 버릴 수도 있고, Construction Phase에서 재사용 할 수도 있다.

(d) Elaboration Phase에서는 다양한 Architecture를 시험해 보므로, 일종의 Throw-Away Prototyping을 수행한다고 볼 수 있다.

6. UP의 Inception과 Elaboration Phase에서 사용되는 Use-Case에 대한 다음의 설명 중 올바르지 않은 것은 모두 몇 개인가?

(a) 개발 후반에 결정될 HWU U에 영향을 주지 않는 추상적인 표현을 사용하는 것이 좋다.

(b) 구체적인 Design 및 구현 내용을 미리 가정하고 작성하지 않는 것이 좋다.

(c) UP의 모든 Iteration에서 항상 Fully-Dressed 포맷으로 작성하는 것이 바람직하다.

(d) 시스템이 내부적으로 어떻게 동작할지에 대해서까지 자세히 작성할수도 좋다.

(e) 일부 Actors가 개발할 시스템을 어떻게 사용하는지를 묘사한 Text Story이다.

(f) Main Scenario가 가장 중요하며, Alternatives (Extensions)을 보조적으로 설명을 위해 추가한다.

(g) Use-Case는 일반적으로 Use-Case Diagram을 의미하며, 이 Diagram을 그리는 것이 가장 중요하다.

(h) 주로 실제 SW 구현 방식이나 UML HWU 등에Independent하게 작성하는 것이 좋다.

답안저장

제출 및 평가종료

검색하여 여기에 입력하십시오.

Windows Taskbar

오늘 5:22 2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 07 : 05

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

Windows 10 x64

검색...

강상우(32678)

수 있다.

① Client가 특별히 중요하게 생각하는 요구사항은 Elaboration Phase에서 명확하게 확정 짜어야 한다.

② Elaboration Phase에 만든 Software는 버릴 수도 있고, Construction Phase에서 재사용 할 수도 있다.

③ Elaboration Phase에서는 다양한 Architecture를 시험해 보므로, 일종의 Throw-Away Prototyping을 수행한다고 볼 수 있다.

6. UP의 Inception과 Elaboration Phase에서 사용되는 Use-Case에 대한 다음의 설명 중 올바르지 않은 것은 모두 몇 개인가?

(ㄱ) 개발 후반에 결정될 UI와 UI에 영향을 주지 않는 추상적인 표현을 사용하는 것이 좋다.
(ㄴ) 구체적인 Design 및 구현 내용을 미리 가정하지 작성하지 않는 것이 좋다.
(ㄷ) UP의 모든 Iteration에서 항상 Fully-Dressed 포맷으로 작성하는 것과 바람직하다.
(ㄹ) 시스템이 내부적으로 어떻게 동작할지에 대해서까지 자세히 작성할수록 좋다.
(ㅁ) 외부 Actors가 개별할 시스템을, 어떻게 사용하는지를 묘사한 Text Story이다.
(ㅂ) Main Scenario가 가장 중요하며, Alternatives (Extensions)은 보조적으로 설명을 위해 추가한다.
(ㅅ) Use-Case는 일반적으로 Use-Case Diagram을 의미하며, 이 Diagram을 그리는 것이 가장 중요하다.
(ㅇ) 가능하면 SW 개발 범위 밖에 있는 UI, NW 등에 Independent하게 작성하는 것이 좋다

① 1개
② 2개
③ 3개
④ 4개

7. 다음의 SSD (System Sequence Diagram)에 대한 설명 중 올바른 것은?

① System을 White-Box로 생각하고 자세하게 작성할수록 좋다.
② External Actor와 System 간의 모든 Interactions을 System Operation으로 선정한다.
③ System이 각 System Operation에 대해서 어떻게 내부적으로 동작할 것인지도 충분히 고려해서 작성한다.
④ 특정 Database나 I/O Device의 사용을 미리 가정하고 System Operation을 도출하면 안 된다.

8. Object Design에 대한 다음의 설명 중 올바른 것은 모두 몇 개인가?

(ㄱ) Structure Diagram은 Class나 Package를 정의하는데 도움이 되며, Class Diagram과 Sequence Diagram 등이 해당된다.
(ㄴ) Behavior Diagram은 Method나 Logic 등을 정의하는데 도움이 되며, Object Diagram, Timing Diagram, Activity Diagram 등이 해당된다.
(ㄷ) Class Diagram으로부터 Skeleton Code를 생성하고 구현(Implementation)을 시작하므로, Class Diagram이 가장 중요한 Object Design Model이다.
(ㄹ) Sequence Diagram으로부터 Class를 구성하는 Operation들의 내용(Body)을 유추하여 구현할 수 있으므로, Sequence Diagram이 가장 중요한 Object Design Model이다.
(ㅁ) UP 기반의 OOD에서는 혼(混) Iteration에서 개별할 Use Case에 대해서, Sequence Diagram과 Class Diagram을 반복적(Iterative)이고 점증적(Incremental)으로 작성한다.
(ㅂ) Interaction Overview Diagram을 사용하면 시스템의 다양한 Scenario들을 하나로 연결해서 분석할 수 있다.

① 1개
② 2개
③ 3개
④ 4개

9. 아래 Class Diagram에 대한 설명 중 올바르지 않은 것은?

```

classDiagram
    class Student
    class Course
    class Professor
    Student "*" -- "1..*" Course : enrolls
    Course "1..*" * Professor : teaches
  
```

① 학생(Student)은 최소 한 과목(Course) 이상을 수강신청(enrolls) 해야 한다.
② 교수(Professor)는 최소 한 과목을 담당(teaches) 해야 한다.

답안저장
제출 및 평가종료

검색...
검색하면 여기에 입력하십시오.

오후 5:22
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 56

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

(*) Structure Diagram은 Class나 Package를 정의하는데 도움이 되며, Class Diagram과 Sequence Diagram 등이 해당된다.
(L) Behavior Diagram은 Method나 Logic 등을 정의하는데 도움이 되며, Object Diagram, Timing Diagram, Activity Diagram, 등이 해당된다.
(C) Class Diagram으로부터 Skeleton Code를 생성하고 구현(Implementation)을 시작하므로, Class Diagram이 가장 중요한 Object Design Model이다.
(S) Sequence Diagram으로부터 Class를 구성하는 Operation들의 내용(Body)을 유추하여 구현할 수 있으므로, Sequence Diagram이 가장 중요한 Object Design Model이다.
(D) UP 기반의 OOAD에서는, 원(現) Iteration에서 개발할 Use Cases에 대해서, Sequence Diagrams와 Class Diagram을 반복적(Iterative)이고 점진적(Incremental)으로 작성한다.
(U) Interaction Overview Diagram을 사용하면, 시스템의 다양한 Scenario들을 하나로 연결해서 분석할 수 있다.

① 1개
② 2개
③ 3개
④ 4개

9. 아래 Class Diagram에 대한 설명 중 올바르지 않은 것은?

```

classDiagram
    class Student
    class Course
    class Professor
    Student "*" -- "1..*" Course : enrolls
    Course "*" -- "1..*" Professor : teaches
  
```

① 학생(Student)은 최소 한 과목(Course) 이상을 수강신청(enrolls) 해야 한다.
② 교수(Professor)는 최소 한 과목을 담당(teaches) 해야 한다.
③ 여러 명의 교수가 한 교과목을 가르칠 수 있다.
④ 여러 명의 교수가 함께 팀 티칭 하는 과목 중에 수강신청 인원이 0인 수업이 있을 수도 있다.

10. 다음의 POS 시스템에 대한 초기 버전의 Class Diagram을 보면, Register 클래스와 ProductSpecification 클래스 간의 관계가 실선(Association)으로 표현되어 있다. 아래의 코드를 분석하여, 이 Association Relation이 어떤 Visibility를 표현한 것인지 확인하세요

public class Register
{
 private ProductCatalog catalog;
 private Sale currentSale;

 public Register(ProductCatalog catalog)
}

답안저장 제출 및 평가종료

검색하려면 여기에 입력하십시오.

Windows 10 x64

강상우(32678)

오후 5:23
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 50

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.
연락처 : 최강원
(kangwon.choi@samsung.com)

④ 여러 명의 교수가 함께 팀 티칭 하는 과목 중에 수강신청 인원이 0인 수업이 있을 수도 있다.

10. 다음의 POS 시스템에 대한 초기 버전의 Class Diagram을 보면, Register 클래스와 ProductSpecification 클래스 간의 관계가 실선(Association)으로 표현되어 있다. 아래의 코드를 분석하여, 이 Association Relation이 어떤 Visibility를 표현한 것인지 확인하세요.

```

classDiagram
    class Store {
        address : Address
        name : Text
        addSale()
        Houses
    }
    class Register {
        ...
        endSale()
        enterItem(itemId : ItemID, quantity : Integer)
        makeNewSale()
        makePayment(amount : Money)
    }
    class ProductCatalog {
        Looks-in
        getSpecification()
    }
    class Sale {
        date : Date
        isComplete : Boolean
        time : Time
        becomeComplete()
        makeLineItem(specification : ProductSpecification, quantity : Integer)
        makePayment()
        getTotal()
    }
    class ProductSpecification {
        description : Text
        price : Money
        itemID : ItemID
    }
    class SalesLineItem {
        quantity : Integer
        getSubtotal()
    }
    class Payment {
        amount : Money
    }

    Store "1" -- "1" Register : Houses
    Store "1" -- "1" ProductCatalog : Looks-in
    ProductCatalog "1" -- "1" Sale : Captures
    ProductCatalog "1" -- "1" ProductSpecification : Contains
    ProductSpecification "1..3" -- "1" Sale : Describes
    Sale "1" -- "1" Payment : Paid-by
    Register "*" -- "*" Sale : Logs-completed
  
```

```

public class Register
{
    private ProductCatalog catalog;
    private Sale currentSale;

    public Register(ProductCatalog catalog)
    {
        this.catalog = catalog;
    }

    public void endSale()
    {
        currentSale.becomeComplete();
    }

    public void enterItem(ItemID id, int quantity)
    {
        ProductSpecification desc = catalog.getSpecification(id);
        currentSale.makeLineItem(desc, quantity);
    }

    public void makeNewSale()
    {
        currentSale = new Sale();
    }

    public void makePayment(Money cashTendered)
    {
        currentSale.makePayment();
    }
}
  
```

SAMSUNG

검색하면 여기에 입력하십시오.

답안저장 | 제출 및 평가종료

오후 5:23 2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 47

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

```

        }
        public void endSale()
        {
            currentSale.becomeComplete();
        }

        public void enterItem(ItemID id, int quantity)
        {
            ProductSpecification desc = catalog.getSpecification(id);
            currentSale.makeLineItem(desc, quantity);
        }

        public void makeNewSale()
        {
            currentSale = new Sale();
        }

        public void makePayment(Money cashTendered )
        {
            currentSale.makePayment(cashTendered );
        }
    }
}

```

① A. Parameter Visibility ② B. Local Visibility ③ C. Attribute Visibility ④ D. Global Visibility

11. 다음의 Statechart Diagram에서, start, e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12의 순서로 이벤트가 차례대로 발생되는 경우, 최종적으로 도달된 상태는 무엇인가요?

The screenshot shows a Samsung SWE Expert Academy test interface on a Windows 10 x64 system. The browser address bar shows the URL: <https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc>. The page title is "Windows 10 x64". The user is logged in as 강상우(32678).

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 41

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 답안저장 혹은 제출
해주시기 바랍니다.
- 답안저장 혹은 '제출'을 누르지
않으면 작성한 답안이 제출되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

Q C. Attribute Visibility

Q D. Global Visibility

11. 다음의 Statechart Diagram에서 start, e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12의 순서로 이벤트가 차례대로 발생되는 경우, 최종적으로 도달된 상태는 무엇인가요?

(최대길이 : 4000)
S5.2

12. 다음의 Sequence Diagram에서 해석 가능한 모든 Order of Messages는 총 몇 개 인가요?

(최대길이 : 4000)
12

답안저장 **제출 및 평가종료**

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

12. 다음의 Sequence Diagram에서 해석 가능한 모든 Order of Messages는 총 몇 개인가요?

```

sequenceDiagram
    actor aaa as aaa:A
    actor bbb as bbb:B
    actor ccc as ccc:C
    actor ddd as ddd:D
    aaa->>bbb: a
    bbb-->>ccc: b
    ccc-->>ddd: c
    ddd-->>aaa: d
    aaa-->>bbb: e
    bbb-->>ccc: f
  
```

(최대길이 : 4000)
12

13. 다음의 Class Diagram의 Register 클래스에서 "Public"으로 정의되는 attribute은 모두 몇 개인가요?

```

classDiagram
    class Register {
        id: int
        pw: int
        -name: string
        doSomething(aa: Sale) : void
    }
    class Sale {
        +time: DateTime
        doSomething() : void
    }
    class Store {
        -address: Address
        -doSomething() : void
        +doOthers() : int
    }
    Register "1" -- "1" Sale : currentReg
    Register "1" -- "1" Store : currentReg
    Sale "1" -- "1" Store : location
  
```

(최대길이 : 4000)
1

14. 다음은 학생이 학교 시스템에 로그인 후 등록을 하는 시나리오에 대한 Sequence Diagram입니다. Student는 로그인을 최소/최대 각각 몇 번까지

SAMSUNG

검색하면 여기에 입력하십시오.

답안저장

제출 및 평가종료

오후 5:23
2021-04-21

Windows 10 x64

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

검색...

강상우(32678)

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 17

응시 안내

- 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.
2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

14. 다음은 학생이 학교 시스템에 로그인 후 등록을 하는 시나리오에 대한 Sequence Diagram입니다. Student는 로그인을 최소/최대 각각 몇 번까지 시도할 수 있나요?

```

sequenceDiagram
    actor Student
    actor AdminSystem
    Note over Student: loop(3,5) [incorrect PW]
    Note over AdminSystem: Loop(0,3) [incorrect PW]
    Note over AdminSystem: break [correct PW]
    Student->>AdminSystem: login(ID, PW)
    AdminSystem-->>Student: Login result
    AdminSystem-->>Student: Error message
    Student->>AdminSystem: login(ID, PW)
    AdminSystem-->>Student: Login result
    AdminSystem-->>Student: Additional Notification
    AdminSystem-->>Student: register()
    AdminSystem-->>Student: OK
  
```

(최대길이 : 4000)
최대 8번
최소 3번

답안저장 제출 및 평가종료

SAMSUNG

검색하려면 여기에 입력하십시오.

오늘 5:23 2021-04-21

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 12

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

15. 다음은 교재의 "Process Sale" Use-Case와 관련된 System Sequence Diagram과 Package Diagram의 일부입니다. 가장 기본적인 3 Layered Architecture를 기준으로, Application Logic and Domain Objects Layer에 해당하는 SW를 개발한다고 가정합니다. Cashier로부터 시작된 system operation 'enterItem()'이 어떤 과정을 거쳐 Application Logic and Domain Objects Layer까지 도달하는지 리턴을 포함해서 정확하게 설명하세요. 단, Elaboration Phase의 거의 마지막 단계에서 GUI, UI, IO Devices가 거의 결정되었다고 가정합니다.

the system operations handled by the system in an SSD represent the operation calls on the Application or Domain layer from the UI layer

(최대길이 : 4000)
Cashier는 Swing 구성된 UI 화면에서 ProcessSale Frame 내의 enterItem()을 누른다.
이 액션은 Domain Object내의 Register 모듈에서 처리되게 된다.
Register모듈은 description과 total을 자신을 호출한 Swing UI로 돌려준다.
Swing UI는 화면을 보여주는 UI에 이 결과를 표시하게 되고 Cashier는 화면에 표시된 값을 인지한다.

16. 다음은 일반적인 UP Development Case를 설명하는 그림입니다. Domain Model이 Construction Phase에서 빈칸으로 되어 있는데, 그 빈칸의 의미와 그 이유를 각각 정확하게 설명하세요.

Discipline	Practice	Artifact Iteration→	Incep. I1	Elab. E1..En	Const. C1..Cn	Trans. T1..T2
Business Modeling	agile modeling req. workshop	Domain Model		s		
Requirements	req. workshop vision box exercise dot voting	Use-Case Model Vision Supplementary Specification Glossary	s s s	r r r		
Design	agile modeling test-driven dev.	Design Model SW Architecture Document Data Model		s s s	r r	
Implementation	test-driven dev. pair programming continuous integration coding standards	...				
Project Management	agile PM daily Scrum meeting	...				

답안저장
제출 및 평가종료

검색하려면 여기에 입력하십시오.

Windows 10 x64

2021-04-21 오전 5:23

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 07

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

16. 다음은 일반적인 UP Development Case를 설명하는 그림입니다. Domain Model의 Construction Phase에서 빙간으로 되어 있는데, 그 빙간의 의미와 그 이유를 각각 정확하게 설명하세요.

Discipline	Practice	Artifact Iteration→	Incep. I1	Elab. E1..En	Const. C1..Cn	Trans. T1..T2
Business Modeling	agile modeling req. workshop	Domain Model		s		
Requirements	req. workshop vision box exercise dot voting	Use-Case Model	s	r		
		Vision	s	r		
		Supplementary Specification	s	r		
		Glossary	s	r		
Design	agile modeling test-driven dev.	Design Model		s	r	
		SW Architecture Document		s		
		Data Model		s	r	
Implementation	test-driven dev. pair programming continuous integration coding standards	...				
Project Management	agile PM daily Scrum meeting	...				
...						

(최대길이 : 4000)
Elaboration Phase에서 Domain model은 sequence diagram과 class diagram으로 번영되어 iteration을 통해서 구체화되기 때문에 Construction에서는 구체화 되기 이전의 Domain Model을 사용하지 않는다.

17. 다음의 Sequence Diagram을 보고, System Operation "systemOP()"의 body를 C++ 또는 JAVA 프로그램으로 최대한 정확하게 작성하세요. 단, "systemOP()"와 직접적으로 연관된 부분만 작성합니다.

```

sequenceDiagram
    participant Student
    participant ClassA
    participant ClassB
    participant ClassC
    Note over ClassA: alt [a > b]
    Note over ClassA: alt [a < b]
    Note over ClassA: alt [else]
    Student->>ClassA: systemOP(a,b)
    activate ClassA
    Note over ClassA: alt [a > b]
    ClassA->>ClassB: a(a)
    activate ClassB
    ClassB->>ClassA: aa
    deactivate ClassB
    Note over ClassA: alt [a < b]
    ClassA->>ClassB: b(b)
    activate ClassB
    ClassB->>ClassA: bb
    deactivate ClassB
    Note over ClassA: alt [else]
    ClassA->>ClassB: a'(e)
    activate ClassB
    ClassB->>ClassA: ee
    deactivate ClassB
    deactivate ClassA
    ClassA-->>Student: aa, bb
  
```

답안저장 제출 및 평가종료

검색하면 여기에 입력하십시오.

Windows 10 x64

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

강상우(32678)

오후 5:23
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 06 : 00

응시 안내

1. 17:30에 평가가 종료되며, 그 전에 '답안저장' 혹은 '제출' 해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지 않으면 작성한 답안이 계산되지 않습니다.
- 시간이 경과하여 평가가 종료된 경우, 마지막으로 '답안저장' 한 답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수 없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

17. 다음의 Sequence Diagram을 보고, System Operation "systemOP()"의 body를 C++ 또는 JAVA 프로그램으로 최대한 정확하게 작성하세요. 단, "systemOP()"와 직접적으로 연관된 부분만 작성합니다.

```
(최대길이 : 4000)
public class Student {
    public static void main(String[] args) {
        A objectA = new A();
        int a = 5;
        int b = 3;
        ...
    }
}
```

18. (문항14)의 Sequence Diagram은 작성자의 의도와는 다르게, 로그인에 실패하더라도 등록을 할 수 있게 모델링 되어 있습니다. 로그인에 성공한 학생만 등록을 수행할 수 있고, 최대 7번까지 로그인을 시도할 수 있도록 Sequence Diagram을 수정하세요.

32678_강상우_OOD 기본_설계형18.png

19. 다음은 University Information System (UIS)에 대한 요구사항입니다. 이를 잘 반영하는 Class Diagram을 작성하세요. 단, Aggregation을 세 번 이상 사용하세요.

A. 학교(University)는 최소 3개 최대 15개의 단과대학(School)으로 구성되어 있다.
B. 각 단과대학은 최소 1개, 최대 10개의 학과(Department)로 구성되어 있다.
C. 학교는 직원(Staff)과 교수(Professor)를 고용한다.
D. 각 단과대학은 1명의 학장교수(Dean), 각 학과는 1명의 주임교수(Head)가 배정되어 있다.
E. 각 학과는 적어도 1명 이상의 교수와 2명의 직원으로 구성된다.
F. 모든 교수는 적어도 한 학과에 소속되어야 한다. 여러 학과에 소속될 수도 있다.
G. 모든 학생(Student)은 적어도 한 학과에 소속되어야 한다. 여러 학과에 소속될 수도 있다.
H. 모든 학생은 여러 과목을 수강(Enroll)할 수 있다.
I. 모든 교수는 최소 1개, 최대 5 과목(Course)을 강의(Teach)한다.
J. 한 과목은 최소 1개, 최대 3개의 분반(Class)으로 구성될 수 있다.

32678_강상우_OOD 기본_설계형19.png

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

Windows 10 x64

out.swexpertacademy.sams... X

강상우(32678)

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 55

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

19. 다음은 University Information System (UIS)에 대한 요구사항입니다. 이를 잘 반영하는 Class Diagram을 작성하세요. 단, Aggregation을 세 번 이상 사용하세요.

A. 학교(University)는 최소 1개 최대 15개의 단과대학(School)으로 구성되어 있다.
B. 각 단과대학은 최소 1개 최대 10개의 학과(Department)로 구성되어 있다.
C. 학교는 직원(Staff)과 교수(Professor)를 고용한다.
D. 각 단과대학은 1명의 학장교수(Dean), 각 학과는 1명의 주임교수(Head)가 배정되어 있다.
E. 각 학과는 적어도 1명 이상의 교수와 2명의 직원으로 구성된다.
F. 모든 교수는 적어도 한 학과에 소속되어야 한다. 여러 학과에 소속될 수도 있다.
G. 모든 학생(Student)은 적어도 한 학과에 소속되어야 한다. 여러 학과에 소속될 수도 있다.
H. 모든 학생은 여러 과목을 수강(Enroll)할 수 있다.
I. 모든 교수는 최소 1개 최대 5 과목(Course)을 강의(Teach)한다.
J. 한 과목은 최소 1개, 최대 3개의 분반(Class)으로 구성될 수 있다.

32678_강상우_OOAD 기본_설계형19.png [찾아보기...]
현재 첨부되어 있는 파일 : 32678_강상우_OOAD 기본_설계형19.png

20. (문항19)에서 작성한 University Information System (UIS)에 대한 Class Diagram을 기반으로, 다음을 잘 반영하는 Component Diagram (Composite Structure Diagram)의 draft를 작성하세요. UIS의 Class Diagram에 있는 모든 클래스와 관계들이 Component Diagram을 구성하는 Component 내부에 빼김없이 포함되도록 작성하세요.

A. UIS는 "浣사 시스템"과 "행정 시스템"으로 구성되며, 각각 독립적인 Server에 Deploy 된다.
B. 필수한 경우, 데스크톱이나 DB Server(s)의 운영할 수 있다.
C. 시스템-시스템, 시스템-서버 간에는 다른 네트워크 프로토콜을 사용한다.

32678_강상우_OOAD 기본_설계형20.png [찾아보기...]
현재 첨부되어 있는 파일 : 32678_강상우_OOAD 기본_설계형20.png

Design Pattern 기본

1. 설계 패턴에 대한 설명으로 가장 옳은 것은?

① 애자일 방식의 소프트웨어 개발 방법을 따를 경우 GRASP 패턴은 적용하기 용이하지만 GoF 패턴은 적용하기 쉽지 않다.

② 일반적으로 설계 패턴은 바로 불러 쓸 수 있는 코드 형태의 라이브러리(library)로 배포되며, 이중 가장 인기있는 것이 GoF 패턴이라 불린다.

③ MVC 패턴은 관점에 따라 코딩 패턴으로 분류되기도 한다.

④ 설계 패턴은 재사용 가능하며 추상화 수준이 높은 설계 지식이다.

2. 설계에 관련된 설명으로 가장 옳은 것은?

① 저수준의 모듈이 고수준의 모듈에 의존관계를 가져서는 안된다.

② 반복자(iterator) 패턴에 DIP 설계원칙이 적용되어 있다.

③ 하나의 클래스에는 메소드를 한 개만 두는 것이 좋고, 이를 SRP라고 칭한다

④ 좋은 설계를 만들기 위해 클래스간 연결에서 association 보다 composition을 선호하는 것이 좋다.

3. 설계 원칙 OCP에 대한 설명으로 옳은 것을 모두 고르면?

(a) 전략(strategy) 패턴에 해당 설계 원칙이 잘 적용되어 있다.
(b) 장식자(decorator) 패턴에 해당 설계 원칙이 잘 적용되어 있다.
(c) 해당 설계 원칙을 모든 모듈에 대해 지지도로 하는 경우 오히려 불필요한 복잡도와 수행속도 감소가 일어날 수 있다.

① (a), (b)

② (a), (c)

③ (b), (c)

SAMSUNG

검색하면 여기에 입력하십시오.

답안저장

제출 및 평가종료

오후 5:24
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 47

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

3. 설계 원칙 OCP에 대한 설명으로 좋은 것을 모두 고르면?

(a) 전략(strategy) 패턴에 해당 설계 원칙이 잘 적용되어 있다.
(b) 장식자(decorator) 패턴에 해당 설계 원칙이 잘 적용되어 있다.
(c) 해당 설계 원칙을 모든 모듈에 대해 지속화하는 경우 오히려 불필요한 복잡도와 수행속도 감소가 일어날 수 있다.

① (a), (b)
② (a), (c)
③ (b), (c)
④ (a), (b), (c)

4. 객체 합성(object composition)과 위임(delegation)을 통해 동작하는 패턴을 모두 고르면?

(a) 템플릿 메소드(template method) 패턴
(b) 빌더(builder) 패턴
(c) 추상 팩토리(abstract factory) 패턴

① (a), (b)
② (a), (c)
③ (b), (c)
④ (a), (b), (c)

5. MVC 패턴에 대해 가장 올바른 설명은?

① Controller가 Model을 사용하기 위해 전략(Strategy) 패턴이 적용된다.
② Model 코드는 Controller 타입(type)의 변수를 가지고 있고 Controller 타입이 선언하는 API를 직접 호출할 수 있다.
③ 하나의 Model에 대해 여러 View가 연결될 수 있다.
④ 복합체(composite) 패턴은 View에만 적용되어야 한다.

6. 다음의 설명과 가장 관련이 깊은 설계 원칙은?

"서브 클래스(subclass)는 부모 클래스(superclass)의 전제 조건(precondition)을 유지하거나 약화(weaken)시켜야 한다."

① Information Expert Principle
② Liskov's Substitution Principle
③ Open Closed Principle
④ Pure Fabrication Principle

7. 다음 상황과 가장 밀접한 설계 악취(design smell)를 고르시오.
"시스템의 어느 한 부분을 수정했는데 전혀 관련되지 않다고 생각한 다른 부분이 잘못 동작한다."

① 신적 존재(God class)
② 불필요한 복잡성(needless complexity)
③ 취약성(fragility)
④ 부동성(immobility)

답안저장 제출 및 평가종료

검색하려면 여기에 입력하십시오.

Windows 10 x64

강상우(32678)

오후 5:24
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 42

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

7. 다음 상황과 가장 밀접한 설계 악취(design smell)를 고르시오.
"시스템의 어느 한 부분을 수정했는데 전혀 관련되지 않다고 생각한 다른 부분이 잘못 동작한다."

- ① 신적 존재(God class)
- ② 불필요한 복잡성(needless complexity)
- ③ 취약성(fragility)
- ④ 부동성(immobility)

8. 팩토리 메소드(factory method) 패턴의 정의에 사용된 설계 개념을 모두 고른 것은?
(a) 객체 합성(object composition)
(b) kapsul화(encapsulation)

- ① (a)
- ② (b)
- ③ (a), (b)
- ④ 답 없음

9. UML 클래스 다이어그램에서 좋은 문장을 모두 고른 것은?
(a) 어떤 클래스가 추상 클래스(abstract class)라면 추상 메소드(abstract method)를 하나 이상 포함해야 한다.
(b) 추상 클래스(abstract class)에는 메소드의 구현 부분이 나타날 수 있다.
(c) 구상 클래스(concrete class)에 정의된 메소드는 추상 클래스(abstract class)를 인자(parameter)로 사용할 수 없다.

- ① (a), (b)
- ② (a), (c)
- ③ (b), (c)
- ④ (a), (b), (c)

10. 감시자(Observer) 패턴과 중재자(Mediator) 패턴에 대해 가장 올바른 것은?
(a) 두 패턴 모두 통신(communication)에 참여하는 객체들간 강한 결합(tight coupling)을 만들어낸다.
(b) 중재자 패턴이 적용된 클래스는 코드의 재사용성이 높은 편이다.
(c) 중재자 패턴의 구현에서 싱글턴(singleton) 패턴을 이용해서 중재자 객체를 만드는 경우가 많다.
(d) 두 패턴 모두 GoF 분류 기준으로 구조 패턴(structural pattern)이다.

11. 다음 클래스 다이어그램에서 A1을 변경했을 때, 그 영향을 받을 수 있는 모든 클래스와 인터페이스를 나열하시오. (단, 그림에 나타나지 않은 hidden coupling은 없다고 가정한다)
(이미지(1))

```

classDiagram
    class C4
    class Int1 {
        <<interface>>
    }
    class C3
    C4 --> Int1
    Int1 --> C3
    C3 --> C4
  
```

답안저장 제출 및 평가종료

검색하면 여기에 입력하십시오.

Windows 10 x64

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

강상우(32678)

오후 5:24
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 37

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

11. 다음 클래스 다이어그램에서 A1을 변형했을 때, 그 영향을 받을 수 있는 모든 클래스와 인터페이스를 나열하시오. (단, 그림에 나타나지 않은 hidden coupling은 없다고 가정한다)
<이미지(1)>

```

classDiagram
    class C4
    class Intf
    class C3
    class A1
    class A2
    class C1
    class C2

    C4 <--> Intf
    C4 --> A1
    Intf -.-> C3
    Intf -.-> A2
    A1 --> C1
    A1 -.-> A2
    A2 --> C2
    C1 --> C2
  
```

(최대길이 : 4000)
C4, Intf, A2, C1

12. 다음 클래스 다이어그램을 자바 혹은 C++로 구현했을 때, 예리 없이 동작 되는 코드를 (a)~(i) 중에 모두 고르시오.
<이미지(1)>

(a) A1 a = new C1(); // in Java
A1 *a = new C1(); // in C++
(b) A1 b = new C2(); // in Java
A1 *b = new C2(); // in C++
(c) A1 c = new C4(); // in Java
A1 *c = new C4(); // in C++
(d) C1 d = new C2(); // in Java
C1 *d = new C2(); // in C++
(e) Intf e = new C3(); // in Java
Intf *e = new C3(); // in C++
(f) Intf f = new C4(); // in Java
Intf *f = new C4(); // in C++
(g) Int2 g = new A1(); // in Java
Int2 *g = new A1(); // in C++
(h) Int2 h = new C4(); // in Java
Int2 *h = new C4(); // in C++
(i) Int2 i = new C1(); // in Java
Int2 *i = new C1(); // in C++

```

classDiagram
    class Intf
    class Int2
    class C4
    class A1
    class C1
    class C5

    Intf -.-> Int2
    Intf -.-> A1
    Int2 -.-> C4
    Int2 --> A1
    A1 --> C1
    A1 -.-> C5
    C1 --> C5
  
```

답안저장 제출 및 평가종료

검색하려면 여기에 입력하십시오.

Windows 10 x64

2021-04-21 오후 5:24

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 31

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.
연락처 : 최강원
(kangwon.choi@samsung.com)

12. 다음 클래스 다이어그램을 자바 혹은 C++로 구현했을 때, 예리 없이 동작 되는 코드를 (a)~(i) 중에 모두 고르시오.

<이미지(1)>

```

(a) A1 a = new C1(); // in Java
A1 *a = new C1(); // in C++
(b) A1 b = new C2(); // in Java
A1 *b = new C2(); // in C++
(c) A1 c = new C4(); // in Java
A1 *c = new C4(); // in C++
(d) C1 d = new C2(); // in Java
C1 *d = new C2(); // in C++
(e) Int1 e = new C3(); // in Java
Int1 *e = new C3(); // in C++
(f) Int1 f = new C4(); // in Java
Int1 *f = new C4(); // in C++
(g) Int2 g = new A1(); // in Java
Int2 *g = new A1(); // in C++
(h) Int2 h = new C4(); // in Java
Int2 *h = new C4(); // in C++
(i) Int2 i = new C1(); // in Java
Int2 *i = new C1(); // in C++

```

```

classDiagram
    class C1 {
        <<C1>>
    }
    class C2 {
        <<C2>>
    }
    class C3 {
        <<C3>>
    }
    class C4 {
        <<C4>>
    }
    class C5 {
        <<C5>>
    }

    interface Int1 {
        <<Int1>>
    }
    interface Int2 {
        <<Int2>>
    }

    abstract class A1 {
        <<A1>>
    }

    C1 --> A1
    C2 --> A1
    C3 --> C1
    C4 --> A1
    C5 --> A1

    Int1 <|-- C1
    Int1 <|-- C2
    Int2 <|-- A1
    Int2 <|-- C5

    C1 <|-- C3
    C4 <|-- A1
    C5 <|-- A1

```

(최대길이 : 4000)
(a), (e), (i)

13. 다음 설명에 가장 적합한 GoF 패턴 이름을 쓰시오.

"이 패턴의 의도(intent)는 기존 객체에 동적으로 새로운 책임(responsibility)을 동적으로(dynamically) 추가할 수 있게 하는 것이다. 어떤 클래스에 새로운 책임을 추가하고자 할 때 상속(inheritance)을 이용하여 서브클래스(subclass)를 만들어서 해당 책임에 관련된 메소드를 추가할 수도 있지만, 이렇게 하면 정적으로(statically) 된다는 단점이 있다. 이 패턴은 서브클래스를 생성하는 것보다 유연한 방법을 제공한다. 대표적인 예로 Java 언어의 I/O 관련된 클래스 구성에 이 패턴이 적용되어 있다. 이 패턴은 목적적으로 보면 구조적(structural) 패턴에 속한다."

(최대길이 : 4000)
Decorator Pattern

14. 다음 설명에 가장 적합한 GoF 패턴 이름을 쓰시오.

"글꼴 다이얼로그(Font Dialog)의 구성요소들 간에 M:N의 복잡한 통신 메커니즘(communication mechanism)을 요구한다. 예를 들어 어떤 특정한 글꼴을 콤보박스(combo box)를 통해 선택하면 해당 글꼴이 해당하는 글꼴 스타일과 크기 정보를 각각의 리스트 박스에 표시해야 한다. 또한 미리 보기 영역에 선택된 옵션에 따른 글꼴의 모습을 미리주어어야 한다. 구성 요소들끼리 직접 통신을 하기보다는 허브(hub)의 역할을 하는 객체를 두고 이 객체를 두고 이 객체가 구성요소들간 통신 메커니즘을 캡슐화(encapsulation)한다."

(최대길이 : 4000)

답안저장 제출 및 평가종료

검색하여 여기에 입력하십시오.

Windows 10 x64

오후 5:24
2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 26

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

14. 다음 설명에 가장 적합한 GoF 패턴 이름을 쓰시오.
 '글꼴 디아ログ(Font Dialog)의 구성요소들 간에 M:N의 복잡한 통신 메커니즘(communication mechanism)을 요구한다. 예를 들어 어떤 특정한 글꼴을 콤보박스(combo box)를 통해 선택하면 해당 글꼴이 이용하는 글꼴 스타일과 크기 정보를 각각의 리스트 박스에 표시해야 한다. 또한 미리 보기 영역에 선택된 옵션에 따른 글꼴의 모습을 보여주어야 한다. 구성 요소들끼리 직접 통신을 하기보다는 허브(hub)의 역할을 하는 객체를 두고 이 객체가 구성 요소들간 통신 메커니즘을 kapsul화(encapsulation)함으로써 해당 문제를 해결한다.'

(최대길이 : 4000)
 Mediator Pattern

15. 속집도(cohesion)는 무엇을 뜻하는지 설명하고, 반복자(iterator) 패턴에서 반복자(iterator) 클래스와 집합.aggregate 혹은 collection 클래스 및 그들의 관계를 속집도 관점에서 논하시오.
 (최대길이 : 4000)
 [호감도란 클래스의 elements들이 특정 목적이나 역할을 위해 얼마나 일관되게 지원하는 정도라 할 수 있다.
 각 기능들이 서로 연관되어 하나의 목표지향성을 가질 때 속집도가 높다고 할 수 있다.]
 Iterator 패턴에서 iterator 클래스는 iterator의 객체를 생성하는 역할을 담당한다. 생성과 순회의 방법 각 기능에 따라 서로 분리되어 있고, 이를 각각의 클래스(iterator, aggregate)의 속집도를 높이는 결과를 만들어낸다.

16. 워드우즈의 기본 프로그램인 그림판(Paint)과 유사한 그래픽 도형 에디터 프로그램을 개발하였는데, 설계의 구현에서 MVC 패턴을 충실히 적용하였다고 가정하자. 해당 프로그램의 다음과 같은 기능을 (a) ~ (c)와 같이 변경하고자 한다. (a), (b), (c) 각각의 경우에 따라 코드 변경이 필요한 MVC의 컴포넌트 (Model, View, Controller)를 모두 명시하고 이 (답안 시각적 공정의 예를 든다면 "a)의 경우 View만 수정: 이유는 ..., b)의 경우 Model과 View만 수정: 이유는 ..." 등과 같이 (a),(b),(c) 각각에 대해 변경되는 컴포넌트와 이유를 포함하여 기재).
 (a) 작성된 그림을 jpeg으로 export하는 기능이 기존에도 있었는데, 보다 압축률이 우수한 새로운 jpeg 변환 알고리즘을 사용하도록 변경하고자 함.
 (b) 사용자가 그림 영역에서 오른쪽 마우스 클릭을 하면 기존에는 Help 기능을 호출하였는데, 이제는 편의 색을 바꿀 수 있는 팝업 메뉴가 나타나게 변경하고자 함. 단, 팝업 메뉴를 다루는 메소드는 기존의 Help 기능을 다루는 메소드와 다른 메소드임.
 (c) 사용자는 그림 영역에서 copy & paste 작업을 하기 위해 마우스로 드래그해서 사각형의 영역을 설정하는데, 이 사각형의 테두리 색을 기존의 노란색에서 빨간색으로 변경하고자 함.

(최대길이 : 4000)
 (a) > 모델만 수정
 이유는 모델에서 압축 알고리즘을 사용하여 변환하는 역할을 맡고 있기 때문. 같은 jpeg이므로 화면이나 컨트롤러에서 수정할 부분은 없고, 모델에서의 변경만 필요함
 (b) > 컨트롤러
 (c) > View만 수정
 이유는 View에서 오른쪽 마우스 클릭을 하면 기존에는 Help 기능을 호출하였는데, 이제는 편의 색을 바꿀 수 있는 팝업 메뉴가 나타나게 변경하고자 함. 단, 팝업 메뉴를 다루는 메소드는 기존의 Help 기능을 다루는 메소드와 다른 메소드임.

17. 다음 클래스 다이어그램은 전략 패턴의 적용 예를 나타낸다. 클래스 다이어그램을 충실히 참조하여, runStrategy 메소드와 setStrategy 메소드를 작성하시오.

<이미지(1)>

```

classDiagram
    class Context {
        +void selfStrategy(Strategy)
        +void runStrategy()
    }
    class Strategy {
        +execute()
    }
    class ConcreteStrategyA {
        +execute()
    }
    class ConcreteStrategyB {
        +execute()
    }
    class ConcreteStrategyC {
        +execute()
    }

    Context "1" -- "1" Strategy : algo
    Context "1" -- "1" ConcreteStrategyA : selfStrategy
    Context "1" -- "1" ConcreteStrategyB : selfStrategy
    Context "1" -- "1" ConcreteStrategyC : selfStrategy
    Context "1" -- "1" ConcreteStrategyA : runStrategy
    Context "1" -- "1" ConcreteStrategyB : runStrategy
    Context "1" -- "1" ConcreteStrategyC : runStrategy
  
```

(최대길이 : 4000)
 abstract class Context {
 Strategy algo;
 }
 Strategy {
 +execute();
 }
 ConcreteStrategyA {
 +execute();
 }
 ConcreteStrategyB {
 +execute();
 }
 ConcreteStrategyC {
 +execute();
 }

답안저장
제출 및 평가종료
▲ ▾ 2021-04-21 오후 5:24

https://out.swexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainfileName=lc

out.swexpertacademy.sams... X

Windows 10 x64

강상우(32678)

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 20

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.
연락처 : 최강원
(kangwon.choi@samsung.com)

18. 게임에 등장하는 객체들에는 Marine, Tank, JetFighter가 있고, 동일한 이름의 클래스가 정의되어 있다. 초기의 설계 결과는 다음과 같다.
공격성이 강화된 스팀팩(Stimpack) 기능이 적용된 객체들을 표현할 수 있게 Stimpacked_Marine, Stimpacked_Tank, Stimpacked_JetFighter 클래스를 만들었다. 또한 방어성 강화(Armored) 기능이 적용된 객체들을 표현할 수 있게 Armored_Marine, Armored_Tank, Armored_JetFighter 클래스를 만들었다. 스팀팩과 방어성 강화 기능이 모두 적용된 객체들을 표현할 수 있도록 Stimpacked_Armored_Marine, Stimpacked_Armored_Tank, Stimpacked_Armored_JetFighter 클래스를 만들었다.
향후에도 새로운 기능들이 추가될 수 있는데, 기존의 설계는 기능들의 조합을 표현하기 위해 너무나 많은 수의 클래스 정의를 요구한다.

(a) 장식자(Decorator) 패턴을 사용하여 위의 요구사항들을 표현할 수 있도록 새로 설계하고 코드(자바 혹은 C++)로 나타내시오.
(b) 위 (a) 문제에서 담은 내용을 기반으로, 클라이언트가 다음의 객체(들)을 만들어내는 코드(자바 혹은 C++)를 나타내시오.
- 스팀팩 기능이 있는 Marine
- 스팀팩 기능이 있고 방어성 강화 기능이 있는 Marine

32678_강상우_Design Pattern 기본_설계형18.txt

현재 첨부되어 있는 파일 : 32678_강상우_Design Pattern 기본_설계형18.txt

19. 다음의 코드를 보고, 템플릿 메소드(template method) 패턴을 적용하여 리팩토링 하시오. 참고로 Math.pow 메소드는 Math 유틸리티 클래스에서 제공하는 메소드임.

```
class ArithmeticCalc {
    int result;
    void compute(int a, int f, int c) {
        result = 0;
        if (c <= 0) return;
        for (int n = 1; n <= c; n++) {
            result += (a + (n - 1) * f);
        }
        result = result / c;
        System.out.println("Result = " + result);
    }
}
class GeometricCalc {
    int result;
    void compute(int a, int f, int c) {
        result = 1;
        if (c <= 1) return;
        for (int n = 1; n <= c; n++) {
            result *= (a + (int) Math.pow(f, n - 1));
        }
        result = result / (c - 1);
        System.out.println("Result = " + result);
    }
}
```

32678_강상우_Design Pattern 기본_설계형19.txt

현재 첨부되어 있는 파일 : 32678_강상우_Design Pattern 기본_설계형19.txt

20. 다음 클래스 다이나그램 및 관련 코드는 SOLID 원칙을 위반하고 있다. 향후에도 Client1은 service1과 service2만을 사용하고 Client2는 service3만 사용하는 형태로 남을 것이다. 또한 Client3는 service1, service2, service3를 모두 사용할 것이다. 단, service1, service2, service3는 모두 동종(homogeneous)의 책임(responsibility)을 맡은 메소드들로 가정한다. 또한 ServiceInterface를 사용하는 다른 클래스들이 더주 있다고 가정한다.

```
class Client1 {
    public void m1(ServiceInterface foo) {
        foo.service1();
        foo.service2();
    }
}

class Client2 {
    public void m2(ServiceInterface foo) {
        foo.service3();
    }
}

class Client3 {
    public void m3(ServiceInterface foo, int control) {
        if (control == 0) foo.service1();
        else {
            foo.service2();
        }
    }
}
```

SAMSUNG

답안저장

제출 및 평가종료

검색

5 24

2021-04-21

평가 시간 정보

평가 시작 시간
2021년 04월 21일 13:30

평가 종료 시간
2021년 04월 21일 17:30

남은 시간
00 : 05 : 15

응시 안내

1. 17:30에 평가가 종료되며,
그 전에 '답안저장' 혹은 '제출'
해주시기 바랍니다.
- '답안저장' 혹은 '제출'을 누르지
않으면 작성한 답안이 계산되지
않습니다.
- 시간이 경과하여 평가가 종료된
경우, 마지막으로 '답안저장' 한
답안이 계산됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

32678_강상우_Design Pattern 기본_설계형19.txt

현재 첨부되어 있는 파일 : 32678_강상우_Design Pattern 기본_설계형19.txt

20. 다음 클래스 디자인 그림 및 관련 코드는 SOLID 원칙을 위반하고 있다. 향후에도 Client1은 service1과 service2만을 사용하고 Client2는 service3만 사용하는 형태로 남을 것이다. 또한 Client3는 service1, service2, service3를 모두 사용할 것이다. 단, service1, service2, service3는 모두 동종(homogeneous)의 책임(responsibility)을 맡은 메소드들로 가정한다. 또한 ServiceInterface를 사용하는 다른 클래스들이 다수 있다고 가정한다.

```

class Client1 {
    public void m1(ServiceInterface foo) {
        foo.service1();
        foo.service2();
    }
}

class Client2 {
    public void m2(ServiceInterface foo) {
        foo.service3();
    }
}

class Client3 {
    public void m3(ServiceInterface foo, int control) {
        if (control == 0) foo.service1();
        else {
            foo.service2();
            foo.service3();
        }
    }
}

```

<이미지(1)>

(a) SOLID 원칙 중 위반된 것을 1개 고르고, 위반된 원칙의 이름과 함께 그렇게 생각하는 이유를 쓰시오.
(b) 기존 코드의 변경을 최소화하면서 해당 원칙을 지킬 수 있도록 위 설계를 수정 하시오.

```

classDiagram
    class Client3 {
        +m3(ServiceInterface, int)
    }
    class Client1 {
        +m1(ServiceInterface)
    }
    class Client2 {
        +m2(ServiceInterface)
    }
    class ServiceInterface {
        +service1()
        +service2()
        +service3()
    }
    class ServiceImpl

    Client3 --> ServiceInterface : +m3(ServiceInterface, int)
    Client1 --> ServiceInterface : +m1(ServiceInterface)
    Client2 --> ServiceInterface : +m2(ServiceInterface)
    ServiceInterface <|-- ServiceImpl

```

32678_강상우_Design Pattern 기본_설계형20.txt

현재 첨부되어 있는 파일 : 32678_강상우_Design Pattern 기본_설계형20.txt

SAMSUNG

검색하려면 여기에 입력하십시오.

답안저장

제출 및 평가종료

오후 5:24
2021-04-21