

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 2**

з дисципліни “Програмування. Об'єктно-орієнтоване

програмування та шаблони проєктування”

тема “C# .Net. Розширені можливості реалізації ООП у мові C#. Події.”

|  |  |  |
| --- | --- | --- |
| Виконала  студентка II курсу  групи КП-02  Красношапка Анастасія Андріївна |  | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна |

Київ 2021

**Мета роботи**

Ознайомитися з такими можливостями мови програмування C# як абстрактні класи, інтерфейси, делегати. Вивчити механізми оброблення подій у C#, а також можливості, які мають методи-розширення.

**Постановка задачі**

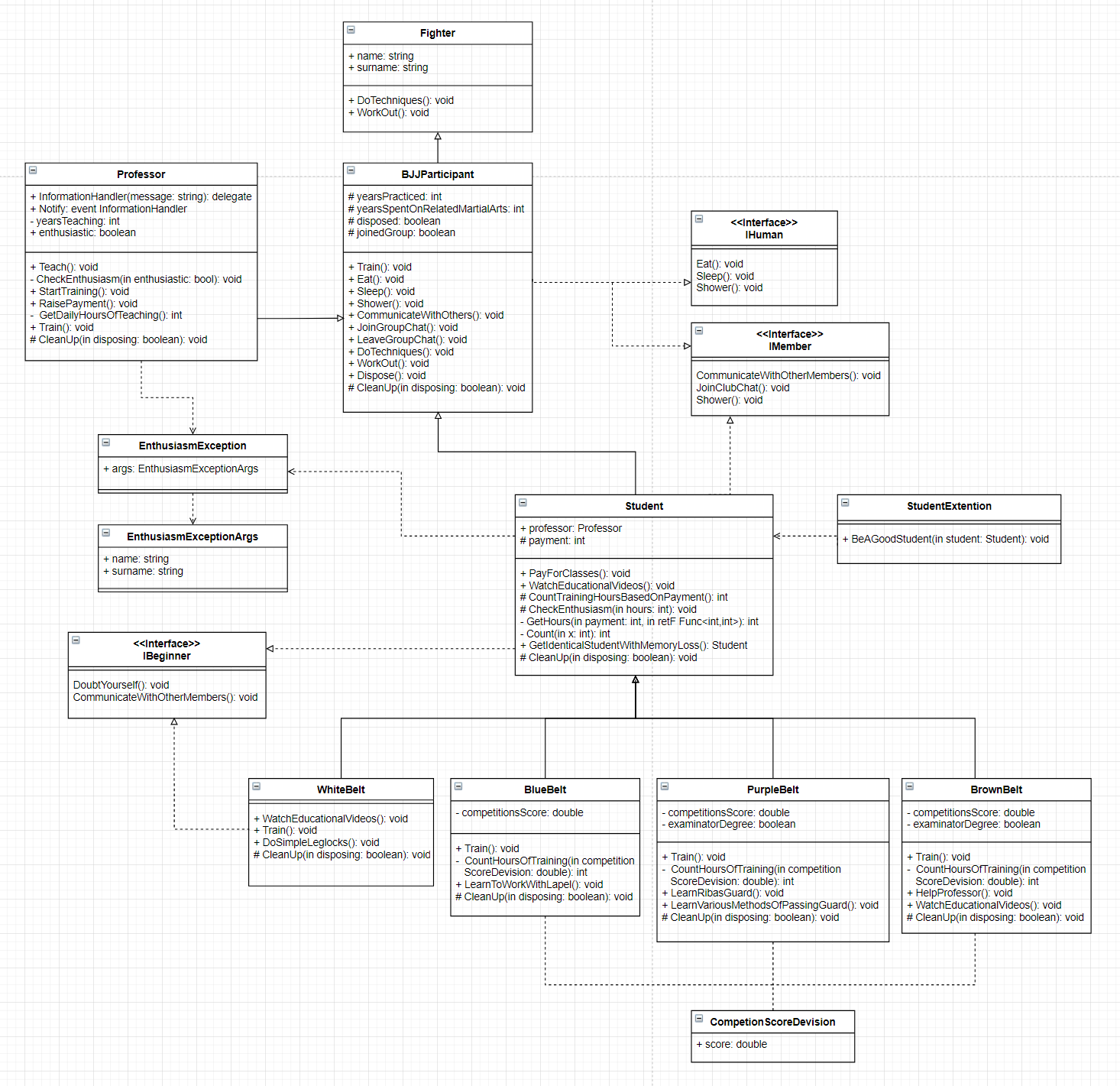
Для ієрархії класів, побудованої в лабораторній роботі №1, реалізувати:

1. Множину інтерфейсів. При чому один з класів повинен реалізовувати щонайменше 2 інтерфейси. Також продемонструвати реалізацію explicit implementation інтерфейса, обгрунтувати її використання (1 бал).
2. Абстрактний клас. Забезпечити його наслідування. Наявність в цьому класі абстрактних методів - обов’язкова (1 бал).
3. Механізм «делегат – подія – обробник події» (2 бали).
4. Перетворити код, який забезпечує роботу з подіями та обробниками подій, на код, що використовує (\*) (2 бали):
5. анонімні методи;
6. lambda-вирази;
7. типи Action та Func (кожен з них).

(\*) - допускається реалізація коду однієї події різними способами, необов’язково різних подій.

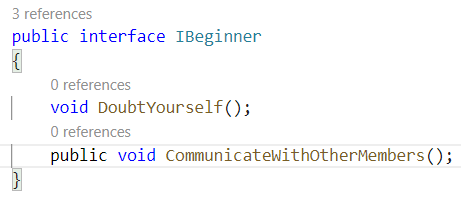
1. Механізм створення та оброблення власних помилок (2 бали):
2. створити новий клас виключної ситуації;
3. створити новий клас аргументів для передачі їх до обробника виключної ситуації;
4. забезпечити ініціювання створеної виключної ситуації та продемонструвати, як працює обробник даної помилки;
5. реалізувати різні сценарії оброблення помилки.
6. Метод-розширення будь-якого класу (1 бал).

**UML діаграма класів**

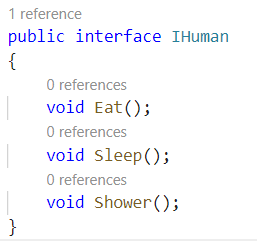


**Виконання поставлених задач**

1. Множину інтерфейсів. При чому один з класів повинен реалізовувати щонайменше 2 інтерфейси.

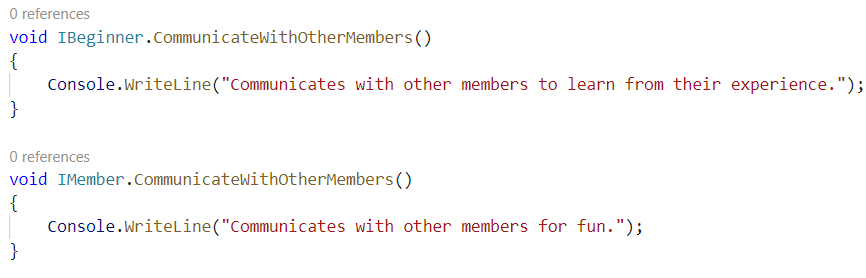




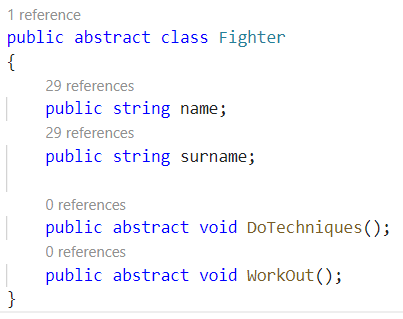




Також продемонструвати реалізацію explicit implementation інтерфейса.

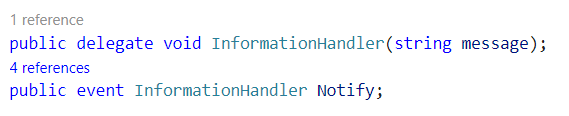


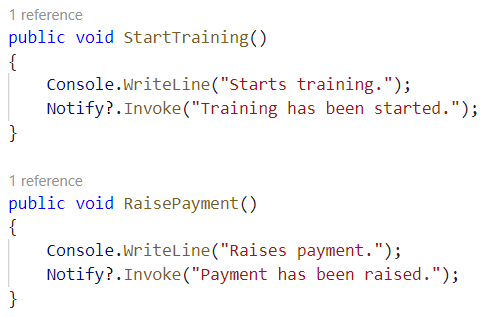
2. Абстрактний клас. Забезпечити його наслідування. Наявність в цьому класі абстрактних методів - обов’язкова.

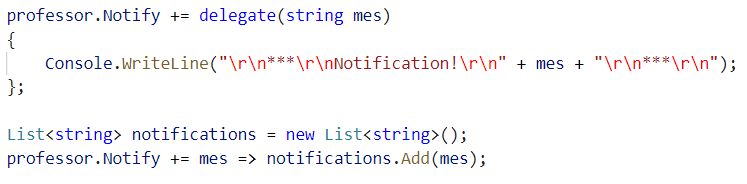




3. Механізм «делегат – подія – обробник події».

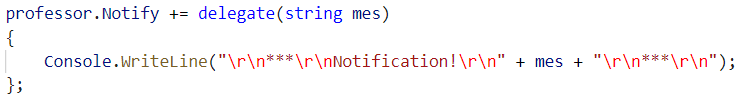






4. Перетворити код, який забезпечує роботу з подіями та обробниками подій, на код, що використовує (\*):

a) анонімні методи;

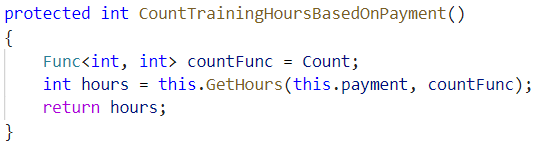


b) lambda-вирази;



c) типи Action та Func



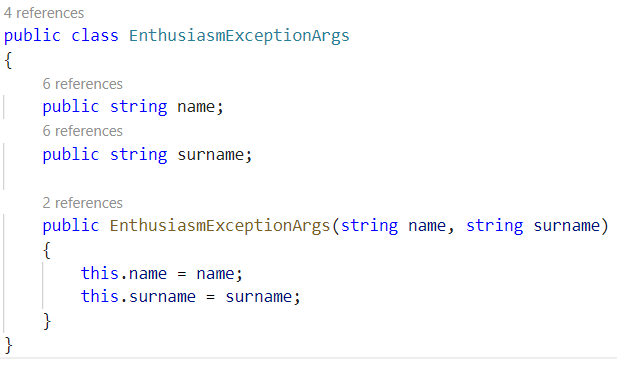


5. Механізм створення та оброблення власних помилок:

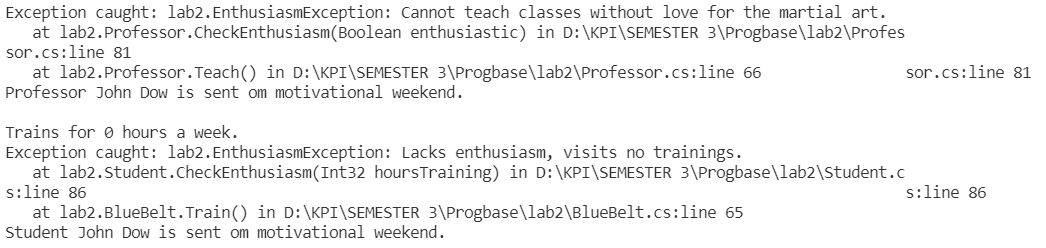
1. створити новий клас виключної ситуації;



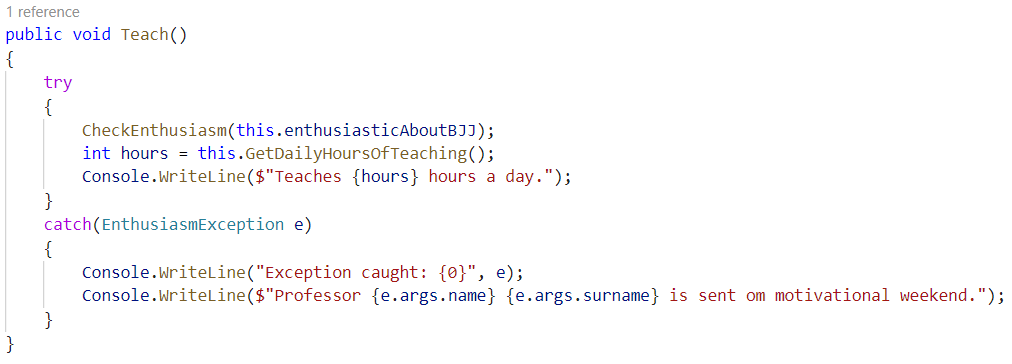
1. створити новий клас аргументів для передачі їх до обробника виключної ситуації;

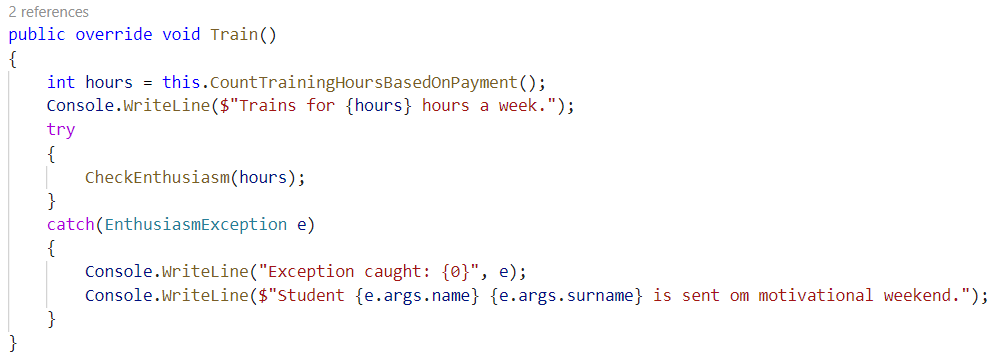


1. забезпечити ініціювання створеної виключної ситуації та продемонструвати, як працює обробник даної помилки;

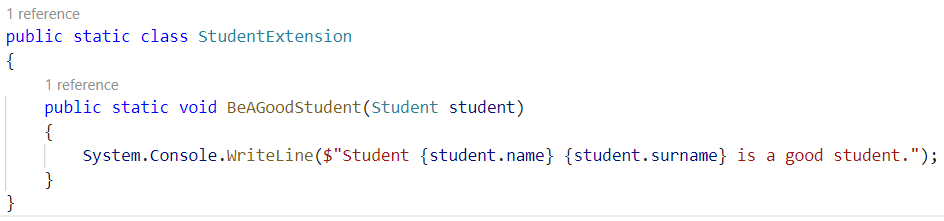


1. реалізувати різні сценарії оброблення помилки.





6. Метод-розширення будь-якого класу.



**Висновки**

Під час лабораторної роботи я зрозуміла особливості використання делегатів, абстрактних класів. Навчилися використовувати події, анонімні методи, lambda-вирази, Func та Action. Навчилася створювати власні типи Exception та повторила оброблення помилок. А також навчилася створювати методи-розширення для недоступних класів.