**2022 Spring OOP Assignment Report**

과제 번호 : ASSN #2 Prob2

학번 : 20210479

이름 : 이주현

Povis ID : lanthanum14

**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **프로그램 개요**
   * Prob2는 1에서 구현한 Vector class를 응용해 피자 가게에서 배달원을 고용해 손님들에게 피자를 배달하는 상황에 필요한 class들을 구현하는 문제이다.
   * Prob2에서는 구현을 위해 Pizza, Customer, Rider, Vector\_Customer, Vector\_Rider라는 다섯 개의 class를 필요로 한다. 이들 class를 사용하기 위해서는 main에 #include로 헤더파일을 포함해야 하며. 해당 class의 객체를 선언함으로써 사용할 수 있다.
   * 피자 가게를 의미하는 Pizza class의 객체와 멤버 함수를 통해 손님과 배달원들, 가게 간의 상호작용을 다룰 수 있다.
2. **프로그램의 구조 및 알고리즘**

* Pizza Class: 피자 가게를 의미하는 class이다. private 변수로 A, B 구역의 가격과 배달료. 각 구역의 손님 리스트와 배달원 리스트를 가지고 있어 각각을 관리할 수 있다.
  + 다른 함수에서는 따로 선언한 변수 없이 각 객체들의 멤버 함수를 가지고 구현할 수 있었지만, enroll 같은 경우에는 새로 손님을 등록하기 위해서 삭제해야 하는 손님 객체 개수를 저장할 변수들을 선언하여 구현하였다.
  + 이 클래스에서 가장 핵심이 되는 함수는 deliver이다. 이 함수는 ASSN pdf에 제시된 것과 같은 라이더 배치 규칙을 준수하여 구현되었으며, 그 과정에서 라이더 배열 인덱스를 저장할수 있는 변수 riderInd와, A구역의 손님 수가 5명 이상이라 라이더가 A 구역에서 두 번 배달을 진행하였는지를 알려주는 loopCnt라는 변수를 선언하여 이용하였다.
  + 또한 pdf에 제시된 것처럼 pizza class의 << operator를 지원하기 위한 friend 함수도 구현하였다.
* Customer Class: 손님에 해당하는 class이다. private 변수로 해당 손님이 가지고 있는 돈, 지불한 돈, 기다린 시간을 가지고 있으며 각각을 관리할 수 있다.
  + 이 class 역시 pdf에 제시된 것처럼 << operator를 지원하기 위한 friend 함수를 가지고 있다.
* Rider Class: 배달원에 해당하는 class이다. private 변수로 해당 배달원이 소지하고 있는 돈, 번 돈, 배달에 걸린 시간을 가지고 있으며 각각을 관리할 수 있다.
  + 이 class 역시 pdf에 제시된 것처럼 << operator를 지원하기 위한 friend 함수를 가지고 있다.
* Vector\_Customer Class: Customer class를 다룰 수 있는 Vector를 구현한 class이다. Prob1과 같은 방법으로 구현되어 있다.
  + 생성자에서 Vector의 size를 입력해 준 경우, 이 class의 경우에는 arrCapacity = realLength = size가 된다는 차이점을 갖는다. 나머지 함수들은 거의 동일하다.
* Vector\_Rider Class: Rider class를 다룰 수 있는 Vector를 구현한 class이다. Prob1과 같은 방법으로 구현되어 있다.
  + 생성자에서 Vector의 size를 입력해 준 경우, 이 class의 경우에는 arrCapacity = realLength = size가 된다는 차이점을 갖는다. 나머지 함수들은 거의 동일하다.

1. **토론 및 개선**
   * + 이번 과제를 통하여 여러 개의 class와 header file을 구현하고자 할 때, #pragma once를 어떻게 사용할 수 있는지, 그리고 사용해야 하는 이유가 무엇인지 알 수 있었다.
     + class들과 그 객체를 활용하여 프로그래밍을 하는 방법에 대해 좀 더 자세하게 알 수 있었다.
     + Prob2에서는 Prob1에서 구현해 놓은 Vector를 이용했으나, Vector에서 다루는 변수 타입이 달라질 때마다 각각 구현해야 한다는 점이 매우 번거로웠다. Template을 쓴다면 한번 Vector를 구현함으로써 모든 변수에 대해 동일한 코드를 사용할 수 있어서 code conciseness가 높아지고, efficiency 면에서도 월등할 것이라고 생각한다.
2. **참고 문헌**