Object-Oriented Analysis and Design (2013년)

성명:_____

※ 총 4 페이지, 제한 시간 60분.
※ Open book. Closed neighbor.
※ 문제지에 직접 답안을 작성한 후 제출하시오.

Part I. *(16점)* **True/False**

1. *(4점)* 클래스 B가 클래스 A의 서브클래스로 선언된 경우, 다음 각 문장에 대해 true/false로 답하시오.

```
// B is a subclass of A
class B extends A { ... }
```

① Every object of type **B** is also an object of type **A**, but *not vice-versa*. ___*True*___
② Anywhere an object of type **A** can be used, an object of type **B** can be used as well, but *not vice-versa*. ___*True*_____
③ Anything that is true of an object of type **B** is also true of an object **A**. ___*False*_____
④ **A** represents a more general concept than **B**, and **B** represents a more specialized concept than **A**. ___*True*_____

2. *(2점)* Waterfall Model is an iterative development process based on the assumptions that requirements are accurate and stable. ___*False*_____
3. *(2점)* UML is a visual modeling language, but not a development process. ___*True*___
4. *(2점)* In UP, implementation and testing disciplines are performed only at Construction phase. ___*False*_____
5. *(2점)* In Design by Contract, preconditions denote the responsibility of callers, while postconditions represent the responsibility of the developer. ___*True*___
6. *(2점)* In OO paradigm, reusability is best achieved or worthwhile at the object-level. ___*False*_____
7. *(2점)* All the vocabularies used in Use Case Model and Domain Model must come from problem domains. ___*True*___

Part II. *(30점)* **객관식 문제**

1. *(2점)* 다음중 객체와 속성(attribute)의 관점에서 볼 때, 열거된 다른 것과 가장 관계가 먼 것은?

① Address ② Age
③ Sex ④ **Student**

2. *(2점)* 다음 중 객체(object)의 정의와 거리가 먼 것은?

① Behavior ② **Name** ③ State ④ Identity

3. *(2점)* 다음 중 separation of interface from implementation과 관계 있는 것은?

① Abstraction ② Instantiation
③ Generalization ④ **Encapsulation**

4. *(2점)* 다음 중 개발프로세스(development process)가 정의하고 있는 것과 가장 거리가 먼 것은?

① Who ② When ③ **Where** ④ What ⑤ How

5. *(2점)* 다음 중 UP의 best practices와 관계 없는 것은?

① **Tackle low-risk issues in early iterations.**
② Build a cohesive, core architecture in early iterations.
③ Adopt iterative and incremental development approach.
④ Continuously engage users for evaluation and feedback.

6. *(2점)* 다음 중 소프트웨어 개발 프로세스에서 설계(design)의 목적을 **가장 잘** 설명하고 있는 것은?

① The system design must be documented so that the clients will understand how to interpret the program code. This will help them to test the system once it is delivered.

② **The development of a logical model of the system that is consistent with the conceptual model and satisfies the system requirements.**

③ It prevents the reinvention of code that already exists in other applications and could be easily reused in the current application.

④ The assignment of responsibility of program modules to programmers. Each programmer should be given responsibility for a module that best matches his or her skills and experience.

7. *(2점)* Unified Process (UP)의 각 phase를 순서대로 올바르게 나열한 것은?

① Elaboration-Inception-Construction-Transition

② Transition-Inception-Elaboration-Construction

③ Construction-Elaboration-Transition-Inception

④ **Inception-Elaboration-Construction-Transition**

8. *(2점)* UML의 interaction diagram에 속하지 않는 것은?

① Sequence diagram

② Timing diagram

③ Communication diagram

④ **Activity diagram**

9. *(4점)* 다음은 UP Phase의 목적을 나타내는 목록이다.

① Feasibility Study

② Beta Test, Tuning

③ Baseline architecture

④ Implementation of Lower risks

위의 목록 가운데 아래 각 Phase의 목적과 가장 밀접한 관련이 있는 것을 고르시오.

가. Construction .....(**4**)

나. Inception .....(**1**)

다. Transition .....(**2**)

라. Elaboration .....(**3**)

10. *(4점)* 다음은 요구사항(requirements)을 기록하는 artifact(s)의 목록이다.

① Vision Statement ② Use case Model

③ Glossary ④ Supplementary Spec.

위의 목록 가운데 아래 항목과 가장 밀접한 관련이 있는 것을 고르시오.

가. Data dictionary .....(③)

나. To summarize high-level requirements..(①)

다. Functional requirements .....(②)

라. Quality attributes (or non-functional requirements) .....(④)

11. *(6점)* 다음은 Use Case에서 actor의 종류에 대한 설명(description)과 그들에 대한 파악이 필요한 이유(reason)를 보여준다.

Description:

① Has an interest in the behavior of the use case without direct interaction with the SUD.

② Has user goals fulfilled through using services of the SuD.

③ Provides a service to the SuD

Reason:

ⓐ To clarify external interfaces and protocols

ⓑ To ensure that all the necessary interests are captured and satisfied

ⓒ To find user goals, which drive the use cases

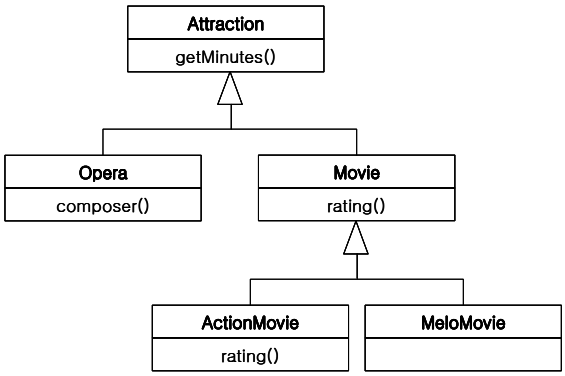
위의 목록 가운데 아래 각 Actor와 가장 밀접한 관련이 있는 것을 **Description**과 **Reason**에서 각각 하나씩 고르시오.

가. Primary Actor .....(② , ⓒ)

나. Supporting Actor .....(③ , ⓐ)

다. Offstage Actor .....(① , ⓑ)

Part III. *(28점)* 다음의 Class diagram을 보고 물음에 답하시오. 단, 모든 operation은 public method이며 dynamic binding이 적용된다고 가정한다.



아래 Demo 클래스 main 메소드의 각 문장에 대해, 합법적인 문장이면 'Yes'로 표시하고, 그렇지 않으면 'No'로 표시하시오. 단, 합법적인 메소드 호출 문장의 경우 호출되는 메소드가 어느 클래스에서 정의된 것인지 보이시오. 예를 들어, Move 클래스의 rating()이 호출되면 Movie::rating()으로 표시하시오.

```
class Demo {
 public static void main(String[] args[]) {
  ActionMovie ref_am
      = new ActionMovie();  (1)___Yes_____
  ref_am.rating();              (2) ActionMovie::rating
  ref_am.getMinutes();      (3) Attraction::getMinutes()
  Movie ref_m = new Movie();
  ref_m.rating();               (4) Movie::rating
  ref_m = new ActionMovie(); (5)___Yes_____
  ref_m.rating();               (6) ActionMovie::rating
  ref_m.getMinutes();       (7) Attraction::getMinutes()
  ref_m.composer();             (8)___No_____
  Attraction ref_a = foo();   (9)___Yes_____
  ref_a.getMinutes();       (10)Attraction::getMinutes()
  ref_a.rating();               (11)___No_____
  ref_m = bar();                (12)___Yes_____
  ref_m.rating();               (13) Movie::rating
  Movie ref_m2 = foo();     (14)___No_____
 }

 public static Attraction foo()
        { return new ActionMovie(); }
 public static Movie bar()
        { return new MeloMovie(); }
}
```

Part IV. *(6점)* GRASP pattern에 관한 문제이다. 기술된 문제를 해결하는 데 있어 가장 적절한 pattern이 무엇인지 pattern name을 보이시오.

1. You want to assign responsibility for handling a system event, but don''t want to have the user interface be tightly coupled to classes in the application logic layer. What pattern would best support this?

Ans.) Controller Pattern

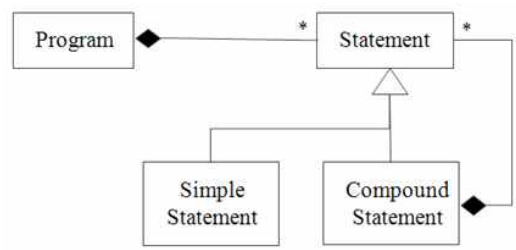
2. Suppose you want to decide which of two classes should be responsible computing something, and one of the two classes involved has an attribute that is needed in the computation, but the other class does not. What GRASP design pattern would best help you decide what class should be responsible?

Ans.) (Information) Expert Pattern

Part V. *(6점)* 다음에 주어진 설명을 잘 읽고, 언급된 conceptual class들 사이의 각종 관계(association, aggregation, composition aggregation, generalization/specialization), role names, multiplicity 등을 가능한 한 상세히 보여주는 UML class diagram을 그리시오.

- 하나의 프로그램(Program)은 여러 개의 문장(Statement)으로 구성된다.
- 하나의 문장은 단순 문장(Simple Statement)이거나 복합 문장(Compound Statement)이다.
- 하나의 복합문장은 여러 개의 문장으로 구성된다.

Ans.)

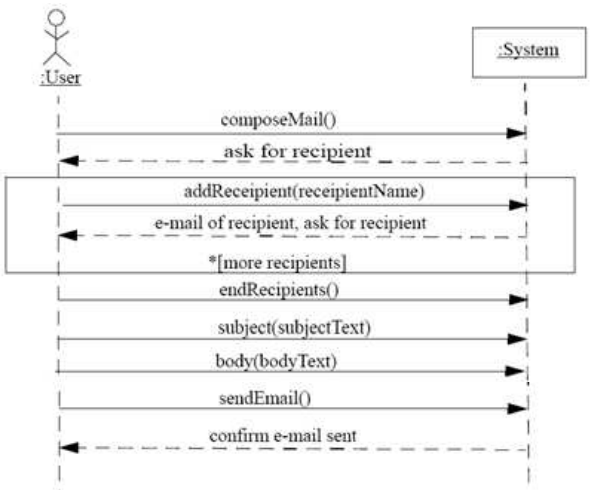


Part VI. *(6점)* Main Success Scenario로부터 도출되는 SSD (System Sequence Diagram)을 보이시오.

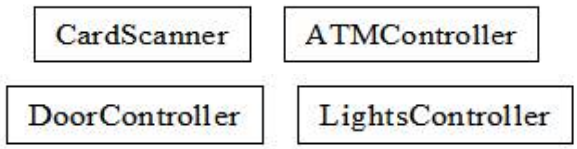
UC1: Send Email
Primary Actor: User
Main Success Scenario (or Basic Flow):

1. The User indicates the System he wants to send an e-mail message.
2. The System asks the user for the name of a recipient.
3. The User tells the system the name of a recipient.
4. The System confirms that it knows the e-mail address of this recipient by showing their address, and remembers the recipient.
   System repeats steps 2-4 until User indicates done.
5. The User tells the system the subject of the e-mail.
6. The System remembers the subject.
7. The User tells the System the body (i.e., the text) of the e-mail message.
8. The System remembers the body of the e-mail.
9. The User tells the System to send the e-mail message.
10. The System sends the message to the recipients, confirming to the User that the message was sent.

Ans.)



Part VI. *(8점)* Draw a simple UML class to solve a problem. You may need to add new classes to the ones given in the problem. Our ATM (Automatic Teller Machine) needs to have a card reader to allow the customer to scan their ATM card for validation. To abstract the inner workings of the scanner, we've designed a CardScanner class which responds when a card is swiped and then informs the ATMController object. We don't want to have the CardScanner depend on the ATMController interface, however, and we'd like to use the CardScanner for other applications. For example, some ATM machines in high-crime areas might be placed inside a bullet-proof booth. At the entrance of the booth is a card scanner. To enter the booth a customer must swipe their card which opens the booth door opens and turns the lights on. In this case, the CardScanner must inform both the DoorController and the LightsController that a customer has swiped their card. How can CardScanner notify the ATMController, in one case, and both the DoorController and the LightsController, in the other. When the CardScanner changes state (when a card is swiped), how can it notify controller objects of its state change without the CardScanner depending directly on which controller is using the CardScanner?



Ans.) Other variations are possible.