

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 “Об'єктно орієнтоване програмування”

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

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) II курсу  групи КП-01  Северин Юрій Юрійович  (*прізвище, ім’я, по батькові*) | Київ 2021 | Перевірила  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

**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 завдання**

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)**:
   1. анонімні методи;
   2. lambda-вирази;
   3. 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)**:
   1. створити новий клас виключної ситуації;
   2. створити новий клас аргументів для передачі їх до обробника виключної ситуації;
   3. забезпечити ініціювання створеної виключної ситуації та продемонструвати, як працює обробник даної помилки;
   4. реалізувати різні сценарії оброблення помилки.
2. Метод-розширення будь-якого класу **(1 бал)**.

**Текст коду програми**

|  |
| --- |
| **Interfaces.cs** |
| interface ISuperPower  {  string SuperPower();  }  interface ICarma  {  void IncreaseCarma();  void DecreaseCarma();  }  interface IDoneGood  {  void IncreaseCarma();  } |

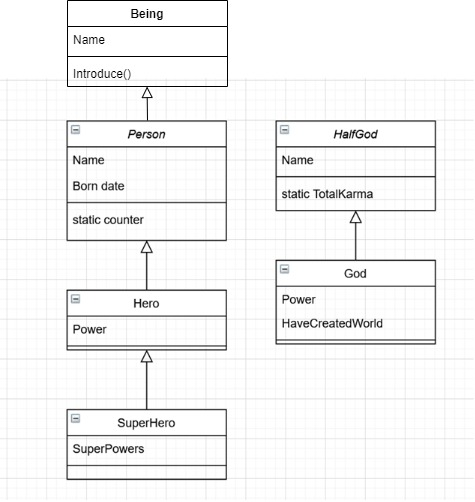
|  |
| --- |
| **Being.cs** |
| public abstract class Being  {  public abstract string Name { get; set; }  public abstract string Introduce();  } |

|  |
| --- |
| **Person.cs** |
| public delegate void FeedPerson(string s);  public class Person : Being, IDisposable  {  public event FeedPerson FeedEvent;  Action<string> action = Notify;  Func<string, string> function = str => { Console.WriteLine(str); return ""; };  private static void Notify(string obj)  {  Console.WriteLine(obj);  }  public void Feed(int feed)  {  Hunger += feed;  FeedEvent.Invoke($"1. I'm {name} and my hunger is: {Hunger}");  action($"2. I'm {name} and my hunger is: {Hunger}");  function($"3. I'm {name} and my hunger is: {Hunger}");  }  } |

|  |
| --- |
| **Program.cs** |
| {  static void Main(string[] args)  {  try  {  Person p1 = new Person("Ivan Ivanov", DateTime.Now.AddDays(1));  }  catch (BornDateException ex)  {  Console.WriteLine("Have caugth exception: " + ex.Message);  }  Person p2 = new Person("EVA Braun");  Console.WriteLine(p2.Beep());  Console.WriteLine();  Person p3 = new Person("Pupkin");  p3.FeedEvent += Notify;  p3.FeedEvent += anonimMethod;  p3.FeedEvent += lambda;  p3.Feed(10);  Console.ReadKey();  }  public static void Notify(string message)  {  Console.WriteLine("Notify " + message);  }  static FeedPerson anonimMethod = delegate (string mes)  {  Console.WriteLine("anonimMethod "+ mes);  };  static FeedPerson lambda = (s) => Console.WriteLine("lambda "+ s);  }  public static class ExtensionsClass  {  public static string Beep(this Person pers)  {  Console.Beep();  return "Beep boop boop";  }  } |

|  |
| --- |
| **InvalidDateException.cs** |
| public class InvalidDateException : Exception  {  public InvalidDateException() { }  public InvalidDateException(string message) : base("Date of birth cannot be in future! " + message) { }  public InvalidDateException(ExceptionData data) : base("Date of birth cannot be in future! " + data.message) { }  public InvalidDateException(string message, Exception innerException) : base("Date of birth cannot be in future! " + message, innerException) { }  }  public class ExceptionData  {  public string message;  public ExceptionData(string message)  {  this.message = message;  }  } |

**UML діаграма**



**Висновки**

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

**Котрольні питання:**

1. Що таке змінна? Як оголосити і ініціалізувати нову змінну?

Змінна – іменована область пам’яті, в якій зберігаються дані певного типу.

Синтаксис оголошення змінної виглядає так:

|  |
| --- |
| тип\_змінної назва\_змінної; |

Синтаксис присвоєння значення:

|  |
| --- |
| назва\_змінної = значення; |

або при оголошенні:

|  |
| --- |
| тип\_змінної назва\_змінної = значення; |

1. Що таке тип даних? Назвіть базові числові типи даних.

Тип даних – тип значення що може зберігатись у змінній.

* Int – цілочисельний
* Double – дійсний з подвійною точністю
* Float – дійсний
* Bool – логічний(так/ні)
* Char – символьний
* String – рядковий

1. Що таке операція? Які є стандартні арифметичні операції?

Операції це деякії дії над операндами(учасниками операції).

Операції діляться на бінарні:

* x + y -- додавання
* x – y -- віднімання
* x \* y -- множення
* x / y -- ділення
* x % y – остача від ділення

Унарні:

* x++ && ++x – постфіксний і префіксний інкремент
* x-- && --x – постфіксний і префіксний інкремент

1. Що виконує операція присвоєння?

Присвоює лівій змінні значення правої.

|  |
| --- |
| int x = 0;  int y = 10;  x = y;  //x = 10 |

1. Назвіть стандартні математичні функції мови прогрмування.

* Abs – модуль числа
* Sqrt – корінь квадратний
* Pow – піднесення в степінь
* Sin - синус
* Cos - косинус
* Max – максимальне значення
* Min – мінімальне значення
* Log - логарифм

1. За допомогою яких стандартних функцій можливий ввід-вивід даних за допомогою консолі?

|  |
| --- |
| string s = Console.ReadLine();//зчитування даних з рядка  Console.WriteLine(“Output”);//вивід даних з нового рядка  Console.Write(“Output”);//вивід даних з рядка |