**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет   
імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗВІТ**

з лабораторної роботи № 10

**«Робота GC»**

**Варіант 5**

Дисципліна «Об'єктно-орієнтоване програмування»

Спеціальність: **Інженерія програмного забезпечення**

121-ЛР.09-208.22120802

***Студент\_\_Д. М.  Дзундза***

*(підпис)*

*\_\_11.01.2021* \_\_

*(дата)*

***Викладач \_\_\_\_\_\_\_\_С. Ю. Боровльова***

*(підпис)*

*\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(дата)*

Миколаїв – 2021

**Лабораторна робота №9**

**Завдання:**

1. Завдання: Використавши клас свого варіанту:

1. забезпечте можливість звільнення керованих та некерованих ресурсів відомими Вам способами,

2. використайте перевантажений варіант using ( для демонстрації обрати один із реальних бібліотечних класів, який реалізує відповідний інтерфейс(наприклад клас для роботи із файлами)),

3. Продемонструйте можливість негайного виклику GC,

4. Продемонструйте реалізація паттерну Dispose.

Варіант 5 – Журнал.

**Лістинг коду:**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Journal: IDisposable

{

static string size = "10x10";

string path;

bool disposed;

StreamReader file;

public Journal(string path)

{

file = new StreamReader(path);

}

~Journal()

{

Dispose(disposing: false);

}

public string GetText()

{

return file.ReadLine();

}

public void Dispose()

{

Dispose(true);

GC.SuppressFinalize(this); //we dont need to call it when cleaning up

}

protected virtual void Dispose(bool disposing)

{

if (!this.disposed)

{

if (disposing)

{

//managed

size = null;

}

//unmanaged resources

path = null;

file.Close();

file.Dispose();

disposed = true;

}

}

}

class Program

{

static void Main(string[] args)

{

Journal journal = new Journal("D:/Num.txt");

journal = null;

GC.Collect();

using (Journal journal2 = new Journal("D:/Num.txt"))

{

Console.WriteLine(journal2.GetText());

}

using (StreamReader file = new StreamReader("D:/Num.txt"))

{

int counter = 0;

string ln;

while ((ln = file.ReadLine()) != null)

{

Console.WriteLine(ln);

counter++;

}

file.Close();

Console.WriteLine($"File has {counter} lines.");

}

GC.Collect(1, GCCollectionMode.Forced);

Console.ReadKey();

}

}

}

**Питання до захисту:**

1. Призначення поняття поколінь при роботі GC. Скільки існує поколінь.

Покоління існують для оптимізації продуктивності збирача сміття.

0 - Це наймолодше покоління містить короткоживучі об'єкти. Прикладом короткоживучого об'єкта є тимчасова змінна. Складання сміття найчастіше виконується у цьому поколінні.

1 - об'єкти, що коротко живуть, і служить буфером між короткоживучими і довгоживучими об'єктами.

2 - довготривалі об'єкти

1. Чому треба уникати перевизначення методу Finalize.

Метод Finalize не можна перевантажити. Його перевантаження може завадити визову деструктора.

1. Інтерфейс IDisposable

Виконує завдання, пов'язані з видаленням, вивільненням або скиданням некерованих ресурсів, що визначаються додатком.

1. В чому різниця між детермінованим та недетермінованим звільненням ресурсів.

Різниця в тому які ресурси звільняються під час збирання сміття.

1. Призначення ключового слова using

Закриття та вивільнення місця з об’єкту після завершення його використання.

using (MyIDisposableObject obj = new MyIDisposableObject())

{

// use the object here

}

MyIDisposableObject obj;

try

{

obj = new MyIDisposableObject();

}

finally

{

if (obj != null)

{

((IDisposable)obj).Dispose();

}

}