**Object Oriented Programming and Practice, Midterm Exam**

DCCS213(02)

**Your ID#: [ 2021270660 ]**

**Your Name: [ 이지원 ]**

**Read each question carefully and do the programming! Be sure to follow the instructions below!**

|  |
| --- |
| **[Instruction] 안내**  **(1)** **Head Javadoc comment** and **standard naming rule** are essential for all program (**If failure -1 point**).  (상단 Javadoc 코멘트와 스탠다드 네이밍 룰은 이제 말하지 않아도 기본입니다. 실패 시 각 -1점 감점).  **(2)** Please compress (.zip) and submit this report (.docx), and the entire Java code (.java file) you wrote into the BlackBoard.  (**이 시험지와 답변에 사용된 자바 코드**를 압축하여 블랙보드에 지정된 시간까지 반드시 제출하십시오.)  Midterm: 16:00 PM ~ 17:30 PM (90 Min.)  \*I highly recommend that you start submitting to the Blackboard at 17:25PM, at least 5 minutes before the end. If you do not submit by the given time, you will receive 0 points.  (시험 끝나기 5분전인 5시 25분에는 블랙보드에 제출을 시작하길 권장합니다. 5시 30분에는 제출할 수 없도록 닫히며, 제출을 못했을 시 당연 0점 처리됩니다.)  **(3)** The Java code should be executable when the TA runs. The submitted compressed file (.zip) must be named Midterm\_YourID\_Yourname.zip.  (자바 코드는 TA가 돌렸을 시 깔끔하게 돌아가야 하며, 제출될 압축파일은 반드시 Midterm\_YourID\_Yourname.zip 로 명명하여 제출하십시오.)  **(4)** There is a partial score. Even if you can't resolve it completely, I hope you can try it as far as you can.  (부분 점수가 있으므로, 완벽하게 해결하지 못하더라도 할 수 있는 만큼 시도해 보길 바랍니다.)  **(5)** There are 6 questions in total, but I don't expect you to resolve all the questions within the given time. I hope you select the problem you can solve and solve it quickly. "Choice and focus" is also one of the programmer's core abilities. If you've followed the class normally and know your proficiency I'm sure you'll do it well.  (총 6개의 질문이 있지만, 이것을 제한시간내에 모두 해결할 수 없을 것입니다. 본인의 수준을 고려했을 때, 풀 수 있는 문제를 제한시간내에 가급적 많이 해결하십시오. “***선택과 집중***”은 프로그래머의 핵심 능력 중 하나입니다. 수업을 정상적으로 따라왔으며, 자신의 프로그래밍 숙달도를 알고 있는 학생이라면, 이를 잘 해낼 수 있을 것입니다.) |

**(Q1) Implement a Q1.java class that receives the name and student ID number from the dialog and outputs it as in the example below. (15 Points).**

**\*Tip:**

**(1) You must separate the entered name and student ID number automatically.**

**(2) Each digit of the number has to be cumulatively summed up through loop grammar.**

**For example, assuming your id “12345”, print out result of “1 + 2 + 3 + 4 + 5”**

(Dialog로부터 이름/학번을 입력 받아 아래의 예제처럼 출력하는 프로그램 Q1.java 를 작성하십시오).

(1) 입력한 이름과 학번은 자동으로 분리되어야 합니다.

(2) 숫자의 각 자릿수는 루프 문법을 통해 누적 합산되어야 합니다.

예를 들어, 귀하의 ID가 "12345"라고 가정하면 "1 + 2 + 3 + 4 + 5"의 결과를 출력하십시오.

|  |
| --- |
|  |

|  |
| --- |
| **[Result Screenshot]**  텍스트이(가) 표시된 사진  자동 생성된 설명 |

**(Q2)** Write a **GUI program (Q2.java) that asks for the user’s first, middle, and last names**. This program should generate a **monogram based on the last letters** of first, middle and last name **(10 Points).**

(사용자의 first, middle, last name을 받아 각 부분의 최종 문자를 바탕으로 약어를 만들 수 있는 GUI 프로그램 Q2.java 를 작성하십시오).

|  |
| --- |
|  |

|  |
| --- |
| **[Result Screenshot]**    텍스트이(가) 표시된 사진  자동 생성된 설명 |

**(Q3)** Freely design and implement a **summation** **calculator (Q3.java)** based on the **overloaded methods**. **You must implement the requirements** below **(15 Points).**

(오버로딩 메소드를 기반으로 합산 계산기 (Q3.java)를 자유롭게 설계하고 구현하십시오. 구현할 때, 아래 요구 사항을 반드시 구현해야 합니다.).

|  |
| --- |
| **(1) You must implement two overloaded getSum() functions should be available with 2, 3 integer parameters.**  (i.e. getSums(int firstNum, int secondNum) and getSums(int firstNum, int secondNum, int thirdNum) )  (반드시 2개 또는 3개의 정수를 매개변수로 하는 두 오버로딩 getSum() 함수를 구현하십시오).  **(2) Overloaded getSum() functions should be available also with int [] parameter for multiple values.**  (4개 이상의 정수를 받을 수 있는 오버로딩 getSum() 함수를 구현하십시오).  **(3) Overloaded getSum() functions should be available also with double [] parameter for multiple double values.**  (다수의 실수를 받을 수 있는 오버로딩 getSum() 함수를 구현하십시오).  **(4) Implement a UI that can sum through the overloaded getSum() functions when an integer or double values are entered from the UI.**  (어떠한 실수라도 받을 수 있는 UI를 자유롭게 구현하고, 구현된 overloaded methods에 따라 합산 결과를 제공하십시오.)  **(5) Exception handling is essential! (non-integer or non-double values should be processed).**  예외 처리는 필수입니다! (정수 또는 실수가 아닌 값은 예외 처리되어야 함). |

|  |
| --- |
| **[Result Screenshot]**  텍스트이(가) 표시된 사진  자동 생성된 설명  텍스트이(가) 표시된 사진  자동 생성된 설명 |

**(Q4)** Design and write a program (**Q4.java**) that receives a total of 9 integers and outputs all received values, row sum, column sum, and total sum as follows: **(10 Points).**

(사용자로부터 총 9개의 정수를 입력받아 다음과 같은 형식으로 입력값과 함께, 행 합, 열 합, 총 합을 출력하는 프로그램, Q4.java, 를 자유롭게 설계하고 구현하십시오).

|  |
| --- |
|  |

|  |
| --- |
| **[Result Screenshot]**  텍스트이(가) 표시된 사진  자동 생성된 설명 |

**(Q5) Write a Student & Q5.java main classes** that satisfies the following requirements based on the following class diagram **(30 Points):**

(Kor. 아래의 요구조건을 만족할 수 있도록 Student 클래스와 Q5 메인 클래스를 클래스 다이어그램을 기반으로 구현 하십시오).

|  |
| --- |
| **[Class Diagram]** |

**(Step 1)** Implement data members for student's name, id, gender, and grades, and grant suggested access.

(Kor. 이름, 학번, 성별, 총학점을 제시된 접근 제어자에 따라 데이터 맴버로 구현하십시오)

**(Step 2)** Implement a constructor that can initialize a given data members and a member methods that can insert or subtract values ​​from each data member.

(Kor. 주어진 데이터 맴버를 초기화 할 수 있는 생성자와 각 데이터 맴버에 값을 넣거나 뺄 수 있는 맴버함수를 구현하십시오.)

**(Step 3)** studentName and studentID should be randomly determined to avoid overlapping created objects.

(Kor. studentName과 studentID는 생성되는 객체끼리 겹치지 않도록 랜덤하게 결정되어야 합니다.)

**(Step 4)** studentSex must be randomly assigned a value of "Male" or "Female", and studentGPA should be randomly generated within 0.0 to 4.5 scale.

(Kor. studentSex 는 "Male " 또는 "Female" 값으로 랜덤하게 할당되어야 하며, studentGPA는 0.0 ~ 4.5 내에서 랜덤하게 생성하십시오.)

**(Step 5)** Generate 10,000 students from the main class Q5.java who are randomly assigned each data member based on the implemented Student class.

(Kor. 구현된 Student 클래스를 바탕으로 각 데이터 맴버값을 랜덤하게 부여 받은 10,000명의 학생을 메인 클래스 Q5에서 생성하십시오.)

**(Step 6)** Finally, implement a method that displays the name, student number, gender, and GPA of the student with the highest GPA among 10,000 randomly generated students, and call it in the main function to find it.

(Kor. 마지막으로, 랜덤하게 생성된 10,000명의 학생 중 최고의 GPA를 보이는 학생의 이름, 학번, 성별, 학점을 보이는 함수를 구현하고, 메인 함수에서 호출하여 찾아내십시오.)

|  |
| --- |
| **[Result Screenshot]** |

**(Q6)** Implement the **system program Q6.java** that meets the following requirements: **(20 Points)**

(Kor. 다음 요구조건을 만족하는 시스템 프로그램 Q6.java 를 구현하십시오.).

|  |
| --- |
| **(1)** We will create a simple compiler that can decipher simple arithmetic ('+', '-', '\*', '/') expressions.  간단한 산술('+', '-', '\*', '/') 표현식을 해독할 수 있는 컴파일러를 만들 것입니다.    **(2)** All formula inputs are assumed to be space-separated.  (모든 수식의 입력값은 공백으로 구분되어 있다고 가정합니다.)  **(3)** We assume that the formula compiler we've built unconditionally performs left-to-right decoding, regardless of operator precedence.  (우리가 만든 수식 컴파일러는 연산자 우선 순위에 관계없이 무조건 왼쪽에서 오른쪽으로 디코딩을 수행한다고 가정합니다.)  **(4)** As shown in the example screenshot above, when we enter an expression as an argument to the main function, write a program so that we can display the operation sequence and operation result of the arithmetic interpreter we created step by step.  (위의 예시 스크린샷과 같이 메인 함수에 인수로 표현식을 입력하면 우리가 생성한 산술 인터프리터의 연산 순서와 연산 결과를 단계별로 표시할 수 있도록 프로그램을 작성하십시오).  **(5)** Also, you should print out the operation result through the actual Java compiler that implements the operator precedence rule as shown in the screenshot above.  (또한, 위의 스크린샷과 같이 연산자 우선순위 규칙을 구현한 실제 자바 컴파일러를 통한 연산 결과도 출력하십시오.) |

|  |
| --- |
| **[Result Screenshot]** |

**-At the end-** Well done for half of semester!