2021년도 1학기 게임프로그래밍을위한캡스톤디자인 중간고사 (화요일)

문항#1

Data 관점에서 일반적인 프로그램들의 실행 흐름을 세 단계로 나누어 볼 수 있었습니다. 이 때의 각 단계들을 각각 30자 내외로 설명해 주세요. (총 4점, 5분)

문항#2

Data 이름을 '사용'하는 것이 어떤 의미인지 적고, Data 흐름을 구성하기 위해 Data 이름을 어떻게 사용할 수 있는지 적어 주세요. (두 가지 방법이 있어요) (총 4점, 6분)

문항#3. 답안: ( )

다음 보기들 중 문장에 대한 설명으로 옳지 <u>않은</u> 것을 <u>모두</u> 고르세요: (3점, 2분)

1. 문장은 실행의 대상이 되는 친구입니다.
2. 문장들은 프로그램 실행 도중 최소 한 번 이상 실행되며, 여러 번 실행되는 문장도 존재할 수 있습니다.
3. 모든 문장들은 그 디테일을 지정하기 위해 적절한 자리에 수식을 적게 됩니다.
4. 어떤 문장의 경우 수식을 적을 수 있는 자리가 둘 이상일 수 있습니다.
5. 우리는 분기 흐름을 구성하기 위해 if문을 적을 수 있습니다.

문항#4

프로그램의 실행을 Data 흐름과 Code 흐름의 양쪽 관점에서 바라볼 수 있습니다. 이들을 각각 간략히 설명한 다음, 두 흐름이 어떻게 밀접하게 연관되어 있는지를 단어 '문장'을 가운데에 두고 설명해 주세요. (7점, 7분)

(다음 페이지에서 계속됩니다)

문항#5(화수). 답안: ( )

수업자료에서 언급한, 프로그래밍에서 사용되는 이름에 대한 분류로 알맞지 않은 것을 하나 고르세요: (3점, 1분)

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

문항#6. 답안: ( )

Java의 primitive 형식에 대한 설명으로 옳지 <u>않은</u> 것을 <u>모두</u> 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

+ 연산자를 사용한 덧셈식을 예시로 들어, 프로그래밍에서 수식의 형식이 중요한 이유를 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

수식 this의 의미를, 어디에 적을 수 있는지, 계산하면 무슨 형식 값이 나오는지, 계산하면 어떤 값이 나오는지를 포함하여 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 값에 2를 더하여 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 `answer14`에 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.Duel()`에서는 볼 수 없는 새 `int` 형식 Data 이름 `answer15`에 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 3의 배수 값을 return하는 새 메서드 Answer6()에 대한 메서드 정의 (9점, 5분)

- 아예 메서드 정의를 처음부터 새로 적어야 해요.

문항#17 (문항 번호 복붙 안 해도 될 듯)

Answer6()를 반복 호출하며, '100보다 작은 임의의 3의 배수의 자릿수를 더했을 때 항상 3의 배수가 나오는지 여부'를 연거푸 확인하도록 main() 메서드 정의 완성하기 (10점, 8분)

작성 규칙:

- 반례(더했는데 3의 배수가 아닌 수가 나옴)을 발견한 경우 적절한 실패 메시지를 출력해야 해요.
- 반례가 나올 때까지 계속 반복 확인해야 해요.
- 여러분의 답안에 대해 Eclipse가 어떤 오류도 내면 안 돼요.

Hint: 아마 선언을 몇 개 추가로 적어야 할 거예요. 어떤 int 형식 값이 가 3의 배수인지 확인할 때는 % 연산자를 쓰거나 '3으로 나누고 다시 곱했을 때 동일한 값이 나오나' 확인하면 돼요.

(다음 페이지에서 계속됩니다)

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

문항#18

먼저, 아래 규칙에 맞도록 `class NPC`에 대한 클래스 정의를 적어 주세요: (6점, 7분)

- 모든 `NPC` 인스턴스에는 `int` 값 두 개(여기서는 각각 pos, speed로 명명)를 담을 수 있어야 합니다. (필드 이름은 자유롭게 정해도 좋아요)
- Nonstatic 메서드 `NPC.Advance()`에 대한 메서드 정의를 적어 두어야 합니다. 이 메서드는 호출하면 해당 인스턴스의 `pos`에 `speed` 값을 더한 다음, `pos` 값이 `100`보다 크거나 같다면 `true`를, 그렇지 않다면 `false`를 return합니다.

문항#19 (마지막 문항)

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

- 평소 실습에서 만든 것과 동일한 느낌으로 `Program.main()` 메서드 정의를 자동완성당해야 합니다.
  이 메서드는 호출하면...
  - 새 `NPC` 인스턴스 두 개를 만듭니다.
  - 방금 만든 인스턴스들의 필드 자리에 임의의 값을 담습니다.
  - 아래 내용물을 반복 실행합니다:
    - 두 인스턴스에 대해 각각 NPC.Advance()를 번갈아 가며 호출합니다.
    - 이 때 return값이 true인 경우 해당 인스턴스에 대한 적절한 승리 메시지를 Console 탭에 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)