**Лабораторна робота №2**

## Тема: Перевантаження операцій.

**Мета***.* ***Одержати практичні навички створення абстрактних типів даних і перевантаження операцій у мові С++.***

Завдання 1.

#pragma once

#include <iostream>

#include <string>

using namespace std;

class IntMatrix

{

private:

int\*\* IntArray;

int n;

public:

IntMatrix(int size);

IntMatrix(const IntMatrix& other);

IntMatrix& operator=(const IntMatrix& other);

~IntMatrix();

void input();

void output() const;

int sumOfColumn(int col) const;

int getZeroCount() const;

void setMainDiagonal(int value);

IntMatrix& operator++();

IntMatrix operator++(int);

IntMatrix& operator--();

IntMatrix operator--(int);

operator bool() const;

IntMatrix operator+(const IntMatrix& other) const;

operator string() const;

static IntMatrix fromString(const string& str);

};

#include "IntMatrix.h"

#include <iomanip>

#include <sstream>

IntMatrix::IntMatrix(int size) : n(size)

{

IntArray = new int\* [n];

for (int i = 0; i < n; ++i)

IntArray[i] = new int[n] {};

}

IntMatrix::IntMatrix(const IntMatrix& other) : n(other.n)

{

IntArray = new int\* [n];

for (int i = 0; i < n; ++i) {

IntArray[i] = new int[n];

for (int j = 0; j < n; ++j)

IntArray[i][j] = other.IntArray[i][j];

}

}

IntMatrix& IntMatrix::operator=(const IntMatrix& other)

{

if (this == &other) return \*this;

for (int i = 0; i < n; ++i) delete[] IntArray[i];

delete[] IntArray;

n = other.n;

IntArray = new int\* [n];

for (int i = 0; i < n; ++i) {

IntArray[i] = new int[n];

for (int j = 0; j < n; ++j)

IntArray[i][j] = other.IntArray[i][j];

}

return \*this;

}

IntMatrix::~IntMatrix()

{

for (int i = 0; i < n; ++i)

delete[] IntArray[i];

delete[] IntArray;

}

void IntMatrix::input()

{

cout << "Enter elements of the matrix (" << n << "x" << n << "):\n";

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j) {

cout << "Element [" << i << "][" << j << "]: ";

cin >> IntArray[i][j];

}

}

void IntMatrix::output() const

{

cout << "Matrix elements:\n";

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j)

cout << setw(5) << IntArray[i][j];

cout << endl;

}

}

int IntMatrix::sumOfColumn(int col) const

{

int sum = 0;

for (int i = 0; i < n; ++i)

sum += IntArray[i][col];

return sum;

}

int IntMatrix::getZeroCount() const

{

int count = 0;

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j)

if (IntArray[i][j] == 0)

++count;

return count;

}

void IntMatrix::setMainDiagonal(int value)

{

for (int i = 0; i < n; ++i)

IntArray[i][i] = value;

}

IntMatrix& IntMatrix::operator++()

{

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j)

++IntArray[i][j];

return \*this;

}

IntMatrix IntMatrix::operator++(int)

{

IntMatrix temp(\*this);

++(\*this);

return temp;

}

IntMatrix& IntMatrix::operator--()

{

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j)

--IntArray[i][j];

return \*this;

}

IntMatrix IntMatrix::operator--(int)

{

IntMatrix temp(\*this);

--(\*this);

return temp;

}

IntMatrix::operator bool() const

{

return n > 0;

}

IntMatrix IntMatrix::operator+(const IntMatrix& other) const

{

if (n != other.n) {

throw invalid\_argument("Matrix sizes do not match for addition");

}

IntMatrix result(n);

for (int i = 0; i < n; ++i)

for (int j = 0; j < n; ++j)

result.IntArray[i][j] = IntArray[i][j] + other.IntArray[i][j];

return result;

}

IntMatrix::operator string() const

{

ostringstream oss;

oss << n << "\n";

for (int i = 0; i < n; ++i) {

for (int j = 0; j < n; ++j)

oss << IntArray[i][j] << " ";

oss << "\n";

}

return oss.str();

}

IntMatrix IntMatrix::fromString(const string& str)

{

istringstream iss(str);

int size;

iss >> size;

IntMatrix m(size);

for (int i = 0; i < size; ++i)

for (int j = 0; j < size; ++j)

iss >> m.IntArray[i][j];

return m;

}

#include <iostream>

#include "IntMatrix.h"

using namespace std;

int main()

{

IntMatrix m1(2), m2(2);

cout << "Fill matrix m1:\n";

m1.input();

cout << "Fill matrix m2:\n";

m2.input();

cout << "\nMatrix m1:\n"; m1.output();

cout << "\nMatrix m2:\n"; m2.output();

cout << "\nApplying ++ to m1:\n";

++m1;

m1.output();

cout << "\nApplying -- to m2:\n";

m2--;

m2.output();

cout << "\nMatrix m1 + m2:\n";

IntMatrix sum = m1 + m2;

sum.output();

if (m1) cout << "\nm1 is square!\n";

string str = string(m1);

cout << "\nMatrix m1 as string:\n" << str << endl;

IntMatrix m3 = IntMatrix::fromString(str);

cout << "\nMatrix m3 (restored from string):\n";

m3.output();

return 0;

}

Зображення, що містить текст, знімок екрана, програмне забезпечення, Мультимедійне програмне забезпечення

Вміст на основі ШІ може бути неправильним.

Зображення, що містить текст, знімок екрана, програмне забезпечення

Вміст на основі ШІ може бути неправильним.

Завдання 2.

**using System;**

**using System.Globalization;**

**using System.IO;**

**using System.Windows.Forms;**

**namespace OOP\_C\_\_lab2**

**{**

**public partial class Form1 : Form**

**{**

**public Form1()**

**{**

**InitializeComponent();**

**textBox1.MaxLength = 8;**

**textBox1.KeyPress += Article\_KeyPress;**

**textBox3.KeyPress += Brand\_KeyPress;**

**textBox4.MaxLength = 10;**

**textBox4.KeyPress += Price\_KeyPress;**

**textBox5.KeyPress += Year\_KeyPress;**

**tabControl1.SelectedIndexChanged += TabControl1\_SelectedIndexChanged;**

**button2.Click += Button2\_Click;**

**}**

**private void Form1\_Load(object sender, EventArgs e)**

**{**

**ShowHideArithmetic(false);**

**}**

**class Product**

**{**

**public string Article { get; set; }**

**public string Name { get; set; }**

**public string Brand { get; set; }**

**public double Price { get; set; }**

**public int Year { get; set; }**

**public Product() { }**

**public Product(string article, string name, string brand, double price, int year)**

**{**

**Article = article;**

**Name = name;**

**Brand = brand;**

**Price = price;**

**Year = year;**

**}**

**public static Product operator +(Product p, double value)**

**{**

**return new Product(p.Article, p.Name, p.Brand, p.Price + value, p.Year);**

**}**

**public static Product operator -(Product p, double value)**

**{**

**return new Product(p.Article, p.Name, p.Brand, p.Price - value, p.Year);**

**}**

**public static Product operator \*(Product p, double value)**

**{**

**return new Product(p.Article, p.Name, p.Brand, p.Price \* value, p.Year);**

**}**

**public static Product operator /(Product p, double value)**

**{**

**return new Product(p.Article, p.Name, p.Brand, p.Price / value, p.Year);**

**}**

**public static implicit operator bool(Product p) => p.Price > 0;**

**public static implicit operator string(Product p) => p.Info();**

**public string Info() =>**

**$"Артикул: {Article}, Назва: {Name}, Бренд: {Brand}, Ціна: {Price} грн, Рік: {Year}";**

**}**

**private void Article\_KeyPress(object sender, KeyPressEventArgs e)**

**{**

**TextBox tb = sender as TextBox;**

**if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))**

**e.Handled = true;**

**if (!char.IsControl(e.KeyChar) && tb.Text.Length >= 8)**

**e.Handled = true;**

**}**

**private void Brand\_KeyPress(object sender, KeyPressEventArgs e)**

**{**

**if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar) && e.KeyChar != ' ')**

**e.Handled = true;**

**}**

**private void Price\_KeyPress(object sender, KeyPressEventArgs e)**

**{**

**TextBox tb = sender as TextBox;**

**if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar) && e.KeyChar != '.' && e.KeyChar != ',')**

**e.Handled = true;**

**if ((e.KeyChar == '.' || e.KeyChar == ',') && (tb.Text.Contains(".") || tb.Text.Contains(",")))**

**e.Handled = true;**

**if (!char.IsControl(e.KeyChar) && tb.Text.Length >= 10)**

**e.Handled = true;**

**}**

**private void Year\_KeyPress(object sender, KeyPressEventArgs e)**

**{**

**if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))**

**e.Handled = true;**

**}**

**private void TabControl1\_SelectedIndexChanged(object sender, EventArgs e)**

**{**

**if (tabControl1.SelectedTab.Text == "обрахувати")**

**{**

**ShowHideArithmetic(true);**

**ShowHideMainControls(false);**

**}**

**else**

**{**

**ShowHideArithmetic(false);**

**ShowHideMainControls(true);**

**}**

**}**

**private void ShowHideArithmetic(bool show)**

**{**

**label6.Visible = show;**

**label7.Visible = show;**

**label8.Visible = show;**

**label9.Visible = show;**

**textBox6.Visible = show;**

**textBox7.Visible = show;**

**textBox8.Visible = show;**

**textBox9.Visible = show;**

**button2.Visible = show;**

**}**

**private void ShowHideMainControls(bool show)**

**{**

**label1.Visible = show;**

**label2.Visible = show;**

**label3.Visible = show;**

**label4.Visible = show;**

**label5.Visible = show;**

**textBox1.Visible = show;**

**textBox2.Visible = show;**

**textBox3.Visible = show;**

**textBox4.Visible = show;**

**textBox5.Visible = show;**

**button1.Visible = show;**

**}**

**private void button1\_Click(object sender, EventArgs e)**

**{**

**string article = textBox1.Text.Trim();**

**string name = textBox2.Text.Trim();**

**string brand = textBox3.Text.Trim();**

**string priceStr = textBox4.Text.Trim().Replace(',', '.');**

**string yearStr = textBox5.Text.Trim();**

**if (tabControl1.SelectedTab.Text == "додавання")**

**{**

**if (string.IsNullOrWhiteSpace(article) ||**

**string.IsNullOrWhiteSpace(name) ||**

**string.IsNullOrWhiteSpace(brand) ||**

**string.IsNullOrWhiteSpace(priceStr) ||**

**string.IsNullOrWhiteSpace(yearStr))**

**{**

**MessageBox.Show("Заповніть усі поля!", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**return;**

**}**

**if (!double.TryParse(priceStr, NumberStyles.Any, CultureInfo.InvariantCulture, out double price))**

**{**

**MessageBox.Show("Введіть правильну ціну!", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**return;**

**}**

**if (!int.TryParse(yearStr, out int year) || year < 0 || year > 2025)**

**{**

**MessageBox.Show("Рік випуску має бути від 0 до 2025!", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**return;**

**}**

**Product p = new Product(article, name, brand, price, year);**

**dataGridView1.Rows.Add(p.Article, p.Name, p.Brand, p.Price, p.Year);**

**}**

**else if (tabControl1.SelectedTab.Text == "Видалення")**

**{**

**bool deleted = false;**

**foreach (DataGridViewRow row in dataGridView1.Rows)**

**{**

**if (!row.IsNewRow && (row.Cells[0].Value?.ToString() == article || row.Cells[1].Value?.ToString() == name))**

**{**

**dataGridView1.Rows.Remove(row);**

**deleted = true;**

**break;**

**}**

**}**

**if (deleted)**

**{**

**foreach (DataGridViewRow row in dataGridView2.Rows)**

**{**

**if (!row.IsNewRow && (row.Cells[0].Value?.ToString() == article || row.Cells[1].Value?.ToString() == name))**

**{**

**dataGridView2.Rows.Remove(row);**

**break;**

**}**

**}**

**}**

**else**

**{**

**MessageBox.Show("Такого товару немає у списку!", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**}**

**}**

**else if (tabControl1.SelectedTab.Text == "Пошук")**

**{**

**dataGridView2.Rows.Clear();**

**bool found = false;**

**foreach (DataGridViewRow row in dataGridView1.Rows)**

**{**

**if (row.IsNewRow) continue;**

**string rowArticle = row.Cells[0].Value?.ToString() ?? "";**

**string rowName = row.Cells[1].Value?.ToString() ?? "";**

**bool match = true;**

**if (!string.IsNullOrWhiteSpace(article) && !rowArticle.StartsWith(article))**

**match = false;**

**if (!string.IsNullOrWhiteSpace(name) && !rowName.Contains(name, StringComparison.OrdinalIgnoreCase))**

**match = false;**

**if (match)**

**{**

**dataGridView2.Rows.Add(**

**row.Cells[0].Value,**

**row.Cells[1].Value,**

**row.Cells[2].Value,**

**row.Cells[3].Value,**

**row.Cells[4].Value**

**);**

**found = true;**

**}**

**}**

**}**

**textBox1.Clear();**

**textBox2.Clear();**

**textBox3.Clear();**

**textBox4.Clear();**

**textBox5.Clear();**

**}**

**private void Button2\_Click(object sender, EventArgs e)**

**{**

**if (dataGridView1.SelectedRows.Count == 0 && dataGridView2.SelectedRows.Count == 0)**

**return;**

**void ApplyOperation(DataGridViewRow row, double value, Func<Product, double, Product> operation)**

**{**

**if (row.IsNewRow) return;**

**double oldPrice = Convert.ToDouble(row.Cells[3].Value);**

**Product p = new Product(**

**row.Cells[0].Value.ToString(),**

**row.Cells[1].Value.ToString(),**

**row.Cells[2].Value.ToString(),**

**oldPrice,**

**Convert.ToInt32(row.Cells[4].Value)**

**);**

**Product newP = operation(p, value);**

**row.DataGridView.Rows.Add(newP.Article, newP.Name, newP.Brand, newP.Price, newP.Year);**

**}**

**if (double.TryParse(textBox6.Text.Replace(',', '.'), NumberStyles.Any, CultureInfo.InvariantCulture, out double valPlus))**

**{**

**foreach (DataGridViewRow row in dataGridView1.SelectedRows)**

**ApplyOperation(row, valPlus, (p, v) => p + v);**

**foreach (DataGridViewRow row in dataGridView2.SelectedRows)**

**ApplyOperation(row, valPlus, (p, v) => p + v);**

**}**

**if (double.TryParse(textBox7.Text.Replace(',', '.'), NumberStyles.Any, CultureInfo.InvariantCulture, out double valMinus))**

**{**

**foreach (DataGridViewRow row in dataGridView1.SelectedRows)**

**ApplyOperation(row, valMinus, (p, v) => p - v);**

**foreach (DataGridViewRow row in dataGridView2.SelectedRows)**

**ApplyOperation(row, valMinus, (p, v) => p - v);**

**}**

**if (double.TryParse(textBox8.Text.Replace(',', '.'), NumberStyles.Any, CultureInfo.InvariantCulture, out double valMul))**

**{**

**foreach (DataGridViewRow row in dataGridView1.SelectedRows)**

**ApplyOperation(row, valMul, (p, v) => p \* v);**

**foreach (DataGridViewRow row in dataGridView2.SelectedRows)**

**ApplyOperation(row, valMul, (p, v) => p \* v);**

**}**

**if (double.TryParse(textBox9.Text.Replace(',', '.'), NumberStyles.Any, CultureInfo.InvariantCulture, out double valDiv))**

**{**

**foreach (DataGridViewRow row in dataGridView1.SelectedRows)**

**ApplyOperation(row, valDiv, (p, v) => p / v);**

**foreach (DataGridViewRow row in dataGridView2.SelectedRows)**

**ApplyOperation(row, valDiv, (p, v) => p / v);**

**}**

**textBox6.Clear();**

**textBox7.Clear();**

**textBox8.Clear();**

**textBox9.Clear();**

**}**

**private void зберегтиToolStripMenuItem\_Click(object sender, EventArgs e)**

**{**

**SaveFileDialog sfd = new SaveFileDialog { Filter = "Text Files|\*.txt|All Files|\*.\*" };**

**if (sfd.ShowDialog() == DialogResult.OK)**

**{**

**using (StreamWriter sw = new StreamWriter(sfd.FileName))**

**{**

**foreach (DataGridViewRow row in dataGridView1.Rows)**

**{**

**if (!row.IsNewRow)**

**{**

**sw.WriteLine($"{row.Cells[0].Value};{row.Cells[1].Value};{row.Cells[2].Value};{row.Cells[3].Value};{row.Cells[4].Value}");**

**}**

**}**

**}**

**}**

**}**

**private void відкритиToolStripMenuItem\_Click(object sender, EventArgs e)**

**{**

**OpenFileDialog ofd = new OpenFileDialog { Filter = "Text Files|\*.txt|All Files|\*.\*" };**

**if (ofd.ShowDialog() == DialogResult.OK)**

**{**

**dataGridView1.Rows.Clear();**

**string[] lines = File.ReadAllLines(ofd.FileName);**

**foreach (string line in lines)**

**{**

**string[] parts = line.Split(';');**

**if (parts.Length == 5)**

**{**

**dataGridView1.Rows.Add(parts[0], parts[1], parts[2], parts[3], parts[4]);**

**}**

**}**

**}**

**}**

**}**

**}**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Вміст на основі ШІ може бути неправильним.**

**Зображення, що містить текст, знімок екрана, число, схема

Вміст на основі ШІ може бути неправильним.**

**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Вміст на основі ШІ може бути неправильним.**

**Зображення, що містить текст, знімок екрана, схема, число

Вміст на основі ШІ може бути неправильним.**

**Висновки:** У процесі виконання роботи було набуте практичне вміння створювати абстрактні типи даних у мові C++ та застосовувати механізм перевантаження операторів.