**Державний вищий навчальний заклад  
Ужгородський національний університет  
Факультет інформаційних технологій**

ЛАБОРАТОРНА РОБОТА № 15  
**Тема:** Опрацювання виняткових ситуацій.

Виконав студент  
І курсу спеціальності «Інженерія програмного забезпечення»

Романюк Артем

**Ужгород-2025**

**Мета:** Ознайомлення з принципами роботи механізму обробки винятків.

**Вивчити питання:** Обробка виняткових ситуацій. Клас System.Exception. Застосування пари ключових слів try та catch. Блок finally. Типи винятків.

Завдання до роботи:

Модифікувати програму, побудовану в лабораторній роботі №13:

Нехай М – задана у програмі максимально можлива кількість записів (відповідно до умови індивідуального завдання). Користувач задає потрібну для заповнення кількість записів з консолі – N.

Додати клас для обробки винятку: число N, задане користувачем, виходить за межі інтервалу [0; M].

Оформити звіт про виконання роботи та завантажити його в системі електронного навчання ДВНЗ «УжНУ» в установлений термін.

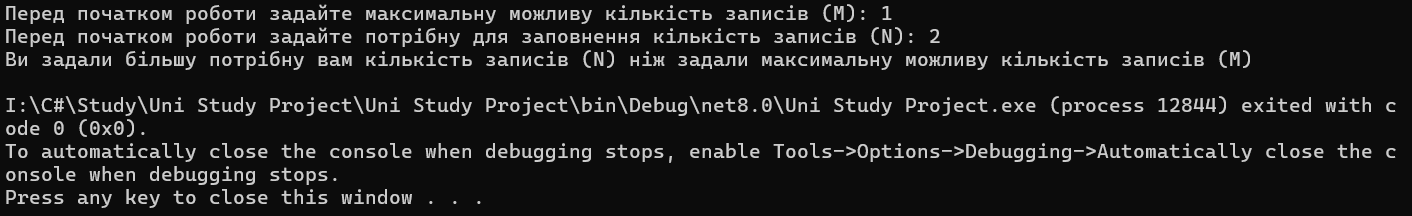
**Хід роботи:**

**Варіант 13**

1. Записати інформацію про товари, що експортуються: найменування товару, країна, що імпортує товар, об'єм партії (кількість товару). Знайти країни, в які експортується вибраний товар і загальний об'єм його експорту.

Реалізація в коді:

| using System; using System.Text;  namespace UniStudyProject {  struct Importer  {  public string name;  public int quantity;   public Importer(string name, int quantity)  {  this.name = name;  this.quantity = quantity;  }  }  struct ExportProduct  {  public string productName;  public Importer[] importers;   public ExportProduct()  {  this.productName = "Unknown";  this.importers = [];  }   public ExportProduct(string productName, Importer[] importers)  {  this.productName = productName;  this.importers = importers;  }   public void getImportersInfo()  {  if (this.productName == "Unknown")  {  Console.WriteLine("Product is undefined");  return;  }  Console.WriteLine(productName + " exports to:");  for (int i = 0; i < importers.Length; i++)  {  Console.WriteLine(importers[i].name + " in quantity of " + importers[i].quantity);  }  }  }  class InvalidRecordCountException : Exception { };  class Program  {  public static List<ExportProduct> exportProducts = new List<ExportProduct>();  public delegate void ExitHandler();  public static event ExitHandler Exit;  public static bool IsRunning = true;  static void Main(string[] args)  {  Console.InputEncoding = Encoding.UTF8;  Console.OutputEncoding = Encoding.UTF8;   Exit += GoodBye;  Exit += ExitLoop;  try  {  Console.Write("Перед початком роботи задайте максимальну можливу кількість записів (M): ");  int m = ValidatedInt(1);  Console.Write("Перед початком роботи задайте потрібну для заповнення кількість записів (N): ");  int n = ValidatedInt(1);  if (n > m) throw new InvalidRecordCountException();  } catch (InvalidRecordCountException e)  {  Console.WriteLine("Ви задали більшу потрібну вам кількість записів (N) ніж задали максимальну можливу кількість записів (M)");  return;  }    Console.WriteLine("\n--------------- Меню роботи з масивом структур експортованих продуктів ----------------------");  while (IsRunning)  {  Console.WriteLine("\n1. Уведення даних до масиву структур");  Console.WriteLine("2. Сортування масиву структур");  Console.WriteLine("3. Пошук у масиві структур за заданими параметрами");  Console.WriteLine("4. Внесення змін до структури");  Console.WriteLine("5. Вилучення структури з масиву");  Console.WriteLine("6. Вивід на екран даних масиву структур");  Console.WriteLine("Щоб вийти натисніть Ctrl+F3");  Console.Write("Виберіть фунцію (введіть номер): ");  try  {  ConsoleKeyInfo key = Console.ReadKey();  if (key.Key == ConsoleKey.F3 && key.Modifiers == ConsoleModifiers.Control)  {  Exit?.Invoke();  }  else  {  switch (key.KeyChar - '0')  {  case 1:  AddProduct();  break;   case 2:  SortProducts();  break;   case 3:  FindProduct();  break;   case 4:  EditProduct();  break;   case 5:  DeleteProduct();  break;   case 6:  ShowProducts();  break;   default:  Console.WriteLine("\nВведіть запропоноване з меню число!");  break;  } // switch  }  }  catch (FormatException e)  {  Console.WriteLine("\nВведіть ціле запропоноване з меню число!");  } // try/catch  } // Main loop  } //Main method   public static void AddProduct()  {  Console.Write("\nВведіть назву продукту: ");  string productName = Console.ReadLine();  Console.Write("Введіть кількість імпортерів: ");  int importersNumber = ValidatedInt(1);  Importer[] importers = new Importer[importersNumber];  for (int i = 0; i < importersNumber; i++)  {  Console.Write($"Введіть {i+1} імпортера: ");  string importerName = Console.ReadLine();  Console.Write($"Введіть кількість імпорту {i+1} імпортера: ");  int importerQuantity = ValidatedInt(1);  importers[i] = new Importer(importerName, importerQuantity);  }  exportProducts.Add(new ExportProduct(productName, importers));  Console.WriteLine($"\nУспішно додано новий товар {productName}");  }  public static void SortProducts()  {  if (exportProducts.Count == 0)  {  Console.WriteLine("\nТоварів немає. Масив пустий. Додайте якийсь товар");  return;  }  for (int i = 0; i < exportProducts.Count-1; i++)  {  for (int j = 0; j < exportProducts.Count-1-i; j++)  {  if (exportProducts[j].productName.ToLower().CompareTo(exportProducts[j+1].productName.ToLower()) > 0)  {  ExportProduct temp = exportProducts[j];  exportProducts[j] = exportProducts[j + 1];  exportProducts[j + 1] = temp;  }  }  }  Console.WriteLine("\nТовари успішно відсортовано в алфавітному порядку");  }  public static void FindProduct()  {  if (exportProducts.Count == 0)  {  Console.WriteLine("\nТоварів немає. Масив пустий. Додайте якийсь товар");  return;  }  Console.Write("\nВкажіть назву товару, який хочете знайти: ");  string input = Console.ReadLine();  foreach (ExportProduct product in exportProducts)  {  if (product.productName.ToLower() == input.ToLower())  {  Console.WriteLine("\nТовар знайдено!");  Console.WriteLine("Назва: " + product.productName);  Console.WriteLine("Імпортери:");  int j = 1;  foreach (Importer importer in product.importers)  {  Console.WriteLine($"{j++}. {importer.name} - {importer.quantity}");  }  return;  }  }  Console.WriteLine("\nТовар не знайдено");  }  public static void EditProduct()  {  if (exportProducts.Count == 0)  {  Console.WriteLine("\nТоварів немає. Масив пустий. Додайте якийсь товар");  return;  }  ShowProducts();  Console.Write("\nВведіть порядковий номер товару, який ви хочете редагувати: ");  int productNumber = 0;  do  {  try  {  productNumber = Convert.ToInt32(Console.ReadLine());  if (productNumber < 1 || productNumber > exportProducts.Count) throw new Exception();  }  catch (Exception e)  {  Console.WriteLine("\nВведіть існуючий номер товару!");  }  } while (productNumber < 1 || productNumber > exportProducts.Count);  ExportProduct modifiedProduct = exportProducts[productNumber - 1];  Console.Write("\nВведіть нову назву для товара: ");  string input = Console.ReadLine();  modifiedProduct.productName = input;  Console.Write("Введіть нову кількість імпортерів: ");  int importersNumber = ValidatedInt(1);  Importer[] importers = new Importer[importersNumber];  for (int i = 0; i < importersNumber; i++)  {  Console.Write($"Введіть {i + 1} імпортера: ");  string importerName = Console.ReadLine();  Console.Write($"Введіть кількість імпорту {i + 1} імпортера: ");  int importerQuantity = ValidatedInt(1);  importers[i] = new Importer(importerName, importerQuantity);  }  modifiedProduct.importers = importers;  exportProducts[productNumber - 1] = modifiedProduct;  Console.WriteLine("\nТовар успішно змінено!");  }  public static void DeleteProduct()  {  if (exportProducts.Count == 0)  {  Console.WriteLine("\nТоварів немає. Масив пустий. Додайте якийсь товар");  return;  }  ShowProducts();  Console.Write("\nВведіть порядковий номер товару, який ви хочете видалити: ");  int productNumber = 0;  do  {  try  {  productNumber = Convert.ToInt32(Console.ReadLine());  if (productNumber < 1 || productNumber > exportProducts.Count) throw new Exception();  }  catch (Exception e)  {  Console.WriteLine("\nВведіть існуючий номер товару!");  }  } while (productNumber < 1 || productNumber > exportProducts.Count);  exportProducts.RemoveAt(productNumber - 1);  Console.WriteLine("\nТовар успішно видалено!");  }  public static void ShowProducts()  {  if (exportProducts.Count == 0)  {  Console.WriteLine("\nТоварів немає. Масив пустий. Додайте якийсь товар");  return;  }  Console.WriteLine("\n---------- Список товарів ---------------");  int i = 1;  foreach (ExportProduct product in exportProducts)  {  Console.WriteLine($"\n{i++}. Продукт: " + product.productName);  Console.WriteLine("Імпортери:");  int j = 1;  foreach (Importer importer in product.importers)  {  Console.WriteLine($"{j++}. {importer.name} - {importer.quantity}");  }  }  }  public static void GoodBye()  {  Console.WriteLine("\nЗавершення програми. Дякую за користування!");  }  public static void ExitLoop()  {  IsRunning = false;  }  public static int ValidatedInt(int fromNumber)  {  int number = 0;  do  {  try  {  number = Convert.ToInt32(Console.ReadLine());  if (number < fromNumber) throw new Exception();  }  catch (Exception e)  {  Console.WriteLine($"Введіть ціле число більше або рівне {fromNumber}!");  }  } while (number < fromNumber);  return number;  }  } } |
| --- |



**Висновок:**

У ході виконання лабораторної роботи було вивчено основи обробки виняткових ситуацій у мові програмування C#. Було розглянуто механізми використання конструкцій try-catch-finally та створення власного класу винятків.

У рамках завдання було модифіковано програму з попередньої лабораторної роботи, додавши перевірку на допустимість введеної кількості записів та виняткову ситуацію у разі її перевищення. Це підвищило надійність і стабільність роботи програми.

Також було реалізовано функціонал для роботи з масивом структур, що дозволяє додавати, сортувати, редагувати, шукати та видаляти записи про товари, що експортуються. Використання виняткових ситуацій дозволило уникнути некоректного введення та потенційних помилок під час виконання програми.

Загалом, робота дала змогу закріпити знання з обробки виняткових ситуацій та їхнього практичного застосування в розробці програмного забезпечення.