

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 37

응시 안내

1. 17:30에 평가가 종료되며,
그 경계에 당연하게 혹은 '제출'
라고 표기되는 경우는 평가가 종료된다.
당연히 해당 평가는 평가를 놓았기
않았다면 평가가 종료될
경우 마지막으로 '당연'한
당연이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

2020-21년 Associate Architect 양성과정 추가 선발 평가(4월 1차)

OOAD 기본

1. 다음은 Object-Oriented Principle 중 "Composition"에 대한 설명이다. 옮바르지 않은 것은?

- 한 Object이 다른 Objects를 포함(Composing)하는 방법이다.
- "Has_a" 관계와 불현타.
- Run-Time에 Compatible한 다른 Object로 교체할 수 있게 한다.
- Shared Aggregation이 Composition 보다 강력한 (Exclusive) 포함 관계를 가진다.

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

- 해당 도메인에서 사용되는, 개발할 시스템 외부의 Concepts/Objects를 찾는 단계이다.
- 개발할 Software Object를 정확하게 정의하는 단계이다.
- 한 번에 정확하고 완벽하게 진행해야 다음 단계인 OOD (Object-Oriented Design)로 쉽게 진행할 수 있다.
- 해당 도메인 범위 안에 있는 Concepts/Objects를 찾는 데 집중하는 것이 좋다.

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

- User Requirements를 Use-Case를 이용해서 정확하게 정의하는 단계이다.
- 개발할 Software Object를 정확하게 정의하는 단계이다.
- 한 번에 정확하고 완벽하게 진행해야 다음 단계인 OOI (OO Implementation)로 쉽게 진행할 수 있다.
- 시스템 내부 Objects와 외부의 Objects를 같은 Collaboration을 정확하게 표현하는데 집중해야 한다.

4. 다음과은 UML을 구현하는 다양한 디자인 그램을 Class Diagram을 이용해서 정리한 그림이다. 이를 참고하여, 다음과은 UML에 대한 설명 중 가장 옳바른 것은 고르세요.

```

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]
    ClassDiagram --> ProfileDiagram[Profile Diagram]
    ClassDiagram --> CompositeStructureDiagram[Composite Structure Diagram]
    ClassDiagram --> DeploymentDiagram[Deployment Diagram]
    ClassDiagram --> PackageDiagram[Package Diagram]
    ActivityDiagram --> InteractionDiagram[Interaction Diagram]
    ActivityDiagram --> StateMachineDiagram[State Machine Diagram]
    InteractionDiagram --> SequenceDiagram[Sequence Diagram]
    InteractionDiagram --> CommunicationDiagram[Communication Diagram]
    InteractionDiagram --> InteractionOverviewDiagram[Interaction Overview Diagram]
    InteractionDiagram --> TimingDiagram[Timing Diagram]
  
```

Notation: UML

© UML은 artifacts of a system을 다양한 관점과 수준에서 모델링 하는데 사용된다.

답안지장 계출 및 평가종료

▲ 5.6% 2021-04-07

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 35

응시 안내

1. 17:30에 평가가 종료되며,
그 경기 당연히 가장 혹은 '제출'
라고 표기됩니다.
"당안자장" 혹은 "제출"을 누르기
앞으면 작동한 답변이 제출되지
않습니다.
- 시스템은 평가자가 종료될
경우, 마지막으로 "당안자장" 한
답변이 제출됩니다.

2. 평가가 후에는 평가를 중단할 수
있습니다.

연락처 : 최강원
(kangwon_choi@samsung.com)

평가 시간 정보

Notation: UML

Sequence Diagram Communication Diagram Interaction Overview Diagram Timing Diagram

① UML은 artifacts of a system을 다양한 관점과 수준에서 모델링 하는데 사용된다.

② Sketch 수준에서 Intercommunication Medium으로 UML을 사용하더라도, UML 문법은 경작하게 지켜야 한다.

③ UML은 Software System뿐만 아니라 다양한 사용, 환경, 비즈니스 토크 등을 모델링 할 때에도 사용될 수 있다.

④ Blueprint 수준으로 UML을 사용할 때에는, Implementation (OOI) 단계에서 훈련을 주지 않도록, 디자인 의도를 경작하게 표현해야 한다.

5. UP 기반 OOA&D 방법론에 대한 다음의 설명 중 가장 올바르지 않은 것은?

① 원형에서 사용되는 실질적인 표준 개발방법론이다.

② Agile 기법에 기반을 두고 있어, 전통적인 기법인 Waterfall Model과는 크게 관련이 없다.

③ Elaboration Phase에서는 User Requirements가 크게 수정될 수 있다.

④ Elaboration Phase는 마무리되면 Stakeholder 간에 합의하고 다양한 방법으로 검증된 하나의 Software Architecture가 확정된다.

6. UP의 Inceptional Elaboration Phase에서 사용되는 Use-Case에 대한 다음의 설명 중 올바른 것은 모두 몇 개인가?

(-) 가장 초기에 결정한 UP의 UML에 영향을 주지 않는 유기적인 표물을 사용하는 것이 좋다.
(.) 구체적인 Design 및 구현 내용을 미리 가정하고 각 하위 기능은 어떤 미리 챙겨야 한다.
(.) 시스템이 대부분으로 보통 몇 개의 통합한 기능에 대해서까지 차세대 작성할수록 좋다.
(.) 시스템을 구성하는 각각의 기능은 그 자체로 독립적으로 사용된다.
(d) Construction Phase 단계가 전략적 단계로 업데이트 된다.
(w) 외부 Actor가 개별 할 시스템을 어떻게 사용하는지를 묘사한 Text Story이다.
(x) Use-Case는 일반적으로 Use-Case Diagram을 외부로 Diagram을 그려는 것이 가장 중요하다.
(o) Use-Case는 Actor는 시스템 사용되는 실제 사용자에게 시스템 사용자를 위하여 다른 시스템을 거치하는지는 않는다

① 1개

② 2개

③ 3개

④ 4개

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

① External Actor와 System 간의 모든 Interactions 중 System으로 들어오는 것을 System Operation으로 선정한다.

② System이 각 System Operation에 대해서 어떻게 내부적으로 동작할 것인가를 충분히 고려해서 작성한다.

③ 각 Use Case에서 도출될 수 있는 System Operation들을 하는 것을 목표로 한다.

④ 일상적으로, External Actor로부터 Message가 들어오는 것(Actor-Activating)이 아닌, 시스템 내부의 일정 조건이 만족되었을 때 시작되는 Use Case에 대해서는, SSD를 작성할 필요가 없다.

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

(-) Structure Diagram은 Class나 Package를 기본 단위로 모으기이며, Class Diagram, Package Diagram, Component Diagram 등이 해당된다.
(.) Behavior Diagram은 Methods나 Logic 등을 결합하는데 도움이 되며, Interaction Diagrams, State Machine Diagram, Activity Diagram 등이 해당된다.
(c) Class Diagram은 System의 모든 Class와 그 구현(Implementation)을 표기하는 Diagram으로, System Diagram이나 Class Diagram과 가장 중요한 Object Design Model이다.
(d) Object Diagram은 System으로부터 Class, Function, Code, Object 등을 추적하는 Diagram으로, System Diagram이나 Class Diagram과 같은 Object Design Model이다.
(e) UP 기반의 OOA&D에서는, 원(原) Iteration에서 개별 Use Case에 대해서 Sequence Diagram과 Class Diagram을 반복적(Iterative)이고 첨증적(Incremental)으로 작성한다.
(w) 대상 시스템이 State에 기반해서 일련되는 Events나 Messages에 따라 반응하는 동작이 있다면, State Machine Model은 그러면 Object Design에 도움이 된다.

① 1개

② 2개

③ 3개

④ 4개

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

SAMSUNG

○ 결제하려면 여기에 입력하십시오.

Student Enrollment Course Teacher Professor

답변자장

제출 및 평가종료

오늘 5:26
2021-04-07

https://outswExpertacademy.samsung.com/common/arch/test/archTest.do?isDirManager=false&serverName=localhost&locale=ko&backendAdmin=false&logMainfile=logo_company.png&canSeeSolveReference=N&isSystemAdmin=false&hasSDPAdminLinkAuth=false&formFileMaxUploadSize=10... | 검색...

평가 시간 정보

평가 시작 시간: 2021년 04월 07일 13:30
평가 종료 시간: 2021년 04월 07일 17:30
남은 시간: 00 : 03 : 32

응시 안내

1. 17:30에 평가가 종료되며,
고객에 답변하지 않은 '제출' 대상 평가가 평가 종료 후에는 평가로 표기됩니다.
"당안자장" 혹은 "제출" 등을 누르지 않으면 작성한 답변이 제출되지 않습니다.
※ 고객에게 평가가 종료될 경우, 마지막으로 "당안자장"한 답변이 제출됩니다.
2. 평가 시작 후에는 평가를 중단할 수 없습니다.
연락처: 최강원 (kangwon.choi@samsung.com)

```

classDiagram
    class Address {
        name: Text
        addSelect()
    }
    class ProductCatalog {
        getSpecification()
    }
    class ProductSpecification {
        description: Text
        price: Money
        itemID: ItemID
    }
    class Sale {
        date: Date
        itemCount: Boolean
        time: Time
        becomeComplete()
        makeLineItems()
        makePayment()
        getTotal()
    }
    class SalesLineItem {
        quantity: Integer
        getSubtotal()
    }
    class Payment {
        amount: Money
    }

    Address "1" -- "1" ProductCatalog : Looks-in
    ProductCatalog "1" -- "1" ProductSpecification : Contains
    ProductSpecification "*" -- "1" SalesLineItem : Describes
    SalesLineItem "*" -- "1" Sale : Captures
    Sale "*" -- "1" Payment : Paid-by
    Payment "*" -- "1" Sale : Paid-by
  
```

```

public class Sale
{
    private List<SalesLineItem> lineItems = new ArrayList<>();
    private Date date = new Date();
    private boolean isComplete = false;
    private Payment payment;

    public Money getBalance()
    {
        return payment.getAmount().minus( getTotal() );
    }

    public void becomeComplete() { isComplete = true; }

    public boolean isComplete() { return isComplete; }

    public void makeLineItem( ProductDescription desc, int quantity )
    {
        lineItems.add( new SalesLineItem( desc, quantity ) );
    }

    public Money getTotal()
    {
        Money total = new Money(); Money subtotal = null;

        for ( SalesLineItem lineItem : lineItems )
        {
            subtotal = lineItem.getSubtotal(); total.add( subtotal );
        }
        return total;
    }

    public void makePayment( Money cashTendered )
    {
        payment = new Payment( cashTendered );
    }
}
  
```

SAMSUNG

Parameter Visibility

검색하려면 여기에 입력하십시오.

답안자장 계출 및 평가종료

5:26 2021-04-07

The screenshot shows a Samsung S/W EXPERT ACADEMY course page. The top navigation bar includes links for 'Home', 'About Us', 'SAMSUNG', 'S/W EXPERT ACADEMY', 'S/W EXPERT ACADEMY', 'S/W EXPERT ACADEMY', and 'S/W EXPERT ACADEMY'. The main content area contains several sections:

- 평가 시간 정보**: Displays the evaluation time as 00 : 03 : 30.
- 평가 시작 시간**: Shows the start time as 2021년 04월 07일 13:30.
- 평가 종료 시간**: Shows the end time as 2021년 04월 07일 17:30.
- 남은 시간**: Shows the remaining time as 00 : 03 : 30.
- 응시 안내**: Includes instructions and contact information for the test.
- 평가 시험 내용**: Contains programming code and state machine diagrams.
- 평가 시험 내용**: Contains a sequence diagram.

평가 시험 내용 (Code Snippet):

```
        subtotal = lineItem.getSubtotal(); total.add( subtotal );
    }
    return total;
}

public void makePayment( Money cashTendered )
{
    payment = new Payment( cashTendered );
}
```

평가 시험 내용 (State Machine Diagram):

11. 다음은 State Machine Diagram입니다. start, e1, e2, e3, e5, e6, e12의 순서로 이벤트가 차례대로 발생되는 경우, 최종적으로 도달될 상태는 무엇인가요?

```
graph LR; start(( )) -- e1 --> S1[S1]; S1 -- e2 --> S2[S2]; S2 -- e3 --> S4[S4]; S4 -- e4 --> S4_3[S4.3]; S4 -- e6 --> S3[S3]; S3 -- e7 --> S5[S5]; S5 -- e8 --> S4_3; S4_3 -- e9 --> S1; S1 -- e5 --> S2; S2 -- e10 --> S3_1[S3.1]; S3_1 -- e11 --> S5_1[S5.1]; S5_1 -- e12 --> H[H*];
```

평가 시험 내용 (Sequence Diagram):

12. 다음은 학생이 학교 시스템에 로그인 후 등록을 하는 시나리오에 대한 Sequence Diagram입니다. Student는 '로그인(Login)'을 최초/최대 각각 몇 번까지 시도할 수 있나요?

```
sequenceDiagram
    actor Student
    actor AdminSystem
    Note over Student: loop(0,5)
    Student->>AdminSystem: login()
    activate AdminSystem
    AdminSystem-->>Student: incorrect PW
    deactivate AdminSystem
    Student->>AdminSystem: login()
    activate AdminSystem
    AdminSystem-->>Student: incorrect PW
    deactivate AdminSystem
    Student->>AdminSystem: login()
    activate AdminSystem
    AdminSystem-->>Student: incorrect PW
    deactivate AdminSystem
    Student->>AdminSystem: login()
    activate AdminSystem
    AdminSystem-->>Student: correct PW
    deactivate AdminSystem
    Student->>AdminSystem: register()
    AdminSystem-->>Student: registered
```

평가 시간 정보

평가 시작 시간: 2021년 04월 07일 13:30
평가 종료 시간: 2021년 04월 07일 17:30
남은 시간: 00 : 03 : 28

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 답변자정 혹은 '제출'
버튼을 누르면 평가가 종료됩니다.
"당안자정" 혹은 '제출' 버튼을 누르지
않으면 작성한 답변이 제출되지
않습니다.
- 시스템 평가에서 평가가 종료될
경우 마지막으로 "당안자정" 한
답안이 제출됩니다.
2. 평가 시작 후에는 평가를 중단할 수
없습니다.
연락처: 최강원
(kangwon.choi@samsung.com)

12. 다음은 학생이 학교 시스템에 로그인 후 등록을 하는 시나리오에 대한 Sequence Diagram입니다. Student는 '로그인(login)'을 최소/최대 각각 몇 번까지 시도할 수 있나요?

```

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

(화면길이 : 4000)
화면 : 1번
화면 : 0번

13. 다음의 Class Diagram의 Register 클래스에서 "Private"으로 정의되는 attribute는 모두 몇 개인가요?

Sale
+ time: DateTime

SAMSUNG

검색하려면 여기에 입력하십시오.

답안자정
제출 및 평가종료

오늘 5:26
2021-04-07

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 26

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당연자장 혹은 '제출'
화면을 선택합니다.
"당연자장" 혹은 "제출" 누르기
앞으로 작성한 답변이 제출되지
않습니다.
- 시스템은 평가가 종료될
경우 마지막으로 "당연자장" 한
답안이 제출됩니다.
2. 평가 시작 후에는 평가를 중단할 수
없습니다.
연락처 : 최강원
(kangwon.choi@samsung.com)

13. 다음의 Class Diagram의 Register 클래스에서 "Private"로 정의되는 attribute는 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 : location
    Sale "*" -- "1" Register : currentSale
  
```

(최대길이 : 4000)
4개

14. 다음은 교재의 POS 예제 중 "Process Sale" Use Case에 대해서, OOA 과정에서 고려한 SSD (System Sequence Diagram)와 Domain Model의 일부입니다. 이 패널들로부터 시스템의 행위와 구성을 추측하여, 다음의 질문에 맞자로 답하세요.

가정: 모두 5 상품 (카드 1개, 같은 빨간우유 2개, 같은 초코바 2개)을 구매합니다.

(1) 어떤 System Operation인가요?
 (2) 처음부터 endSale 메시지가 돌아오기 직전까지, 모든 몇 개의 Object가 새로 Instantiation 되었나요?
 (3) endSale 메시지가 돌아와서 처리된 직후에, 현재의 Domain Model 기준으로, 모두 몇 개의 Object가 보이나요?
 (4) makePayment(amount) 메시지가 처리되면, 총 몇 개의 Object가 삭제되나요? 단, 물건이 판매되면 그동안 새로 생성된 모든 경보는 삭제한다고 가정합니다.

```

sequenceDiagram
    participant Cashier
    participant System
    loop
        Cashier->>System: makeNewSale
        activate System
        System->>loop: enterItem(itemID, quantity)
        activate loop
        loop->>System: description, total
        deactivate loop
        System->>loop: endSale
        deactivate System
        loop-->>System: total with taxes
        System->>loop: makePayment(amount)
        deactivate System
        loop-->>System: change due, receipt
        deactivate loop
    end
  
```

SAMSUNG

검색하려면 여기에 입력하십시오.

닫기 저작 계출 및 평가종료

오늘 5:26
2021-04-07

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 23

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당한자에게 혹은 '제출'
버튼을 클릭합니다.
당한자'상' 혹은 '제출'을 누르기
않으면 작동한 당한이 계산되지
않습니다.
- 시스템은 평가가 종료될
경우 마지막으로 '당한자'상' 한
당한이 계산됩니다.

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

15. 다음은 고객의 POS 예제 중 'Process Sale' Use Case의 Main Success Scenario입니다. OOD에서 Fully-Dressed 표식으로 작성되었습니다. 7번 단계를 진행하는 중 POS 화면이 일정 시간 동안 멈춘 경우에 대한 Alternative Flows (Extensions)를 각각화하세요.

Main Success Scenario (or Basic Flow):

- Customer arrives at POS checkout with goods and/or services to purchase.
- Cashier starts a new sale.
- Cashier enters item identifier.
- System records sale line item and presents item description, price, and running total.
- Price calculation.
- Cashier waits steps 3-4 until indicator done.
- System presents total with taxes calculated.
- Cashier tells Customer the total, and asks for payment.
- Customer pays and System handles payment.
- System logs completed sale and sends sale and payment information to the external Accounting system (for accounting and commissions) and Inventory system (to update inventory).
- System presents receipt.
- Customer leaves with receipt and goods (if any).

(상세설명 : 4000)
 ① 4
 ② 4
 ③ 8
 ④ 4

16. 다음은 일반적인 UP Development Case를 설명하는 그림입니다. SW Architecture Document(SW Architecture Document) Construction Phase에서 민관으로 되어 있는데, 그 민관의 의미와 민관이 뭔 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	...				

답변자장
제출 및 평가종료

검색하려면 여기에 입력하십시오.

오늘 5:26
2021-04-07

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 22

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 답변자에게 혹은 '제출'
버튼을 클릭합니다.
"당안자장" 혹은 "제출"을 누르지
않으면 작성한 답안이 제출되지
않습니다.
- 시스템은 평가가 종료될
경우, 마지막으로 "당안자장" 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon_choi@samsung.com)

17. 다음의 Sequence Diagram을 보고, System Operation "systemOp1()"의 body를 수도코드 수준에서 최대한 자세히 작성하세요.

(화면길이 : 4000)
SW Architecture Document는 Construction Phase에서 제출 사용이 되지만, 수정되지 않는다.
Elaboration phase에서 요구사항이 결정되고 수정이 되면 인식기 때문입니다.

18. (문항12)의 Sequence Diagram은 작성자의 의도와는 다르게, 로그인에 실패하더라도 등록을 할 수 있게 모델링 되어 있습니다. 로그인에 성공한 학생만 등록을 할 수 있고, 최대 7번까지 로그인을 시도할 수 있도록, Sequence Diagram을 수정하세요.

32678_강상우_OOAD 기본_설계형18.png

19. (문항18)의 수정된 Sequence Diagram을 Activity Diagram으로 모델링 하세요.

32678_강상우_OOAD 기본_설계형19.png

20. 다음은 University Information System (UIS)에 대한 요구사항입니다. 이를 잘 반영하는 Class Diagram을 작성하세요. 단, Generalization (Inheritance)을 제 번 이상 사용하지마요.

A. 학교(University)는 여러 이론 단과대학(School)으로 구성되어 있습니다.
B. 각 단과대학에는 1개 기관, 최대 10개의 학과(Department)로 구성되어 있습니다.
C. 학교는 직원(Staff)과 교수(Professor)를 고용하며, 모두 Name, ID, email 정보를 가진다.
D. 각 단과대학은 1명의 학장교수(Dean), 각 학과는 1명의 주임교수(Head)가 배정되어 있다.
E. 모든 직원은 학교에 등록되어 있으며, 각 직원은 그에 맞는 계약을 맺을 수 있다.
F. 모든 교수는 최소 1개, 최대 3 과목(Course Class)을 강의한다.
G. 과목은 단일으로 개설할 수도 있고, 여러 개의 분반(Class)으로 구성될 수도 있다.
H. 과목은 과목장, ID, 강의실, 강의시간, 개설여부 등을 가진다.

32678_강상우_OOAD 기본_설계형20.png

https://out.swxp.academy.samsung.com/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainFileName=logo_company.png&canSeeSolverReference=N&isSystemAdmin=false&hasSDPAdminLinkAuth=false&formFileMaxUploadSize=10x

17. 다음은 Sequence Diagram을 보고, System Operation "systemOP()"의 body를 주도코드 주준에서 최대한 자세히 작성하세요.

```
sequenceDiagram
    actor Student
    participant directA as directA : Class
    participant directB as directB : Class
    participant directC as directC : Class
    Note over directA: systemOP(a,b)
    Student->>directA: systemOP(a,b)
    activate directA
    directA->>directB: a(s)
    directA->>directC: a(s)
    directB-->>directA: b(s)
    directC-->>directA: b(s)
    deactivate directA
    directA-->>Student: aa -> bb
```

(최대길이 : 4000)
ELSEIF a < b:
call class C method b with parameter b
GET bb
return as aa -> bb

18. (문항12)의 Sequence Diagram은 작성자의 의도와는 다르게, 로그인에 실패하더라도 통로를 할 수 있게 묘사링 되어 있습니다. 로그인에 성공한 확장만 통로를 할 수 있고, 최대 7번까지 로그인을 시도할 수 있도록, Sequence Diagram을 수정하세요.

32678_강상우_OOAD 기본_설계형18.png

19. (문항18)의 수정된 Sequence Diagram을 Activity Diagram으로 모델링 하세요.

32678_강상우_OOAD 기본_설계형19.png

20. 다음은 University Information System (UIS)에 대한 요구사항입니다. 이를 살 반영하는 Class Diagram을 작성하세요. 단, Generalization (Inheritance)을 제외 번 이상 사용하세요.

A. 학생(Student)은 하나 이상의 학교에(School)으로 구성되어 있다.
B. 학생은 학교에 등록되어 있다. 그 학교에 따라 학과(Department)로 구성되어 있다.
C. 학교는 직원(Staff)과 교수(Professor)를 고용하며, 모두 Name, ID, email 정보를 가진다.
D. 각 단과대학은 1명의 원장교수(Dean), 각 학과는 1명의 주임교수(Head)가 배정되어 있다.
E. 모든 학과는 최소 1개, 최대 5 과목(Course Class)을 개설할 수 있다.
F. 모든 교수는 최소 1개, 최대 3 과목(Course Class)을 강의한다.
G. 과목은 단독으로 개설될 수도 있고, 여러 개의 복수(Class)으로 구성될 수도 있다.
H. 과목은 단독으로 개설될 수도 있고, 여러 개의 복수(Class)으로 구성될 수도 있다.

32678_강상우_OOAD 기본_설계형20.png

Design Pattern 기반

1. 패턴에 대한 설명으로 훈은 것을 모두 고르면?

(a) 일반적으로 소프트웨어 설계 시 이어백체 패턴이 먼저 적용되고, 이후 설계 패턴이 적용된다.
(b) MVC 패턴은 설계 패턴이 아니라 그냥 패턴으로 분류된다.
(c) 개발을 위한 프레임워크들이 예를 들어 메타일 패턴을 사용하거나 폴포수 모델을 사용하는 것과 구관하게 GoF 패턴을 설계에 적용할 수 있다.
(d) (a), (b)

SAMSUNG

검색하여 여기에 의견하시시오

Design Pattern 기반

Design Pattern 기반

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:00

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 11

응시 안내

1. 폐현에 대한 설명으로 좋은 것을 모두 고르면?
 (a) 일관적으로 소프트웨어 설계 시 아키텍처 패턴이 먼저 적용되고, 이후 설계 패턴이 적용된다.
 (b) MVC 패턴은 설계 패턴이 아니라 디자인 패턴으로 분류된다.
 (c) 개별 프로세스에 관계없이 이를 놓고 애자일 방식으로 구성을 할 수 있다.

(a), (b)

(b), (c)

(a), (c)

2. 다음 자료(guideLine) 중 좋은 설계를 이용하여는 문제의 것들은?
 (a) 각 페리기마다 별도의 Context 클래스를 두고 해당 페리기에서 있는 모든 다른 클래스의 래퍼리를 계약하는 역할을 맡기는 것이 좋다.
 (b) 각 페리기마다 별도의 Context 클래스를 두고 해당 페리기에서 있는 모든 다른 페리기의 래퍼리를 계약하는 역할을 맡기는 것이 좋다.
 (c) 이에 좋지 않을 때 미리 모드의 소자를 넣어는 것이 바람직하며 이를 위해 미드드의 일부 파라미터 값에 따로 여러 상수판(heterogeneous)한 역할을 수행하도록 만드는 것이 추천된다.

0개

1개

2개

3개

3. 설계 원칙(SDP)에 대한 설명으로 좋은 것을 모두 고르면?
 (a) 하나의 클래스는 하나의 설계 패턴을 적용하는 것이 좋다는 원칙이다.
 (b) 해당 설계 원칙을 잘 지키면 클래스의 융합도(cohesion)가 높아지게 된다.

(a)

(b)

(a), (b)

4. 다음 클래스 다이어그램에 대해 올바른 설명을 모두 고르면?
 (선택지)(1))
 (a) runStrategy() 메소드의 구현 코드에서 ConcreteStrategy 클래스의 이름이 직접적으로 등장해서는 안된다.
 (b) strategy 필드는 Strategy 클래스에서 정의되지 않고 Context 클래스에서 정의된다.
 (c) Context 클래스와 Strategy 클래스는 연결하는 관계를 shared aggregation(복수 관계)로 정의하는 경우 해당 설계는 전략 패턴이 아니라 기교(bridge) 패턴을 의미하게 된다.

```

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

    Context "1" -- "0..1" Strategy : strategy
    Strategy "*" -- "3" ConcreteStrategyA .. ConcreteStrategyC : execute()
  
```

Design Pattern 기본

32678_강상우_OOAD 기본_설계행20.png

원래 첨부되어 있는 파일 : 32678_강상우_OOAD 기본_설계행20.png

찾아보기...

답변자장 계출 및 평가종료

https://out.sweptacademy.samsung.com/common/arch/test/archTest.do?isDirManager=false&serverName=localHost&locale=ko&backedAdmin=false&logMainfileName=logo_company.png&canSeeSolveReference=N&isSystemAdmin=false&hasSDAdminLinkAuth=false&formFileMaxUploadSize=10... 검색...

https://out.sweexpertacademy.samsung.com/common/arch/test/archTest.do?isBrManager=false&serverName=localhost&locale=ko&backendAdmin=false&logMainfileName=logo_company.png&canSeeSolveReference=N&isSystemAdmin=false&hasSDPAdminLinkAuth=false&formFileMaxUploadSize=10- 검색...

S/W EXPERT ACADEMY

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 05

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당선자에게 혹은 '제출'
라고 표기되는 평가가 됩니다.
"당선자" 혹은 "제출"을 누르지
않으면 작성한 답변이 제출되지
않습니다.
- 시스템 평가에서 평가가 종료될
경우, 마지막으로 "당선자"를 할
당수로 평가됩니다.
2. 평가가 종료 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon_choi@samsung.com)

① (a), (b)
② (a), (b), (c)
③ 단일

4. 다음 클래스 다이어그램에 대해 출바쁜 설명을 모두 고르면?
(여러개)(1))

(a) runStrategy() 메소드의 구현 코드에서 ConcreteStrategyA 클래스의 이름이 직접적으로 등장해서는 안된다.
(b) Strategy 뾰노는 Strategy 클래스에서 정의되지 않고 Context 클래스에서 정의된다.
(c) Context 클래스와 Strategy 클래스와 연결하는 양계를 shared aggregation에서 composition으로 바꾸면 해당 설계는 전역 패턴이 아니라 가교(bridge) 패턴을 의미하게 된다.

```

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

    Context "1" -- "1" Strategy : strategy
    Context "1" -- "1" ConcreteStrategyA : strategy
    Context "1" -- "1" ConcreteStrategyB : strategy
    Context "1" -- "1" ConcreteStrategyC : strategy
    Strategy "*" -- "3" ConcreteStrategyA .. ConcreteStrategyC : execute()
  
```

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

5. 객체 합성(object composition)과 위임(delegation)을 통해 동작하는 패턴을 모두 고르면?
(여러개)(1))

(a) 팩토리 패턴 (factory method) 패턴
(b) 추상 패턴 (abstract class) 패턴
(c) 가교 (bridge) 패턴

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

6. 다음 자바 언어로 표현된 코드는 버튼이 클릭될 경우 클릭 이벤트를 처리하는 팬들러의 등록 과정을 보이고 있다. 이 때 사용된 패턴 이름으로 가장 적합한 것은?
(1))

```

Button aButton = new JButton("OK");
ButtonHandler handler = ButtonHandler();
aButton.addActionListener(handler);

```

① 상태 (state) 패턴
② 가교 (bridge) 패턴
③ 장식자 (decorator) 패턴
④ 감시자 (observer) 패턴

7. MVC 패턴의 대체 구조를 설명해 모두 고르면?
(여러개)(1))

(a) Model은 Controller에게 API를 호출하여 사용자의 퍼스트(seriou)를 해석하도록 지시할 수 있다.
(b) Controller는 View의 API를 알고 있으며 해당 API를 직접 호출할 수 있다.
(c) UI/User Interface를 GUI 방식에서 콘솔(Console) 방식으로 분기하는 경우 Model 모듈에 대해서는 코드의 변화가 거의 없다.

① (a), (b)
② (a), (c)

답안저장
제출 및 평가종료

2021-04-07 오전 5:26

The screenshot shows a web-based exam interface for Samsung S/UXpert Academy. The top navigation bar includes the URL https://out.suxpertacademy.sams.com/common/arch/test/archTest.do?isRManager=false&serverName=localhost&locale=ko&backendAdmin=false&logoMainFileName=logo_company.png&canSeeSolveReference=N&isSystemAdmin=false&hasSDPAdminLinkAuth=false&formFileMaxUploadSize=10x, a search bar, and a user profile icon.

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 03

응시 안내

1. 17:30에 평가가 종료되며,
평가는 평가 시작과 같은 예술
과 조각가 비단입니다.
- '당인자장' 혹은 '제술'을 누르지
않으면 각각 당인자와 예술
당인자장을 선택합니다.
- 시리이 경쟁하여 평가가 종료될
경우, 마지막으로 '당인자장' 한
명만 평가를 받습니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처: 최강원
(kangwon.cho@samsung.com)

문제

7. MVC 패턴에 대해 올바른 설명을 모두 고르면?

(a) Model은 Controller의 API를 호출하여 사용자의 폼(gesture)를 해석하도록 자사할 수 있다.
(b) Controller는 View의 API를 알고 있으며 해당 API를 직접 호출할 수 있다.
(c) UI(User Interface)는 GUI 환경에서 사용자에게 명령하는 경우 Model 모듈에 대해서는 코드의 변화가 거의 없다.

(d) (a), (b)
(e) (a), (c)
(f) (b), (c)
(g) (a), (b), (c)

8. 중개자(Mediator) 패턴에 대해 올바른 설명을 모두 고르면?

(a) 중개자 패턴은 적용된 프로그램은 감시자(observer) 패턴이 적용된 것에 비해 코드의 재사용성이 높다.
(b) 중개자 패턴의 Mediator는 일반적으로 각 colleague 객체의 클래스 타입을 알고 있어 이를 통해 다른同事者를 관리합니다.
(c) 중개자 패턴은 갖는 동료(colleague) 객체를 기억해 놓기 위해 각각의 메시지를 모아서 알고 올바른 관례를 통해 통신한다.

(d) (a), (b)
(e) (a), (c)
(f) (b), (c)
(g) (a), (b), (c)

9. 반복자(Iterator) 패턴과 집합.aggregate() 인터페이스에 대한 설명으로 올바른 것을 모두 고르면?

(a) 집합.팩토리 어레이 요소(element) 인터페이스를 포함하기 전에 인터페이스는 실행자(implement) 패턴을 적용할 수 있다.
(b) 반복자 패턴은 Context 클래스 입장에서 보면 반복자 인터페이스는 따로 있는 인터페이스이다.
(c) 반복자 패턴은 SRP 설계원칙을 잘 지키고 있다.

(d) (a)
(e) (b)
(f) (a), (b)
(g) (b), (c)

10. 다음 절개 원칙 DIP에 대한 설명 중 옳은 문장의 개수는?

(a) 추상 팩토리(abstract factory) 패턴에 적용됨을 계약원칙이다.
(b) 저수준의 모듈이 고수준의 모듈에 의존관계를 가지는 안된다.
(c) 인터페이스 전달 필수는 해당 인터페이스를 구현하는 클래스와 동일한 패키지에 배치되어야 한다.

(d) 0 개
(e) 1 개
(f) 2 개
(g) 3 개

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

(이미지 11)

답안 저장 **제출 및 평가 종료**

평가 시간 정부

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 03 : 02

응시 안내

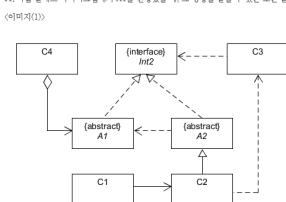
1. 17:30에 평가가 종료되며,
그 경에 당연하게 혹은 '제출'
버튼을 클릭합니다.
당연하게 혹은 '제출'을 누르지
않으면 작동한 딜레이가 계산되지
않습니다.
이후 평가에서 평가가 종료될
경우 마지막으로 '당연하게' 한
답안이 제출됩니다.

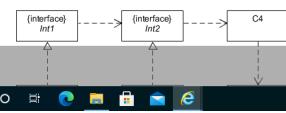
2. 평가가 종료 후에는 평가를 중단할 수
있습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

10. 다음 설계 패턴 DIP에 대한 설명 중 옳은 문장의 것수는?
 (a), (b)
 (b), (c)
 (c), (d)

(a) 추상 패트(abstract factory) 패턴에 적용된 설계 패턴이다.
(b) 자주쓰는 모듈이 고수준의 모듈에 의존성을 가지는 현상을.
(c) 인터페이스 선언 외에는 해당 인터페이스를 구현하는 클래스와 동일한 패키지에 위치되어야 한다.

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

 (예제값) : 4000
 C4, A1, C2, C1

12. 다음 클래스 다이어그램을 바라 혹은 C++로 구현했을 때, 예의 말이 풍자 되는 코드를 (a)~(l) 중에 모두 고르시오.

 (예제값) : 1000
 (a) A1 a = new C2(); // in Java
 A1 * a = new C2(); // in C++
 (b) A1 b = new C3(); // in Java
 A1 * b = new C3(); // 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) C2 e = new C3(); // in Java
 C2 * e = new C3(); // in C++
 (f) Int1 f = new C1(); // in Java
 Int1 * f = new C1(); // 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 C3(); // in Java
 Int2 * i = new C3(); // in C++

답안자료
제출 및 평가종료

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 02 : 59

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당한자에게 혹은 '제출'
버튼을 클릭합니다.
"당한자" 혹은 "제출"을 누르지
않으면 작동한 당한이 계승되며
있으나,
-이전 평가에서 평가가 종료된
경우, 마지막으로 '당한자'한
당한이 계승됩니다.
2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

12. 다음 클래스 다이어그램을 바탕 혹은 C++로 구현했을 때, 예의 양이 풍자 되는 코드를 (a)~(l) 중에 모두 고르시오.
(선택지)(1)

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

```

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

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

    C1 --> Int1
    C2 --> Int1
    C3 --> Int2
    C4 --> Int2
    C5 --> Int2

    abstract class A1 {
        <<Abstract>>
    }

    C1 --> A1
    C2 --> A1
    C4 --> A1
    C5 --> A1
  
```

(최대길이 : 4000)
b, f, i

13. 다음 설명에 해당하는 GoF 패턴 이름을 쓰시오.
"서로 다른 표현 방법을 가진 복잡한(complex) 객체를 생성하고자 한다. 이때 통일한 생성 과정을 사용할 수 있도록 생성 방법과 표현 방법을 분리해준다."

(최대길이 : 4000)
Template Method Pattern

14. 다음 GRASP 원칙 중 Pure Fabrication의 대안 설명이다. 다음 빈칸 (a)와 (b)에 알맞는 풀어쓸 각각 쓰시오.
'(a) 원칙이 개시하는 책임은 적합하지 않을 때, 높은 유팅도(High Cohesion)과 낮은 결합도(Low Coupling) 원칙을 달성하도록 하면서 개사용상을 간단하여 (b) 모델에 동장하기 않는 가공의 클래스에 책임(responsibility)을 부여한다.'

(최대길이 : 4000)
(a) Information Expert
(b) 도제인

15. 다음 코드에 적용된 패턴 이름을 GoF 패턴 23가지 중에 하나를 풀어서 써고 그 이유를 설명하시오.

```

public class Client {
    public static void main (String args []) {
        ServiceProvider sp = new ServiceProvider ();
        sp.startService ();
    }
}

abstract class ServiceProvider {
    Server s = null;
    public void startService () {
        // ...
    }
}
  
```

답안자료
제출 및 평가종료

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 02 : 57

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당연처럼 혹은 '제출'
라고 하는 단어를 사용합니다.
"당연처럼" 혹은 "제출"을 누르기
않으면 작성한 답변이 제출되지
않습니다.
- 예전에는 평가가 종료될
경우 마지막으로 "당연처럼" 한
답안이 제출됩니다.

2. 평가 시작 후에는 평가를 중단할 수
없습니다.

연락처 : 최강원
(kangwon_choi@samsung.com)

13. 다음 설계 패턴 이름을 쓰시오.
"서로 다른 패턴 방법을 가진 복잡화(complex) 객체를 생성하고자 한다. 이때 동일한 생성 과정을 사용할 수 있도록 생성 방법과 표현 방법을 분리해준다."
(최대길이 : 4000)
Template Method Pattern

14. 다음은 GRASP 설계 원칙 중 Pure Fabrication의 대체 설명이다. 다음 빙간 (a)와 (b)에 알맞는 풀어를 각각 쓰시오.
"(a) 워킹이 계시하는 해결책은 적합하지 않을 때, 높은 유통도(High Cohesion)와 낮은 결합도(Low Coupling) 원칙을 달성하도록 하면서 개사용상을 간단하여 (b) 모델에 등장하기 않는 가공의 클래스에 책임(responsibility)을 부여한다."
(최대길이 : 4000)
(a) Information Expert
(b) 도구인

15. 다음 코드에 적용되는 패턴 이름을 GoF 패턴 23가지 중에 하나를 불러서 써고 그 이유를 설명하시오.

```
public class Client {
    public static void main(String args[]) {
        ServiceProvider sp = new ServiceAProvider();
        sp.startService();
    }
}

abstract class ServiceProvider {
    Server s = null;
    protected Server getServer() {
        public void startService() {
            s = getServer();
            writeLog();
        }
        protected void writeLog() {
            // generate log message here
        }
    }
}

class ServiceAProvider extends ServiceProvider {
    protected Server getServer() { return new ServerA(); }
}

class ServiceBProvider extends ServiceProvider {
    protected Server getServer() { return new ServerB(); }
}

interface Server {
    class ServerA implements Server {
        // implementing Server interface here
    }
    class ServerB implements Server {
        // implementing Server interface here
    }
}
```

(최대길이 : 4000)
Bridge pattern.
ServiceProvider Server의 인터페이스를 통해 결합하고 일정한 operation을 통해 접근한다.
이는 M에 지파면의 관통부인 특성으로 2개의 관심사가 다른 부분을 분리하고 서로의 영향을 최소화하는 패턴 방식이다.

16. 아래 코드에 하이라이트 구성을하는 `java.sql.Date` 클래스는 상위 클래스인 `java.util.Date` 클래스로부터 상속받고 있다. 구현을 살펴보면 `java.sql.Date` 클래스의 `getHours()`, `getMinutes()`, `getSecond()` 메소드는 항상 `IllegalArgumentException`을 발생시키도록 오버라이드 된 것을 볼 수 있다. 본 설계의 문제점과 가장 연관성이 있는 M에 지파면의 관통부인 특성으로 2개의 관심사가 다른 부분을 분리하고 서로의 영향을 최소화하는 패턴 방식이다.
<이미지(1)>

`java.util.Date`

- * int getDate()
- * int getDay()
- * int getMonth()
- * int getMinutes()
- * int getSeconds()
- // other methods

닫기자장 계출 및 평가종료

https://out.sweptpartacademy.samsung.com/common/arch/test/archTest.do?isBirManager=false&serverName=localhost&locale=ko@backendAdmin=false&logMainfileName=logo_company.png&canSeeSolveReference=N&isSystemAdmin=false&hasSDAdminLinkAuth=false&formFileMaxUploadSize=10- 검색...

평가 시간 정보

평가 시작 시간
2021년 04월 07일 13:30

평가 종료 시간
2021년 04월 07일 17:30

남은 시간
00 : 02 : 55

응시 안내

1. 17:30에 평가가 종료되며,
고 경에 당연지어 혹은 '제출'
라고 하는 단어를 사용합니다.
"당연지성 혹은 '제출'을 누르기
앞으로 작성한 답변이 제출되지
않습니다."
나는 평가에 평가가 종료될
경우 마지막으로 "당연지성" 한
단어로 평가를 제출됩니다.

2. 평가가 종료되는 평가를 중단할 수
있습니다.

연락처 : 최강원
(kangwon.choi@samsung.com)

16. 아래 코드가 발생하는 예외는 ?
java.sql.Date 클래스는 상위 클래스인 java.util.Date 클래스로부터 상속받고 있다. 구현을 살펴보면 java.sql.Date 클래스의 getHours(), getMinutes(), getSecond () 메소드는 항상 IllegalArgumentException을 발생시키도록 오버라이드 된 것을 볼 수 있다. 본 문제의 문제점의 가장 큰原因是 `getSeconds()`

```
java.util.Date
+ int getDate()
+ int getDay()
+ int getHour()
+ int getMinute()
+ int getSecond()
.. // other methods
```

```
java.sql.Date
+ int getDate()
+ int getDay()
+ int getHour() // trigger exception
+ int getMinute() // trigger exception
+ int getSecond() // trigger exception
.. // other methods
```

(최대길이 : 4000)
Liskov Substitution Principle 위반.
복잡의 Operation 을 3가지 exception을 발생 시키므로, 만약 Date를 상속한 다른 클래스와 전혀 다른 동작을 해 예측 불가능성을 높인다.
즉, 문법적으로 문제가 없으나, 프로그래머가 솔바른 문법으로 문장을 실행 시켜도 예외가 발생해 혼란을 끌인다.

17. 다음 코드는 무슨 일을 하는지를 설명하고, 감시자(Observer) 패턴에서 결합도가 낮게 설계된 부분을 레인의 코드를 통해 구체적으로 지적하시오.

```
(최대길이 : 4000)  
결합도를 서로 다른 두 모듈이 상호간에 의존적인 것을 말한다. 예를 들어, A, B클래스가 각각 변형될 때 이 변경될 때 이 변경이 서로의 클래스와 영향을 주면 이를 Coupling되었다고 할 수 있다.  
class A extends Observable {  
    public void somethingChanged() {  
        setChanged();  
    }  
}  
class X {}
```

18. 다음 코드에 나타난 클래스 A를 상관된 패턴으로 만들기 위한 구현을 최소한의 수정을 통해 계시하시오. 단, 멀티쓰레드(multi-thread) 사용을 고려할 필요는 없으며 자바 혹은 C++ 언어를 중 하나로 표현하시오.

```
in Java  
class A {  
    private int a1;  
    public int a2;  
    public void set_a1(int a) { this.a1 = a; }  
    public A(int b1, int b2, X b3) {  
        this.b1 = b1;  
        this.a2 = b2;  
        this.b3 = b3;  
    }  
}  
class X {}  
  
// in C++  
class A {  
private: int a1;  
public:  
    int a2;  
    X* a3;  
    void set_a1(int a) { this.a1 = a; }  
    A(int b1, int b2, X* b3) {  
        this.a1 = b1;  
        this.a2 = b2;  
        this.a3 = b3;  
    }  
};  
class X {}
```

32678_강상우_Design Pattern 기본_설계형18.txt [찾아보기...]
현재 첨부되어 있는 파일 : 32678_강상우_Design Pattern 기본_설계형18.txt

답변작성
제출 및 평가종료

검색하려면 여기에 입력하십시오. 0 48 2021-04-07 오전 5:27