МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра КІТАМ

Звіт

з Лабораторної роботі №3

з дисципліни «Програмні засоби інтегрованих систем у виробництві»

на тему «ДОСЛІДЖЕННЯ ОСНОВНИХ МЕТОДІВ СТВОРЕННЯ МЕНЮ КОРИСТУВАЧА»

Виконав:
ст. гр. АКТАКІТ-21-2
Махов Д.Р.

Перевірив:
Асистент
Теслюк.С.І

Харків 2023

3 ДОСЛІДЖЕННЯ ОСНОВНИХ МЕТОДІВ СТВОРЕННЯ МЕНЮ КОРИСТУВАЧА

3.1 Мета роботи

Вивчити основні функції для роботи з консоллю та клавіатурою. Вивчення основних методів створення інтерфейсу користувача.

3.2 Стислі теоретичні відомості

ArrayList – неоднорідний масив змінного розміру.

Незважаючи на те, що звичайні масиви, безумовно, знаходять широке застосування, вони накладають певні обмеження на програміста, оскільки кожний масив може зберігати дані тільки одного типу (однорідність) і до того ж перед використанням масиву ви повинні виділити (allocate) певну кількість елементів. Однак, часто розробникам хочеться чогось більш гнучкого – просту колекцію об’єктів потенційно різного типу, з якими можна було б просто працювати, не непокоячись про питання, пов’язані з виділенням елементів (allocation). Базова бібліотека класів .NET Framework містить таку структуру даних. Вона називається System.Collections.ArrayList.

У фрагменті коду, наведеному нижче, демонструється ArrayList в дії. Зазначте, що якщо ми використовуємо ArrayList, то ми можемо додати до нього будь-який тип, причому ніяких дій з виділення пам'яті виконувати не потрібно. На ділі виходить, що до ArrayList можуть бути додані елементи будь-якого типу. Для того ж, нам більше не потрібно піклуватися про зміну розміру ArrayList. Всі ці дії проводяться для нас за лаштунками.

```
ArrayListcountDown = newArrayList();

countDown.Add(4);

countDown.Add(3);

countDown.Add(2);

countDown.Add(1);

countDown.Add("blast off!");

countDown.Add(new ArrayList());
```

ArrayList використовує System.Array типу object. Оскільки всі типи є прямими або непрямими послідовниками object, масив екземплярів типу object може містити елементи довільного типу. За замовчуванням ArrayList створює масив з 16 елементів типу object, хоча точний розмір може бути вказаний через параметр конструктора у властивості Capacity. Під час додавання елемента за допомогою методу Add () кількість елементів у внутрішньому масиві порівнюється з розміром масиву. Якщо операція додавання елемента викликає перевищення кількості елементів масиву над його розміром, то розмір автоматично подвоюється (redimensioned).

ArrayList, як і звичайний масив, використовує індекс для доступу до своїх елементів за допомогою аналогічного синтаксису:

```
// Читання елементу

int x = (int) countDown[0];

string y = (string) countDown[5];

// Запис даних

countDown[1] = 5;

// ** Буде згенеровано ВИКЛЮЧЕННЯ rgumentOutOfRange **

countDown[7] = 5;
```

Оскільки ArrayList зберігає масив об'єктів, ви повинні таким чином привести (explicitly cast) зчитане з ArrayList значення до того типу даних, який зберігався в даному елементі ArrayList-а. Також зауважте, якщо ви спробуєте послатися на елемент ArrayList-а номер, якого більше, ніж розмір ArrayList-а, буде згенеровано виняток System.ArgumentOutOfRange.

3.3 Хід роботи

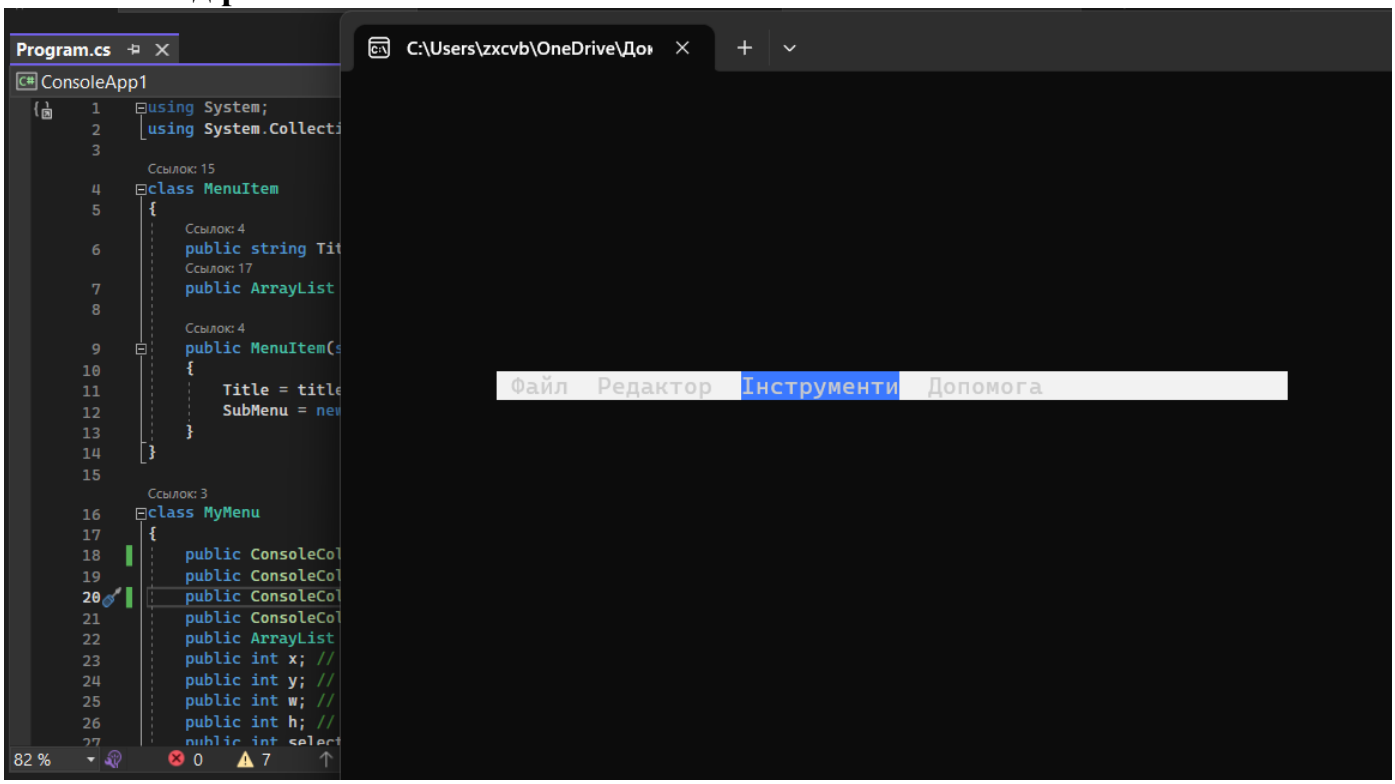

Рисунок 3.1-Приклад роботи програми

Рисунок 3.2-Приклад роботи випадаючого меню

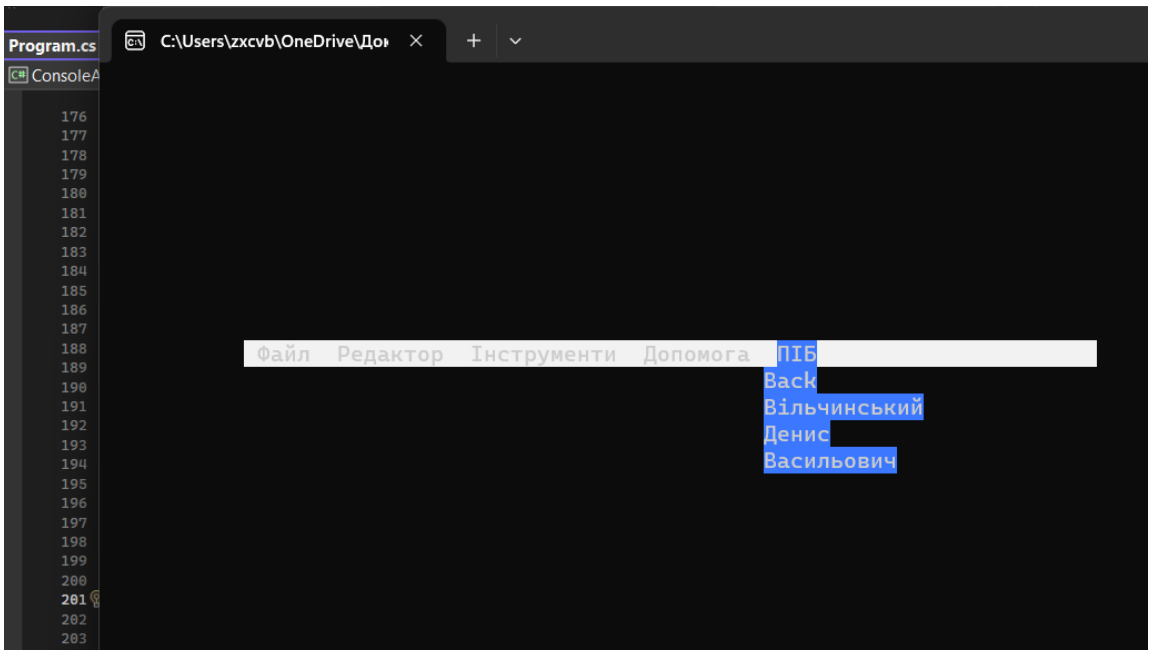

Рисунок 3.3-Приклад з ПІБ

```
class MyMenu
{

   public ConsoleColor TextColor = ConsoleColor.Blue;
   public ConsoleColor FonColor = ConsoleColor.White;
   public ConsoleColor MenuColor = ConsoleColor.White;
   public ConsoleColor ItemsColor = ConsoleColor.Blue;
   public ArrayList Items = new ArrayList();
   public int x; // початкова позиція меню
   public int y; // початкова позиція меню
   public int w; // ширина меню
   public int h; // висота меню
   public int selectedIndex; // обраний елемент меню
   public int subMenuIndex; // обраний підменю
   public bool isSubMenuVisible; // прапорець, що вказує, чи видиме підменю
   public ArrayList currentMenu; // поточне меню

   public MyMenu(ArrayList items, int xPos, int yPos)
   {
      Items = items;
      x = xPos;
      y = yPos;
      h = 1;

      // визначаємо ширину меню
      w = 0;
      for (int i = 0; i < Items.Count; i++)
      {
         MenuItem menuItem = (MenuItem)Items[i];
         w += menuItem.Title.Length + 6;
         menuItem.SubMenu.Insert(0, "Back");
         if (menuItem.SubMenu.Count > h)
            h = menuItem.SubMenu.Count;
      }

...

// малюємо опції
   public void DrawItemsMenu()
```

```
  {
    int startX = x;
    for (int i = 0; i < Items.Count; i++)
    {
      MenuItem menuItem = (MenuItem)Items[i];
      Console.BackgroundColor = FonColor;
      if (i == selectedIndex)
      {
        Console.BackgroundColor = ItemsColor;
        if (isSubMenuVisible)
        {
          // малюємо підменю
          for (int j = 0; j < menuItem.SubMenu.Count; j++)
          {
            Console.SetCursorPosition(startX, y + j + 1);
            Console.Write(menuItem.SubMenu[j]);
          }
        }
      }
      Console.SetCursorPosition(startX + 1, y);
      Console.Write(menuItem.Title);
      startX += menuItem.Title.Length + 2;
    }
    Console.BackgroundColor = MenuColor;
  }

  // скануємо клавіші та переміщуємо курсор

  public void ScanMenu()
  {
    ConsoleKeyInfo key;
    do
    {
      DrawBackGroundMenu();
      DrawItemsMenu();

      key = Console.ReadKey(true);
      if (key.Key == ConsoleKey.RightArrow)
      {
        selectedIndex++;
        if (selectedIndex >= Items.Count)
        {
          selectedIndex = 0;
        }
        isSubMenuVisible = false;
        currentMenu = Items;

...

  static void Main(string[] args)
  {
    ArrayList menuItems = new ArrayList();
    MenuItem fileItem = new MenuItem("Файл");
    fileItem.SubMenu.Add("Відкррити");
    fileItem.SubMenu.Add("Зберехти");
    menuItems.Add(fileItem);
```

3.4 Контрольні запитання

1. Що таке структура даних ArrayList
5. Що таке „Динамічний масив"?

1. ArrayList є, що дозволяє створювати динамічні колекції об'єктів будь-якого типу. Вона автоматично збільшує або зменшує свій розмір та надає доступ до елементів за до допомогою індексів.

5. Динамічний масив це масив розміри якого змінюється по ходу роботи програми на відміну від статичних масивів розмір якого задається перед початком роботи коду і не змінюється на протязі всієї роботи. З плюсів такого програми можливо більш економічно працювати з пам'ятью з мінусів потребується більша продуктивність процесору.

Висновки

На цій лабораторній роботі було вивчено основні функції для роботи з консолью та клавіатурою. Було вивчено основні методи створення інтерфейсу користувача.