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 №** 1

з дисципліни “Програмування”

тема “ C# .Net. Реалізація основних принципів ООП мовою C#”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) II курсу  групи КП-01  Грабовська Анастасія Дмитрівна  (*прізвище, ім’я, по батькові*) |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Заболотня Тетяна Миколаївна  (*прізвище, ім’я, по батькові*) |

Київ 2021

**Мета роботи**

Ознайомитися з основами об’єктно-орієнтованого підходу до створення ПЗ у мові С#, створенням класів, об’єктів, механізмами інкапсуляції, наслідування та поліморфізму. Вивчити механізм управління ресурсами, реалізований у .Net.

**Постановка задачі**

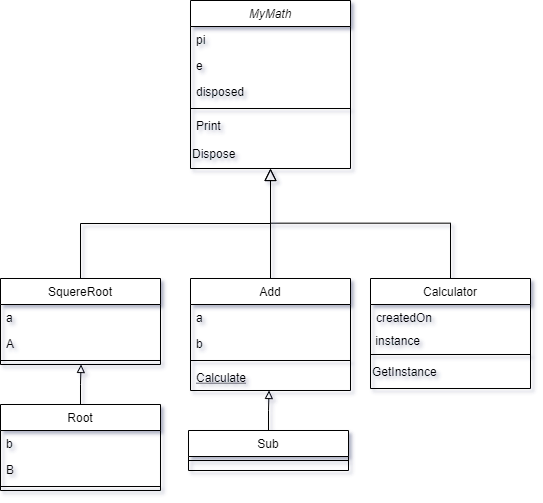
Побудувати ієрархію класів, що відтворюватимуть відношення наслідування між об’єктами реального світу (кількість класів >= 5). При цьому:

1. Забезпечити наявність у класах полів та методів з різними модифікаторами доступу, пояснити свій вибір **(1 бал)**.
2. Забезпечити наявність у класах властивостей: складніше, ніж просто get;set;, обгрунтувати доцільність створення властивості **(1 бал)**.
3. Створити для розроблюваних класів такі конструктори **(2 бали)**:
   * + конструктор за замовчанням;
     + конструктор з параметрами;
     + приватний конструктор;
     + статичний конструктор.

Продемонструвати, яким чином викликаються конструктори базового та дочірнього класів.

1. Використати віртуальні та перевизначені методи **(1 бал)**.
2. Додати до класів методи, наявність яких дозволятиме управляти знищенням екземплярів цих класів **(2 бали)**:
   1. реалізувати інтерфейс IDisposable;
   2. створити деструктори;
   3. забезпечити уникнення конфліктів між Dispose та деструктором.
3. Забезпечити виклики методів GC таким чином, щоб можна було простежити життєвий цикл об’єктів, що обробляються (зокрема, використати методи Collect, SupressFinalize, ReRegisterForFinalize, GetTotalMemory, GetGeneration, WaitForPendingFinalizers). Створити ситуацію, яка спровокує примусове збирання сміття GC **(2 бали)**.

**UML діаграма класів**



**Фрагменти коду:**

1.

|  |
| --- |
| class Calculator : MyMath      {          private static string createdOn;  public static Calculator GetInstance()          {              if (instance == null)              {                  instance = new Calculator();              }              return instance;          }  … |
| class Add : MyMath      {          protected int a;  private int Calculate()          {              return a + b;          }  ... |

2.

|  |
| --- |
| public double A          {              get              {                  if(this.a != default) return Math.Sqrt(a);                  else return 0;              }              set              {                  if (value > 0)                  {                      this.a = value;                  }              }          } |

3.

|  |
| --- |
| static MyMath()          {              pi = Math.PI;              e = Math.E;          } |
| private Calculator()          {              createdOn = DateTime.Now.ToLongTimeString();          } |
| public Add(int a, int b)          {              this.a = a;              this.b = b;          } |
| public Sub() : base()          {} |

4.

|  |
| --- |
| public virtual void Print()          {              WriteLine("I like to calculate.");          } |
| public override void Print()          {              base.Print();              WriteLine($"I'm created at {createdOn}.");          } |

5.

|  |
| --- |
| class MyMath : IDisposable      {          protected bool disposed;          ...          public virtual void Dispose()          {              GC.SuppressFinalize(this);              CleanUp(true);              GC.ReRegisterForFinalize(this);          }          void CleanUp(bool disposing)          {              if(!this.disposed)              {                  if (disposing)                  {                      WriteLine("Disposing managed resourses.");                  }                  WriteLine("Disposing unmanaged resourses.");                  disposed = true;              }          }          ~MyMath()          {              CleanUp(false);              WriteLine("MyMath destructed");          }      } |

6.

|  |
| --- |
| WriteLine("Memory before collect: " + GC.GetTotalMemory(false));  WriteLine("Generation: " + GC.GetGeneration(sub));  GC.Collect(2, GCCollectionMode.Forced);  GC.WaitForPendingFinalizers();  WriteLine("Memory after collect: " + GC.GetTotalMemory(false));  WriteLine("Generation: " + GC.GetGeneration(sub));  for(int i = 0; i < 50000; i++)  {  new Sub(i);  }  // Memory before collect: 105528  // Generation: 0  // Memory after collect: 97736  // Generation: 1  // Sub destructed  // Add destructed  // MyMath destructed |

**Висновки:**

Під час виконання лабораторної роботи я ознайомилася з основами об’єктно-орієнтованого підходу до створення ПЗ у мові С#, створенням класів, об’єктів, механізмами інкапсуляції, наслідування та поліморфізму. Також я вивчила механізм управління ресурсами, реалізований у .Net.