**Homework #2 Object-Oriented Programing & Practice (2023 Spring)**

|  |
| --- |
| **\* Java-doc comments must be included in all parts (If it fails -1 point).**  **\* You must follow the Java programming standard naming (If it fails -1 point).**  **\* An F grade will be awarded immediately if a copy of another student’s code is caught.**  **Please fill out and submit the relevant contents in this report (.docx), and additionally submit all the written code (.java) files.**  **The format of the file name of the report is as follows:**  **(1) OOP\_Homework3\_12345678\_MinseokSeo.docx**  **Compress the code related to this report (\*.java) and the report (.docx) as (.zip) format and submit it to the Blackboard. Make the file name of the compressed file the same as above.**  **(Kor.)**  **\* Java doc 주석은 모든 코드에 적절하게 포함되어야 합니다(실패 시 -1점).**  **\* 자바 프로그래밍 표준 네이밍을 따라야 합니다(실패 시 -1점).**  **\* 다른 학생의 코드가 적발될 경우 즉시 F등급이 부여됩니다.**  **본 보고서(.docx)에 해당 내용을 작성하여 제출하고, 작성된 모든 코드(.java) 파일을 추가로 제출하시기 바랍니다.**  **보고서의 파일 이름 형식은 다음과 같습니다.**  **(1) OOP\_Homework3\_12345678\_MinseokSeo.docx**  **본 보고서(.docx) 및 관련 코드(\*.java)를 (.zip) 형식으로 압축하여 블랙보드에 제출합니다. 압축 파일의 파일명은 위와 동일하게 합니다.** |

**[Q1] Write a Student class** that satisfies the following requirements**:**

(Kor. 아래의 요구조건을 만족할 수 있도록 Student 클래스를 작성하십시오)

**(Step 1) Department, student ID, gender, height, and weight** are data members.

(Kor. 학과, 학번, 성별, 키, 몸무게를 데이터 맴버로 구현합니다)

**(Step 2)** **Implement 3 overloaded constructors** that can initialize the corresponding data members on a case-by-case basis.

(Kor. 3개의 오버로딩 생성자를 구현하십시오. 이 생성자들은 각자 다른 상황에서 객체를 만들 수 있도록 디자인 하십시오.)

**(Step 3)** Student ID is set globally starting from Student\_00001 and **incremented by one** each time a new object is created (*i.e. Student\_00001, Student\_00002, Student\_00003, ..., Student\_0000N*)

(Kor. 학번은 Student\_00001 부터 순차적으로 00002, 00003 으로 부여될 수 있도록 구현하십시오.)

**(Step 4)** In the main class Q1.java, **create 10,000 students** through overloaded constructors of all four types implemented. At this time, "gender", "height", and "weight" can be determined randomly. Specifically, when randomly generating “weight” and “height”, generate random numbers from a normal distribution. Generate random numbers so that the mean of the heights to be generated is **173cm for men** and **162cm for women**. In case of weight, average weight to be generated should be **68 kg for males** and **52 kg for females**. **(※Tip:** *nextGaussian() method within java.util.Random package* **will be required)**

(Kor. Q1.java 클래스에서 10,000명의 학생을 위에서 구현한 오버로딩 생성자를 통해서 생성하십시오. 이 때, 성별, 키, 몸무게 등은 랜덤하게 결정될 수 있도록 합니다 키와 몸무게에 대한 난수생성시 정규 분포로부터 확률적으로 생성될 수 있도록 하십시오. 생성될 키의 경우 남자는 평균 173 여자는 162의 평균을 가진 정규분포에서 생성할 수 있도록 하십시오. 생성될 몸무게의 경우 남자는 68 여자는 52가 되도록 생성하십시오. nextGaussian() 함수가 도움이 될 것입니다).

**(Step 5) inherit and implement the SimpleStatistics abstract class below**, and test whether it works normally using the 10,000 data generated above.

(Kor. 마지막으로 하단의 SimpleStatistics 추상 클래스를 상속받아 구현을 완성하십시오. 또한, 구현이 제대로 됐는지를 위에서 생성한 10,000명의 학생자료로부터 시험해보십시오)

텍스트이(가) 표시된 사진

자동 생성된 설명

**(Step 6)** Create a GUI-based UI that can receive data member information from the user, and calculate and show how much the height and weight of the user differ from the already generated height and weight average of 10,000 people when the button is pressed.

(Kor. 유저로부터 데이터 맴버 정보를 입력받을 수 있는 GUI기반 UI를 만들고, 버튼을 누를 시 유저의 키와 몸무게가 이미 입력되어있는 10000명의 키 및 몸무게 평균과 얼마나 차이가 있는지를 계산하여 보여주십시오.)

|  |
| --- |
| **[Result Screen-shot]** |

**[Q2]** Let's **design/implement the following student information system** that can manage objects derived from **[Q1]** **(20 Points):**

(Kor. [Q1]에서 파생된 객체를 관리할 수 있는 다음과 같은 학생 정보 시스템을 설계/구현해 봅시다.)

**(Req. 1)** Let's **freely design/implement GUI software with the function to add student** into the entire student list by inputting information such as student's gender, weight, and GPA.

(Kor. 학생의 성별, 체중, 평점 등의 정보를 입력하여 전체 학생 목록에 학생을 추가하는 기능을 가진 GUI 소프트웨어를 자유롭게 설계/구현해 봅시다.)

**(Req. 2)** Freely design/implement a GUI software that can view all registered/created student information. (Kor. 전체 등록/생성된 학생 정보를 열람할 수 있는 GUI 기반 UI를 자유롭게 설계/구현 하십시오.)

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

**[Q3]** Let's export the entire registered/created student objects as ***"backup[your ID]\_[yourName].data*"**. Also, implement a method that can read the backed-up student object file above to check whether the file backup function for the object works normally. **(20 Points):**

***(\*Hint: Serializable interface & ObjectOutputStream class will be required)***

(Kor. 등록/생성된 전체 학생 객체를 "backup[your ID]\_[yourName].data"로 내보내십시오. 또한, 백업된 student 객체 파일을 읽어올 수 있는 함수부분도 구현하여, 객체에 대한 파일 백업 기능이 정상적으로 작동하는지 확인하십시오.)

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

**[Q4]** Let's develop software that applies parallel processing algorithms based on multithreaded programming through the following requirements: **(20 Points):**

**(Req. 1)** Let’s **calculate/print the average GPA of all enrolled students**.At that time, please create/register enough number of students to take more than 3 seconds of computation time. ([Q1] applicable).

(Kor. 등록된 모든 학생의 평균 GPA를 계산/출력하십시오. 이때, 연산시간이 3초 이상 소요될 만큼 충분한 수의 학생을 생성/등록하시기 바랍니다. ([Q1] 응용 가능).

**(Req. 2)** **Implement a multithread-based algorithm** that can utilize the total threads of the PC you are currently using to get the average faster. Be sure to **show the elapsed time difference** between **before and after** parallel computation.

(Kor. 현재 사용 중인 PC의 전체 스레드를 활용하여 평균을 더 빠르게 얻을 수 있는 다중 스레드 기반 알고리즘을 구현하십시오. 이 때, 병렬 계산 전과 후의 경과 시간 차를 출력하여 확인하십시오.)

**(\*Hint: There is a summing process in the process of finding the average.)**

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

**[Q5]** Assuming that the calculation process in the student management software implemented in [Q2 ~ Q4] is a time-consuming high-level operation, it is reasonable that all calculations should be performed on the server. Please **design/implement the Server-Client model-based software** so that all processes other than the UI implemented above [Q2 ~ Q4] can be performed on the server. **(20 Points):**

(Kor. [Q2~Q4]에서 구현된 학생 관리 소프트웨어의 계산 프로세스가 시간이 많이 소요되는 고등 작업이라고 가정합시다. 그러면 모든 연산 과정은 서버에서 수행하는 것이 합리적입니다. 이를 위해, 위의 문제에서 구현한 UI를 제외한 모든 작업 과정 [Q2 ~ Q4]이 서버에서 수행될 수 있도록 Server-Client 모델 기반 소프트웨어를 설계/구현하시기 바랍니다.)

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

**[Q6] Let's get acquainted with the operation of applying/extending known open source through the steps below.**

(Kor. 아래 단계를 통해 알려진 오픈 소스를 적용/확장하는 작업에 대해 알아 봅시다.)

**(Step 1)** From **https://www.javatpoint.com/notepad** example, click "*download this example*" at the bottom to download the source code, then decrypt the code and get used to it. Several files appear when unzipped but copy only 5 java files in total to the "src folder" of your eclipse specific project folder.

(Kor. **https://www.javatpoint.com/notepad** 예제로부터 하단의 "download this example" 를 눌러 소스코드를 다운받은 뒤 코드를 해독하고 익숙해지십시오. 압축을 풀면 여러 파일이 나타나지만 총 5개의 Java 파일만 Eclipse 특정 프로젝트 폴더의 "src 폴더"로 복사하십시오.)

**(Step 2)** In the notepad program, **create a new "MyMenu" menu bar** **and sub-item "Copy Event"** UI as shown below. Also, **add a "Click Here" button** at the bottom center of the main UI.

(Kor. 해당 프로그램에서 아래와 같이 "MyMenu" 메뉴바와 하위 item "Copy Event" UI를 새로 만드십시오. 또한, 메인 UI 정중앙 하단에 "Click Here" 버튼을 추가하십시오.

텍스트, 스크린샷, 소프트웨어, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

**(Step 3)** Implement an event so that when the "Copy Event" menu or "Click Here" button is pressed, copy-paste is automatically applied to all text on the notepad program.

For example,

ABCDE

[1st event]

ABCDE

ABCDE

[2nd event]

ABCDE

ABCDE

ABCDE

ABCDE

(Kor. "Copy Event" 메뉴 또는 "Click Here" 버튼을 누르면, 화면상 모든 text 가 복사-붙여넣기가 자동으로 적용되도록 이벤트를 구현하십시오.)

**-At the end-** Well done!