UML/OOAD/Design Patterns

성명:_____

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

Part I. *(18점)* **객관식 문제**

1. *(4점)* 다음과 같이 클래스 B가 클래스 A의 서브클래스로 선언된 경우, 그 의미와 **관계 있는** 것은? (Select all that applies)

1,3

```
class B extends A { ... }
```

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

2. *(2점)* 다음중 객체와 속성(attribute)의 관점에서 볼때, 열거된 다른 것과 가장 관계가 먼 것은? (Select one)

3,

① Year ② Age ③ Student ④ Address

3. *(2점)* 다음중 개발프로세스(development process)가 정의하고 있는 것과 가장 거리가 먼 것은?3

① Who ② When ③ Where ④ What ⑤ How

4. *(2점)* 다음 중 UP의 best practices와 관계 있는 것은? (Select all that applies)1,2,3,4

① Tackle high-risk and high-value issues in early iterations.
② Build a cohesive, core architecture in early iterations.
③ Adopt iterative and incremental development approaches based on actions performed by the system.
④ Apply use cases.
⑤ Adopt linear, document-driven approaches.

5. *(2점)* 다음 중 소프트웨어 개발 프로세스에서 분석(analysis)의 목적을 **가장 잘** 설명하고 있는 것은? (select one)3

① Analysis allows programmers to check that their data structures and algorithms are correct before their code is integrated into the system.
② Analysis translates system requirements into UML classes and relationships. This is important because the UML classes are then easily implemented in object-oriented languages such as Java or C++.
③ Analysis results in artifacts that capture the requirements of the system and document the domainl model of the problem domain that is shared by all parties (developers and clients) who have an interest in the system that is being developed.
④ Analysis helps clients understand how computers and software applications are developed by translating computer science concepts and terminology into the equivalent concepts in the problem domain.

6. *(2점)* UML과 가장 거리가 먼 것은? (Select one)2

① Graphical modeling language
② A development process
③ Static and dynamic models
④ Visual notation and semantics

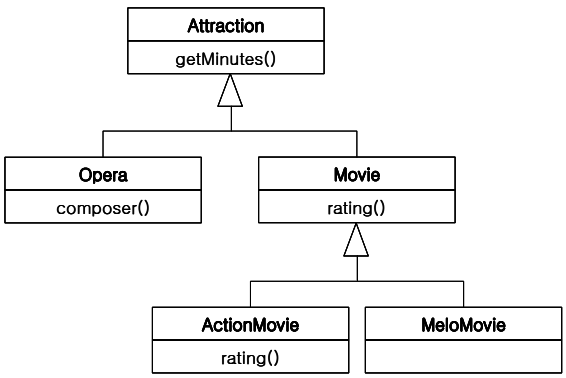
8. *(4점)* 다음은 UP Phase의 목적을 나타내는 목록이다.

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)

Part II. *(28점 - 각 2점씩)* **다음의 Class diagram을 보고 물음에 답하시오. 단, 모든 operation은 public method이며 dynamic binding이 적용된다고 가정한다.**

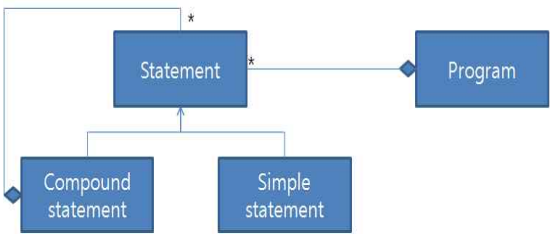


아래 Demo 클래스 main 메소드의 각 문장에 대해, 합법적인 문장이면 'Yes'로 표시하고, 그렇지 않으면 'No'로 표시하시오. 단, 합법적인 메소드 호출 문장의 경우 호출되는 메소드가 어느 클래스에서 정의된 것인지 보이시오. 예를 들어, Move 클래스의 rating()이 호출되면 Movie::rating()으로 표시하시오.

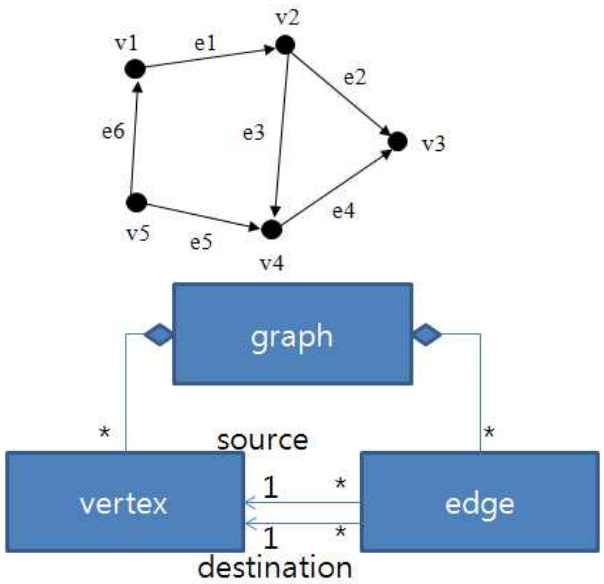
```
class Demo {
  public static void main(String[] args[]) {
    ActionMovie ref_am
        = new ActionMovie();          (1)_____
    ref_am.rating();                  (2)_____
    ref_am.getMinutes();              (3)_____
    Movie ref_m = new Movie();
    ref_m.rating();                   (4)_____

    ref_m = new ActionMovie();        (5)_____
    ref_m.rating();                   (6)_____
    ref_m.getMinutes();               (7)_____
    ref_m.composer();                 (8)_____

    Attraction ref_a = foo();         (9)_____
    ref_a.getMinutes();               (10)_____
    ref_a.rating();                   (11)_____
    ref_m = bar();                    (12)_____
    ref_m.rating();                   (13)_____

    Movie ref_m2 = foo();             (14)_____
    }
   public static Attraction foo()
   { return new ActionMovie(); }
   public static Movie bar()
   { return new MeloMovie(); }
}
```

Part III. *(16점)* 다음에 주어진 설명을 잘 읽고, 언급된 conceptual class들 사이의 각종 관계(association, aggregation, composition aggregation, generalization/specialization, multiplicity, etc.)를 가능한 한 상세히 보여주는 UML class diagram을 그리시오.

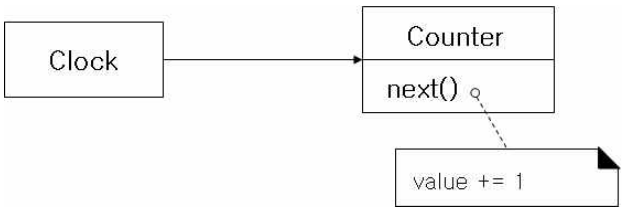
(1) Draw a class diagram to describe programs. A program consists of statements. A statement can be either a compound statement or a simple statement. In other words, compound statements and simple statements are special types of statements. A compound statement can contain either simple statements or compound statements or both.



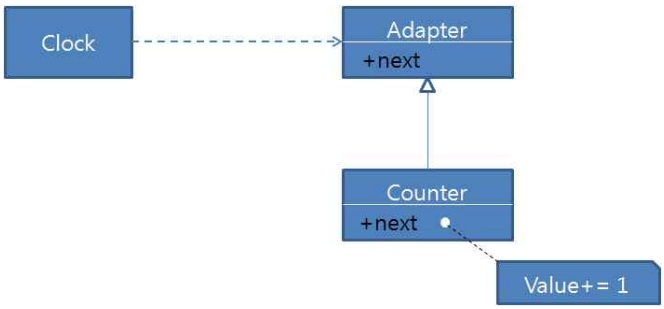
[2] Draw a class diagram to describe directed graphs. A directed graph consists of a set of vertices and a set of edges. Edges connect pairs of vertices (source vertex and destination vertex). Your diagram should capture only the structure of the graphs (i.e. connectivity), and need not be concerned with geometrical details such as lengths of edges. A typical graph is shown below.



Part IV. *(8점)* 다음아래 그림에서 보는 바와 같이, Clock 객체는 자신의 notify() method가 호출되면 자신과 연결된 Counter 객체의 next()를 호출하여 Counter 객체가 자신의 속성인 value를 증가시킬 수 있도록 설계되었다.

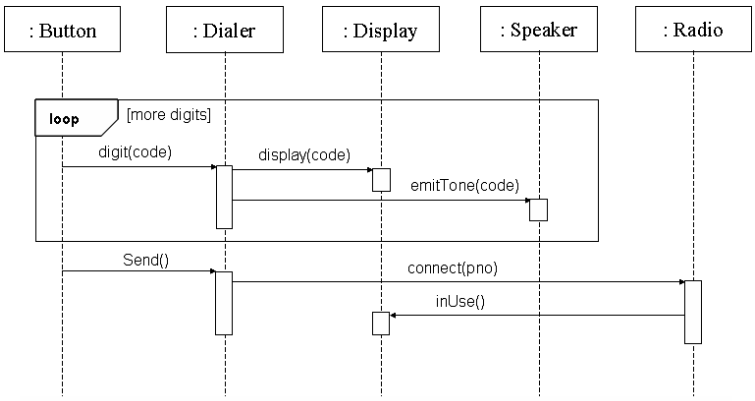


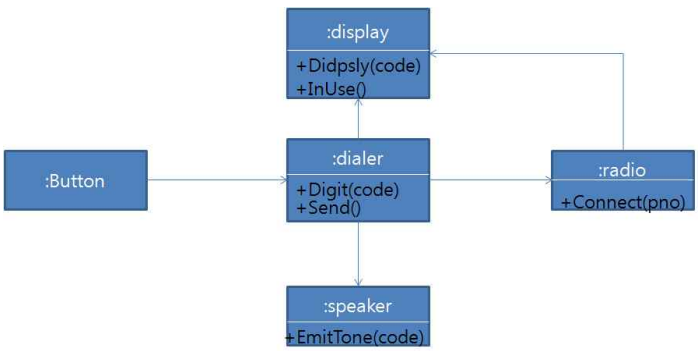
그러나, 이러한 설계는 Clock을 항상 Counter와 함께 사용해야 한다는 제약을 가지고 있다. Clock의 notify()가 호출되었을 때, 자신과 연결된 임의의 객체의 next()를 호출할 수 있도록 함으로써 Clock 객체를 특정 클래스(예, Counter)와 독립적으로 사용할 수 있도록 위의 설계를 변경하시오.



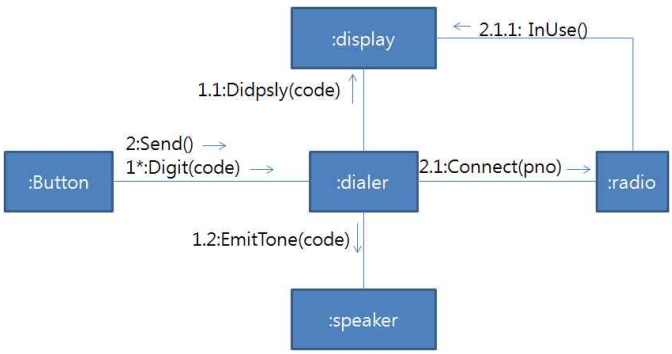
Part V. *(20점)*

[1] 아래에 주어진 sequence diagram으로부터 도출된 class diagram을 그리시오. (단, 객체간의 visibility는 모두 attribute visibility로 가정하시오.





(2) 위의 sequence diagram을 동일한 communication diagram으로 보이시오. 단, 메시지의 번호는 dewey decimal 표기법을 따르시오.



Part VI. *(10점)*

다음에 주어진 Use Case를 읽고 Main Success Scenario로부터 도출되는 SSD(System Sequence Diagram)을 보이시오. 단, 반드시 UML 2.0 표기법을 사용하시오.

Use Case: Checkout books
Primary Actor: Librarian
Main Success Scenario:

1. The Librarian enters the identity of a customer who wishes to check out books.
2. The System confirms that the customer is allowed to checkout books, and remembers the customer's identity.
3. The Librarian enters the identity of a book that the customer is checking out.
4. The System confirms that the book can circulate, calculates the due date based on whether the customer is a faculty member or a student, and records that the customer has checked out this book, which is due on the calculated due date.
5. The System tells the Librarian the due date.

The Librarian repeats steps 3-5 until indicates done.