

( 다음 페이지에서 계속됩니다 )

## 문항#5. 답안: (                    )

이름의 여러 종류들 중, 아래 예시 코드에서 사용되지 않은 것을 하나 고르세요: (3점, 2분)

```
System.out.println("Hello, World!");
```

1. Data 이름
2. Typename(형식 이름)
3. Code 이름
4. 소속 관계를 구성하기 위한 이름

## 문항#6. 답안: (                    )

Java의 primitive 형식에 대한 설명으로 옳지 않은 것을 모두 고르세요: (3점, 1분)

1. Java의 int, double, bool, char 형식은 primitive 형식입니다.
2. Primitive 형식의 경우 typename이 Java 키워드로 되어 있습니다.
3. Literal(예: 3)을 적을 수 있는 경우 해당 형식은 primitive 형식입니다.
4. Primitive typename들은 기본적으로 어떤 소속 관계도 갖지 않습니다.
5. 우리는 클래스 정의를 적음으로써 새 primitive 형식을 만들 수 있습니다.

## 문항#7

수식들 중 하나인 메서드 호출식의 형식은 어떻게 결정되는지(어떤 메서드에 대해 무엇을 어떻게 적어 두면 그 메서드에 대한 메서드 호출식의 형식이 지정되는지) 짧게 설명해 주세요. (4점, 2분)

( 다음 페이지에서 계속됩니다 )

## 문항#8

아래 코드 예시를 먼저 확인해 주세요:

```
System.out.println("Hello, World!");  
  
System.err.println("Hello, World!");
```

위의 두 문장은 Data 이름을 서로 다른 것을 적었을 뿐임에도 실행 양상이 서로 달랐습니다. 이러한 차이가 발생한 이유를 단어 '포함'을 포함하여 설명해 주세요. (7점, 5분)

## 문항#9

'.' 연산자 좌항 자리에 적는 것'을 가운데에 두고 static 메서드와 nonstatic 메서드의 차이점을 설명한 다음, 어떤 메서드를 static 메서드로서 정의할 때의 상대적인 장점을 간략하게 설명해 주세요. (총 10점, 6분)

## 문항#10

우리가 'Data 클래스'라 마음먹고 클래스 정의 내용물을 적는다 하더라도 실제로는 getter, setter 메서드를 추가로 정의함으로써 필드 수보다 메서드 수가 더 많아질 수 있습니다. 이러한 메서드 추가로 우리가 달성 가능한 효과를, Data 클래스의 의의에 빗대어 설명해 주세요. (5점, 4분)

서술 규칙: Data 클래스의 의의를 먼저 적고 나머지를 그 뒤에 이어 적어요.

( 다음 페이지부터 실습문항이 시작됩니다 )

( 11 ~ 13 ) 다음 선언들을 먼저 읽고, 문항별 목표들을 달성할 수 있는 수식을 적어 주세요.

```
int number = 3;  
  
int[] arr = new int[2];
```

### 문항#11

number 자리에 담긴 값을 제공하여 number 자리에 담는 수식 (2점, 1분)

### 문항#12

arr에 담긴 모든 값을 다 더한 값을 number 자리에 담는 수식 (3점, 2분)

### 문항#13

arr의 길이에 해당하는 값을 number 자리에 담는 수식 (3점, 1분)

( 다음 페이지에서 계속됩니다 )

( 14 ~ 15 ) 아래 표에 담긴 텍스트 전체를 답안지 파일에 복붙한 다음,

문항별 목표들을 달성할 수 있는 선언을 클래스 정의 안 적절한 자리에 적어 주세요.

### 문항#14, 15 답안:

```
public class Program {  
  
    public static void main(String[] args) {  
  
    }  
  
    static void Other() {  
  
    }  
  
}
```

- 선언을 어디에 어떻게 적는지가 매우 중요해요. 따라서 복붙한 클래스 정의 내용은 절대 고치지 말아 주세요.
- 여러분의 Java project에 있을 Program.java의 내용물을 위 내용으로 바꿔 적으면 바로 진행 가능해요.
- 문항에서 명시하지 않은 개념은 전혀 신경쓰지 않아도 괜찮아요.

### 문항#14 (문항 번호 복붙 안 해도 될 듯)

Program.main() 정의 중괄호 안에서만 볼 수 있는 새 int 형식 Data 이름 answer14에 대한 선언 (3점, 2분)

### 문항#15 (문항 번호 복붙 안 해도 될 듯)

Program.main() 또는 Program.Other() 정의 중괄호 안에서는 볼 수 있지만 (예시에는 안 적혀 있는) Character.Duel1()에서는 볼 수 없는 새 int 형식 Data 이름 answer15에 대한 선언 (4점, 3분)

( 다음 페이지에서 계속됩니다 )

( 16 ~ 17 ) 아래 표에 담긴 텍스트 전체를 답안지 파일에 복붙한 다음, 문항별 목표들을 달성할 수 있는 메서드 정의를 새로 적거나 완성해 주세요.

### 문항#16, 17 답안:

```
import java.util.Random;

public class Program {

    static Random rand;

    public static void main(String[] args) {

        rand = new Random();

        // 이 아래에 17번 답안을 채워 넣어 main() 정의를 완성할 예정

    }

}
```

### 문항#16 (문항 번호 복붙 안 해도 될 듯)

rand를 사용하여, 매 호출마다 0 이상, 100 미만의 임의의 int 형식 값을 return하는 새 메서드 Answer16()에 대한 메서드 정의 (7점, 3분)

- 아예 메서드 정의를 처음부터 새로 적어야 해요.

### 문항#17 (문항 번호 복붙 안 해도 될 듯)

Answer16()를 반복 호출하며, '100보다 작은 임의의 수를 두 번 뽑았을 때 첫 수는 50보다 작고 두 번째 수는 50보다 클 확률'을 계산하도록 main() 메서드 정의 완성하기 (11점, 10분)

작성 규칙:

- Answer16()을 매 반복마다 두 번 호출하여 임의의 수 두 개를 만들어야 해요.
- 나중에 확률 값을 계산하기 위해 필요한 값들을 매 반복마다 기록해 두어야 해요(출력은 안 해도 돼요)
- 반복은 중단하지 않아도 괜찮아요.
- 여러분의 답안에 대해 Eclipse가 어떤 오류도 내면 안 돼요.

Hint: 어떤 값을 기록해 둘 것인지 결정하고, 그 값을 담기 위한 준비를 해 두어야 해요.

( 18 ~ 19 ) 아래 두 문항에 대한 클래스 정의를 각각 적음으로써 어떤 게임의 '자물쇠 따기' 기능을 구성하려 합니다. 잘 읽고 순서대로 차근차근 도전해 주세요.

## 문항#18

먼저, 아래 규칙에 맞도록 class Lock에 대한 클래스 정의를 적어 주세요: (7점, 7분)

- 모든 Lock 인스턴스에는 int 값 하나(여기서는 pos로 명명)를 담을 수 있어야 합니다. (필드 이름은 자유롭게 정해도 좋아요)
- Nonstatic 메서드 Lock.TryPick()에 대한 메서드 정의를 적어 두어야 합니다. 이 메서드는 int 값 하나를 인수로 받으며, 해당 인스턴스의 pos 값과 인수 값의 차이가 -3 이상 3 이하면 true를, 그렇지 않다면 false를 return합니다.

## 문항#19 (마지막 문항)

마지막으로, 아래 규칙에 맞도록 class Program에 대한 클래스 정의를 적어 주세요: (10점, 11분)

- 평소 실습에서 만든 것과 동일한 느낌으로 Program.main() 메서드 정의를 자동완성당해야 합니다. 이 메서드는 호출하면...
  - 사용자 입력을 받기 위한 새 java.util.Scanner 인스턴스 하나를 적절히 만듭니다.
  - 새 Lock 인스턴스 하나를 만들고 pos 자리에 임의의 값을 담습니다.
  - 방금 만든 인스턴스들의 필드 자리에 임의의 값을 담습니다.
  - 아래 내용물을 반복 실행합니다:
    - ◆ 사용자에게서 임의의 int 값 하나를 입력받습니다.
    - ◆ 입력받은 값을 인수 자리에 담아 Lock.TryPick()을 호출합니다.
    - ◆ 이 때 return값이 true인 경우 적절한 성공 메시지를 출력한 다음 반복을 중단합니다.

Hint: 반복을 중단해야 하는지 여부를 담기 위한 새 local 이름을 선언해 두고 진행할 수 있어요.

( 시험 보느라 고생 많았어요. 실습문항은 부분점수 많으니 너무 걱정 말아요 )