**Практична робота №1**

***з дисципліни “Об’єктно-орієнтоване програмування”***

***Мета:*** Одержати практичні навички реалізації класів на С++.

**Задача 1 *Опису членів-даних класів користувача***

|  |
| --- |
| **5.БІБЛІОТЕКА** |
| **ім'я- char\*** |
| **автор – char\*** |
| **вартість- float** |

**Лістинг програми**

**Book.h**

#pragma once

#include <iostream>

#include <string>

using namespace std;

class Book {

private:

string name;

string author;

float price;

public:

Book();

Book(const string& n, const string& a, float p);

~Book();

void setBook(const string& n, const string& a, float p);

void print() const;

string getName() const;

string getAuthor() const;

float getPrice() const;

};

**Book.cpp**

#include "book.h"

Book::Book() : name(""), author(""), price(0.0f) {

cout << "Створено книгу (порожню)" << endl;

}

Book::Book(const string& n, const string& a, float p) : name(n), author(a), price(p) {

cout << "Створено книгу: " << name << endl;

}

Book::~Book() {

cout << "Видалено книгу: " << name << endl;

}

void Book::setBook(const string& n, const string& a, float p) {

name = n;

author = a;

price = p;

}

void Book::print() const {

cout << "Назва: " << name

<< "\tАвтор: " << author

<< "\tЦiна: " << price << " грн" << endl;

}

string Book::getName() const { return name; }

string Book::getAuthor() const { return author; }

float Book::getPrice() const { return price; }

**lr1 1.cpp**

#include "book.h"

#include <vector>

int main() {

setlocale(0, "ukr");

Book b1;

Book b2("Кобзар", "Тарас Шевченко", 300.0f);

b1.setBook("Мистецтво вiйни", "Сунь-Цзи", 250.5f);

cout << endl;

cout << "\nОкремi книги:\n";

b1.print();

b2.print();

cout << endl;

vector<Book> library;

library.emplace\_back("Хатина дядька Тома", "Гаррiєт Бiчер Стоу", 120.0f);

library.emplace\_back("Майстер i Маргарита", "Михайло Булгаков", 220.0f);

cout << endl;

cout << "\nСписок книг:\n";

for (size\_t i = 0; i < library.size(); ++i) {

cout << i + 1 << ". ";

library[i].print();

}

cout << endl;

return 0;

}

**Результати роботи програми**

Зображення, що містить текст, знімок екрана, Шрифт

Вміст на основі ШІ може бути неправильним.

**Задача 2**

**5. Створити клас для роботи з одновимірним масивом цілих чисел. Розробити наступні елементи класу:**

* 1. **Поля:** 
     1. int [] IntArray;
     2. int n.
  2. **Конструктор, що дозволяє створити масив розмірності n.**
  3. **Методи, що дозволяють:** 
     1. ввести елементи масиву з клавіатури;
     2. вивести елементи масиву на екран;
     3. відсортувати елементи масиву в порядку зростання.
  4. **Властивості:** 
     1. повертаюче розмірність масиву (доступне тільки для читання);
     2. які надають можливість домножить всі елементи масиву на скаляр (доступне тільки для запису).

**Лістинг програми**

**Arr.h**

#pragma once

class myarr {

public:

int\* arr;

int n;

public:

myarr(int size);

~myarr();

void vvid();

void vuvid();

void sort();

int size();

void scalar(int k);

};

**Arr.cpp**

#include "arr.h"

#include <iostream>

using namespace std;

myarr::myarr(int size)

{

n = size;

arr = new int[n];

}

myarr::~myarr()

{

delete[] arr;

}

void myarr::vvid()

{

cout << "Введіть " << n << " елементів масиву:\n";

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

cout << "Елемент [" << i << "]: ";

cin >> arr[i];

}

}

void myarr::vuvid()

{

cout << "Масив: ";

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

cout << arr[i] << " ";

}

cout << endl;

}

void myarr::sort()

{

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

}

int myarr::size()

{

return n;

}

void myarr::scalar(int k)

{

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

arr[i] \*= k;

}

}

**Lr1 2**

#include <iostream>

#include "arr.h"

using namespace std;

int main()

{ setlocale(0, "ukr");

int size;

cout << "Введіть розмір масиву: ";

cin >> size;

myarr arr(size);

arr.vvid();

arr.vuvid();

arr.sort();

cout << "Після сортування: ";

arr.vuvid();

cout << "Розмір масиву: " << arr.size() << endl;

int k;

cout << "Введіть число, на яке домножити масив: ";

cin >> k;

arr.scalar(k);

cout << "Після множення: ";

arr.vuvid();

return 0;

}

**Результати роботи програми**

**Зображення, що містить текст, знімок екрана, монітор, Шрифт

Вміст на основі ШІ може бути неправильним.**

**Задача 3**

Зробити програму на мові С# «Готель», де вионуються наступні дії:

* Запис кімнат та їх характеристики
* Збереження записів
* Вивід їх на екран
* Змінення записів
* Пошук вільних кімнат

**Лістинг програми**

namespace lr\_1\_3

{

public partial class Form1 : Form

{

public class Room

{

public int Місця { get; set; }

public double Площа { get; set; }

public int КількістьЛіжок { get; set; }

public string ВидЛіжка { get; set; }

public string Меблі { get; set; }

public bool WiFi { get; set; }

public string Клімат { get; set; }

public bool Харчування { get; set; }

public string Санвузол { get; set; }

public decimal Ціна { get; set; }

public Room(int місця, double площа, int ліжка, string видЛіжка,

string меблі, bool wifi, string клімат,

bool харчування, string санвузол, decimal ціна)

{

Місця = місця;

Площа = площа;

КількістьЛіжок = ліжка;

ВидЛіжка = видЛіжка;

Меблі = меблі;

WiFi = wifi;

Клімат = клімат;

Харчування = харчування;

Санвузол = санвузол;

Ціна = ціна;

}

}

private List<Room> rooms = new List<Room>();

public Form1()

{

InitializeComponent();

panel1.Hide();

label28.Hide();

dataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dataGridView2.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dataGridView1.Columns.Add("Місця", "К-сть місць");

dataGridView1.Columns.Add("Площа", "Площа (м²)");

dataGridView1.Columns.Add("Ліжко", "К-сть ліжок");

dataGridView1.Columns.Add("ВидЛіжка", "Вид ліжка");

dataGridView1.Columns.Add("Меблі", "Меблі / Техніка");

dataGridView1.Columns.Add("WiFi", "Wi-Fi");

dataGridView1.Columns.Add("Клімат", "Клімат");

dataGridView1.Columns.Add("Харчування", "Харчування");

dataGridView1.Columns.Add("Санвузол", "Санвузол");

DataGridViewTextBoxColumn priceColumn = new DataGridViewTextBoxColumn();

priceColumn.Name = "Ціна";

priceColumn.HeaderText = "Ціна (грн)";

priceColumn.ValueType = typeof(decimal);

priceColumn.SortMode = DataGridViewColumnSortMode.Automatic;

dataGridView1.Columns.Add(priceColumn);

foreach (DataGridViewColumn col in dataGridView1.Columns)

{

DataGridViewColumn copyCol = (DataGridViewColumn)col.Clone();

copyCol.SortMode = DataGridViewColumnSortMode.Automatic;

dataGridView2.Columns.Add(copyCol);

}

dataGridView1.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dataGridView2.DefaultCellStyle.WrapMode = DataGridViewTriState.True;

dataGridView1.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

dataGridView2.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.AllCells;

}

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

{

try

{

int місця = int.Parse(textBox1.Text);

double площа = double.Parse(textBox2.Text);

int ліжка = int.Parse(textBox3.Text);

string видЛіжка = comboBox1.Text;

string меблі = string.Join("; ", checkedListBox1.CheckedItems.Cast<string>());

string клімат = string.Join("; ", checkedListBox2.CheckedItems.Cast<string>());

bool wifi = checkBox1.Checked;

bool харчування = checkBox2.Checked;

string санвузол = comboBox2.Text;

decimal ціна = decimal.Parse(textBox4.Text);

Room room = new Room(місця, площа, ліжка, видЛіжка, меблі, wifi, клімат, харчування, санвузол, ціна);

rooms.Add(room);

dataGridView1.Rows.Add(

room.Місця,

room.Площа,

room.КількістьЛіжок,

room.ВидЛіжка,

room.Меблі,

room.WiFi ? "Є" : "Немає",

room.Клімат,

room.Харчування ? "Є" : "Немає",

room.Санвузол,

room.Ціна

);

MessageBox.Show("Кімната додана!");

textBox1.Text = "";

textBox2.Text = "";

textBox3.Text = "";

textBox4.Text = "";

checkBox1.Checked = false;

checkBox2.Checked = false;

comboBox1.SelectedIndex = -1;

comboBox2.SelectedIndex = -1;

for (int i = 0; i < checkedListBox1.Items.Count; i++)

{

checkedListBox1.SetItemChecked(i, false);

}

for (int i = 0; i < checkedListBox2.Items.Count; i++)

{

checkedListBox2.SetItemChecked(i, false);

}

}

catch

{

MessageBox.Show("Перевірте правильність введених даних!");

}

}

private void вивестиToolStripMenuItem\_Click(object sender, EventArgs e)

{

panel1.Show();

label28.Show();

}

private void label28\_Click(object sender, EventArgs e)

{

panel1.Hide();

label28.Hide();

}

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

{

SaveFileDialog saveFileDialog = new SaveFileDialog();

saveFileDialog.Filter = "CSV файли (\*.csv)|\*.csv|Всі файли (\*.\*)|\*.\*";

saveFileDialog.Title = "Зберегти таблицю";

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

using (StreamWriter sw = new StreamWriter(saveFileDialog.FileName))

{

for (int i = 0; i < dataGridView1.Columns.Count; i++)

{

sw.Write(dataGridView1.Columns[i].HeaderText);

if (i < dataGridView1.Columns.Count - 1) sw.Write(";");

}

sw.WriteLine();

foreach (DataGridViewRow row in dataGridView1.Rows)

{

if (!row.IsNewRow)

{

for (int i = 0; i < row.Cells.Count; i++)

{

sw.Write(row.Cells[i].Value);

if (i < row.Cells.Count - 1) sw.Write(";");

}

sw.WriteLine();

}

}

}

}

}

private void створитиToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog openFileDialog = new OpenFileDialog();

openFileDialog.Filter = "CSV файли (\*.csv)|\*.csv|Всі файли (\*.\*)|\*.\*";

openFileDialog.Title = "Відкрити таблицю";

if (openFileDialog.ShowDialog() == DialogResult.OK)

{

dataGridView1.Rows.Clear();

dataGridView1.Columns.Clear();

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

if (lines.Length > 0)

{

string[] headers = lines[0].Split(';');

foreach (string header in headers)

{

dataGridView1.Columns.Add(header, header);

}

for (int i = 1; i < lines.Length; i++)

{

string[] cells = lines[i].Split(';');

dataGridView1.Rows.Add(cells);

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

dataGridView2.Rows.Clear();

if (!int.TryParse(textBox5.Text, out int критерій))

{

MessageBox.Show("Введіть кількість місць для пошуку!");

return;

}

foreach (Room room in rooms)

{

if (room.Місця == критерій)

{

dataGridView2.Rows.Add(

room.Місця,

room.Площа,

room.КількістьЛіжок,

room.ВидЛіжка,

room.Меблі,

room.WiFi ? "Є" : "Немає",

room.Клімат,

room.Харчування,

room.Санвузол,

room.Ціна

);

}

}

}

}

}

**Результат роботи програми**

Зображення, що містить текст, знімок екрана, програмне забезпечення, Веб-сайт

Вміст на основі ШІ може бути неправильним.

Зображення, що містить текст, знімок екрана, монітор, число

Вміст на основі ШІ може бути неправильним.