**Лабораторна робота №1**

## Тема: Класи й об'єкти в С++ та C#.

***Мета роботи:*** Одержати практичні навички реалізації класів на С++ та C#.

**Варіант 24**

**Задача 1(8)**

**Порядок виконання роботи.**

1.Визначити клас користувача відповідно до варіанта завдання (дивись додаток).

2.Визначити в класі наступні конструктори: без параметрів, з параметрами, копіювання.

3.Визначити в класі деструктор.

4.Визначити в класі компонента-функції для перегляду і встановлення полів даних.

5.Визначити покажчик на компонент-функцію.

6.Визначити покажчик на екземпляр класу.

7.Написати демонстраційну програму, в якій створюються і руйнуються об'єкти класу користувача і кожен виклик конструктора і деструктора супроводжується видачею відповідного повідомлення (який об'єкт, який конструктор чи деструктор викликав).

8.Показати в програмі використання покажчика на об'єкт і покажчика на компонент-функцію.

**Умова:**

Зображення, що містить текст, Шрифт, знімок екрана, ряд

Вміст на основі ШІ може бути неправильним.

**Код:**

1)Опис класу, Product.h

#pragma once

#include <string>

using namespace std;

class Product

{

public:

string name;

int amount;

float value;

public:

Product();

Product(const string& name, int amount, float value);

Product(const Product& other);

~Product(void);

string getName();

int getAmount();

float getValue();

void setProduct(const string& n, int a, float v);

void setName(const string& n);

void setAmount(int a);

void setValue(float v);

void printProduct();

};

2)Product.cpp

#include "Product.h"

#include <iostream>

#include <iomanip>

#include <algorithm>

using namespace std;

Product::Product() : name(""), amount(0), value(0.0f) {}

Product::Product(const string& name, int amount, float value) {

setProduct(name, amount, value);

}

Product::Product(const Product& other)

: name(other.name), amount(other.amount), value(other.value) {

}

Product::~Product(void) {}

string Product::getName() { return name; }

int Product::getAmount() { return amount; }

float Product::getValue() { return value; }

void Product::setProduct(const string& n, int a, float v) {

setName(n);

setAmount(a);

setValue(v);

}

void Product::setName(const string& n) { name = n; }

void Product::setAmount(int a) {

amount = max(0, a);

}

void Product::setValue(float v) {

value = (v < 0.0f) ? 0.0f : v;

}

void Product::printProduct() {

cout << fixed << setprecision(2);

cout << "Product: " << name

<< ", amount: " << amount

<< ", value: " << value

<< ", total: " << (amount \* value)

<< endl;

}

3)main.cpp

#include "Product.h"

int main() {

Product p1;

p1.setProduct("USB Cable", 5, 12.50f);

p1.printProduct();

Product p2("Keyboard", 2, 399.99f);

p2.printProduct();

Product p3 = p2;

p3.setAmount(3);

p3.setValue(379.99f);

p3.printProduct();

Product bad("Test", -10, -5.0f);

bad.printProduct();

}

РЕЗУЛЬТАТ ПРОГРАМИ

Зображення, що містить текст, Шрифт, знімок екрана, чорний

Вміст на основі ШІ може бути неправильним.

**Задача 2(4)**

**Умова:**Створити клас Money, розробивши наступні елементи класу:

* + Поля:
    - int first;//номинал купюри
    - int second; //кількість купюр
  + Конструктор, що дозволяє створити екземпляр класу із заданими значенням полів.
  + Методи, що дозволяють:
    - вивести номінал і кількість купюр;
    - визначити, чи вистачить грошових коштів на покупку товару на суму N гривнів.
    - визначити, скільки шт товару вартості n гривнів можна купити на наявні грошові кошти.
  + Властивості:
    - які надають можливість одержати-встановити значення полів (доступне для читання і запису);
    - які надають можливість расчитатать суму грошей (доступне тільки для читання).

**Код:**1)Опис класу, Money.h  
#pragma once

#include <iostream>

using namespace std;

class Money {

private:

int nominal;

int count;

bool isCoin;

public:

Money();

Money(int n, int c, bool coin);

void input();

void print();

int getNominal();

int getCount();

bool getIsCoin();

long long amount();

};

long long totalAmount(Money arr[], int n);

bool canBuy(Money arr[], int n, long long sum);

long long howManyItems(Money arr[], int n, long long price);

2)Money.cpp

#include "Money.h"

Money::Money() {

nominal = 0;

count = 0;

isCoin = true;

}

Money::Money(int n, int c, bool coin) {

nominal = n;

count = c;

isCoin = coin;

}

void Money::input() {

char t;

do {

cout << "Це монета (m) чи купюра (b)? ";

cin >> t;

} while (!(t == 'm' || t == 'M' || t == 'b' || t == 'B'));

isCoin = (t == 'm' || t == 'M');

cout << "Дозволені номінали: ";

if (isCoin) cout << "1 2 5 10\n";

else cout << "20 50 100 200 500 1000\n";

bool ok = false;

do {

cout << "Введіть номінал: ";

cin >> nominal;

if (isCoin)

ok = (nominal == 1 || nominal == 2 || nominal == 5 || nominal == 10);

else

ok = (nominal == 20 || nominal == 50 || nominal == 100 ||

nominal == 200 || nominal == 500 || nominal == 1000);

if (!ok) cout << "Недопустимий номінал! Спробуйте ще раз.\n";

} while (!ok);

do {

cout << "Введіть кількість: ";

cin >> count;

if (count <= 0) cout << "Кількість має бути > 0. Спробуйте ще раз.\n";

} while (count <= 0);

}

void Money::print() {

if (isCoin)

cout << "Монета ";

else

cout << "Купюра ";

cout << nominal << " грн × " << count

<< " = " << amount() << " грн\n";

}

int Money::getNominal() { return nominal; }

int Money::getCount() { return count; }

bool Money::getIsCoin() { return isCoin; }

long long Money::amount() {

return 1LL \* nominal \* count;

}

long long totalAmount(Money arr[], int n) {

long long s = 0;

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

s += arr[i].amount();

return s;

}

bool canBuy(Money arr[], int n, long long sum) {

return totalAmount(arr, n) >= sum;

}

long long howManyItems(Money arr[], int n, long long price) {

if (price <= 0) return 0;

return totalAmount(arr, n) / price;

}

3) Main.cpp

#include <iostream>

#include "Money.h"

using namespace std;

int main() {

setlocale(0, "ukr");

int n;

cout << "Кількість різних номіналів: ";

cin >> n;

if (n <= 0 || n > 20) {

cout << "Некоректне n!\n";

return 0;

}

Money wallet[20];

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

cout << "\nЕлемент #" << i + 1 << endl;

wallet[i].input();

}

cout << "\nВаш набір грошей:\n";

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

wallet[i].print();

long long N;

cout << "\nСума покупки: ";

cin >> N;

if (canBuy(wallet, n, N))

cout << "Грошей вистачить.\n";

else

cout << "Грошей не вистачить.\n";

long long p;

cout << "Ціна товару: ";

cin >> p;

cout << "Максимум можна купити: "

<< howManyItems(wallet, n, p) << " шт\n";

cout << "Загальна сума: "

<< totalAmount(wallet, n) << " грн\n";

return 0;

}

РЕЗУЛЬТАТ ПРОГРАМИ

Зображення, що містить текст, знімок екрана, чорний

Вміст на основі ШІ може бути неправильним.

**Windows form Application  
Тема : «Рієлторська кантора»**Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Вміст на основі ШІ може бути неправильним.

**Об`єкти на формі  
Зображення, що містить текст, знімок екрана, програмне забезпечення, монітор

Вміст на основі ШІ може бути неправильним.**

**Код:  
Опис класу**using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace рієлторська\_контора\_приклад\_гуменна

{

[Serializable]

public class PropertyItem

{

public string Type { get; set; }

public string Address { get; set; }

public decimal Area { get; set; }

public int Rooms { get; set; }

public int YearBuilt { get; set; }

public decimal Price { get; set; }

public string Status { get; set; }

public string Currency { get; set; }

public PropertyItem() { }

public PropertyItem(string type, string address, decimal area, int rooms, int yearBuilt, decimal price, string status)

{

Type = type;

Address = address;

Area = area;

Rooms = rooms;

YearBuilt = yearBuilt;

Price = price;

Status = status;

}

}

}

**Код форми**using System;

using System.Collections.Generic;

using System.Drawing;

using System.IO;

using System.Text;

using System.Text.RegularExpressions;

using System.Windows.Forms;

namespace рієлторська\_контора\_приклад\_гуменна

{

public partial class Form1 : Form

{

private List<PropertyItem> properties = new List<PropertyItem>();

private string lastSortColumn = "";

private bool lastSortAsc = true;

public Form1()

{

InitializeComponent();

this.Load += FormMain\_Load;

txtStreetName.Validating += (s, e) => ValidateStreetName();

mtxHouse.Validating += (s, e) => ValidateHouse();

txtSearch.TextChanged += (s, e) => ApplyView();

cmbFilterType.SelectedIndexChanged += (s, e) => ApplyView();

cmbFilterStatus.SelectedIndexChanged += (s, e) => ApplyView();

btnResetFilters.Click += (s, e) =>

{

txtSearch.Clear();

if (cmbFilterType.Items.Count > 0) cmbFilterType.SelectedIndex = 0;

if (cmbFilterStatus.Items.Count > 0) cmbFilterStatus.SelectedIndex = 0;

ApplyView();

};

dataGridView1.ColumnHeaderMouseClick += dataGridView1\_ColumnHeaderMouseClick;

}

private void FormMain\_Load(object sender, EventArgs e)

{

ConfigureGrid();

mtxHouse.Mask = "000>LL/000";

mtxHouse.PromptChar = ' ';

if (cmbStreetType.Items.Count == 0)

{

cmbStreetType.DropDownStyle = ComboBoxStyle.DropDownList;

cmbStreetType.Items.AddRange(new object[]

{ "вул.", "просп.", "пров.", "бульв.", "пл.", "шосе", "набережна", "узвіз", "тракт", "кв-л", "мікрорайон" });

cmbStreetType.SelectedIndex = 0;

}

if (cmbCurrency.Items.Count == 0)

{

cmbCurrency.DropDownStyle = ComboBoxStyle.DropDownList;

cmbCurrency.Items.AddRange(new object[] { "UAH", "USD", "EUR" });

cmbCurrency.SelectedIndex = 0;

}

if (cmbFilterType.Items.Count == 0)

{

cmbFilterType.DropDownStyle = ComboBoxStyle.DropDownList;

cmbFilterType.Items.AddRange(new object[] { "всі", "Квартира", "Будинок", "Офіс", "Ділянка" });

cmbFilterType.SelectedIndex = 0;

}

if (cmbFilterStatus.Items.Count == 0)

{

cmbFilterStatus.DropDownStyle = ComboBoxStyle.DropDownList;

cmbFilterStatus.Items.AddRange(new object[] { "всі", "В продажу", "Оренда", "Продано", "Резерв" });

cmbFilterStatus.SelectedIndex = 0;

}

ApplyView();

}

private void ConfigureGrid()

{

var dgv = dataGridView1;

dgv.AllowUserToAddRows = false;

dgv.ReadOnly = true;

dgv.RowHeadersVisible = false;

dgv.AutoGenerateColumns = false;

dgv.AutoSizeRowsMode = DataGridViewAutoSizeRowsMode.DisplayedCells;

dgv.BackgroundColor = Color.White;

dgv.Columns.Clear();

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colType", HeaderText = "Тип", Width = 100 });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colAddress", HeaderText = "Адреса", Width = 260, DefaultCellStyle = { WrapMode = DataGridViewTriState.True } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colArea", HeaderText = "Площа", Width = 80, DefaultCellStyle = { Alignment = DataGridViewContentAlignment.MiddleRight, Format = "N2" } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colRooms", HeaderText = "Кімнат", Width = 70, DefaultCellStyle = { Alignment = DataGridViewContentAlignment.MiddleCenter, Format = "N0" } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colYear", HeaderText = "Рік", Width = 70, DefaultCellStyle = { Alignment = DataGridViewContentAlignment.MiddleCenter, Format = "N0" } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colPrice", HeaderText = "Ціна", Width = 100, DefaultCellStyle = { Alignment = DataGridViewContentAlignment.MiddleRight, Format = "N2" } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colCurrency", HeaderText = "Валюта", Width = 70, DefaultCellStyle = { Alignment = DataGridViewContentAlignment.MiddleCenter } });

dgv.Columns.Add(new DataGridViewTextBoxColumn { Name = "colStatus", HeaderText = "Статус", Width = 110 });

}

private bool ValidateStreetName()

{

string text = txtStreetName.Text.Trim();

bool ok = Regex.IsMatch(text, @"^[\p{L}][\p{L}\-’'\s]{0,}$", RegexOptions.CultureInvariant);

if (!ok)

errorProvider1.SetError(txtStreetName, "Тільки літери, пробіли, дефіси, апостроф.");

else

errorProvider1.SetError(txtStreetName, "");

return ok;

}

private bool ValidateHouse()

{

string raw = mtxHouse.Text.Trim();

bool ok = !string.IsNullOrWhiteSpace(raw);

if (!ok)

errorProvider1.SetError(mtxHouse, "Вкажіть номер (напр. 12, 12А, 12/3).");

else

errorProvider1.SetError(mtxHouse, "");

return ok;

}

private string BuildAddress()

{

string type = (cmbStreetType.Text ?? "").Trim();

string name = Regex.Replace((txtStreetName.Text ?? "").Trim(), @"\s+", " ");

string house = (mtxHouse.Text ?? "").Trim();

return type + " " + name + ", " + house;

}

private void ClearInputs()

{

txtStreetName.Clear();

mtxHouse.Clear();

numArea.Value = 0;

numRooms.Value = 0;

numYear.Value = DateTime.Now.Year;

numPrice.Value = 0;

errorProvider1.SetError(txtStreetName, "");

errorProvider1.SetError(mtxHouse, "");

txtStreetName.Focus();

}

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

{

if (!ValidateStreetName() | !ValidateHouse())

{

MessageBox.Show("Перевірте коректність адреси.", "Увага");

return;

}

PropertyItem item = new PropertyItem();

item.Type = cmbType.Text;

item.Address = BuildAddress();

item.Area = numArea.Value;

item.Rooms = (int)numRooms.Value;

item.YearBuilt = (int)numYear.Value;

item.Price = numPrice.Value;

item.Currency = cmbCurrency.Text;

item.Status = cmbStatus.Text;

properties.Add(item);

ApplyView();

ClearInputs();

}

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

{

ClearInputs();

}

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

{

ClearInputs();

if (cmbStreetType.Items.Count > 0) cmbStreetType.SelectedIndex = 0;

if (cmbCurrency.Items.Count > 0) cmbCurrency.SelectedIndex = 0;

if (cmbType.Items.Count > 0) cmbType.SelectedIndex = 0;

if (cmbStatus.Items.Count > 0) cmbStatus.SelectedIndex = 0;

}

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

{

if (properties.Count == 0)

{

MessageBox.Show("Немає даних для збереження.");

return;

}

SaveFileDialog sfd = new SaveFileDialog();

sfd.Filter = "Binary (\*.bin)|\*.bin";

sfd.FileName = "properties.bin";

if (sfd.ShowDialog() != DialogResult.OK) return;

SaveToBin(sfd.FileName);

MessageBox.Show("Дані збережено.", "OK");

}

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

{

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Binary (\*.bin)|\*.bin";

if (ofd.ShowDialog() != DialogResult.OK) return;

LoadFromBin(ofd.FileName);

MessageBox.Show("Дані завантажено.", "OK");

ApplyView();

}

private void SaveToBin(string path)

{

using (FileStream fs = new FileStream(path, FileMode.Create, FileAccess.Write))

using (BinaryWriter bw = new BinaryWriter(fs, Encoding.UTF8))

{

bw.Write(properties.Count);

foreach (PropertyItem it in properties)

{

bw.Write(it.Type ?? "");

bw.Write(it.Address ?? "");

bw.Write(it.Area);

bw.Write(it.Rooms);

bw.Write(it.YearBuilt);

bw.Write(it.Price);

bw.Write(it.Currency ?? "");

bw.Write(it.Status ?? "");

}

}

}

private void LoadFromBin(string path)

{

properties.Clear();

using (FileStream fs = new FileStream(path, FileMode.Open, FileAccess.Read))

using (BinaryReader br = new BinaryReader(fs, Encoding.UTF8))

{

int count = br.ReadInt32();

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

{

PropertyItem it = new PropertyItem();

it.Type = br.ReadString();

it.Address = br.ReadString();

it.Area = br.ReadDecimal();

it.Rooms = br.ReadInt32();

it.YearBuilt = br.ReadInt32();

it.Price = br.ReadDecimal();

it.Currency = br.ReadString();

it.Status = br.ReadString();

properties.Add(it);

}

}

}

private void ApplyView()

{

string q = (txtSearch.Text == null) ? "" : txtSearch.Text.Trim().ToLower();

string type = (cmbFilterType.SelectedItem as string) ?? "всі";

string status = (cmbFilterStatus.SelectedItem as string) ?? "всі";

dataGridView1.Rows.Clear();

foreach (PropertyItem it in properties)

{

bool ok = true;

if (q.Length > 0)

{

string addr = (it.Address ?? "").ToLower();

string t = (it.Type ?? "").ToLower();

string st = (it.Status ?? "").ToLower();

if (!addr.Contains(q) && !t.Contains(q) && !st.Contains(q))

ok = false;

}

if (ok && type != "всі")

{

if (!string.Equals(it.Type, type, StringComparison.OrdinalIgnoreCase))

ok = false;

}

if (ok && status != "всі")

{

if (!string.Equals(it.Status, status, StringComparison.OrdinalIgnoreCase))

ok = false;

}

if (ok)

{

dataGridView1.Rows.Add(

it.Type, it.Address, it.Area, it.Rooms,

it.YearBuilt, it.Price, it.Currency, it.Status

);

}

}

}

private void dataGridView1\_ColumnHeaderMouseClick(object sender, DataGridViewCellMouseEventArgs e)

{

string col = dataGridView1.Columns[e.ColumnIndex].Name;

if (lastSortColumn == col)

lastSortAsc = !lastSortAsc;

else

{

lastSortColumn = col;

lastSortAsc = true;

}

Comparison<PropertyItem> cmp = (a, b) =>

{

int r = 0;

switch (col)

{

case "colType": r = string.Compare(a.Type, b.Type, StringComparison.OrdinalIgnoreCase); break;

case "colAddress": r = string.Compare(a.Address, b.Address, StringComparison.OrdinalIgnoreCase); break;

case "colArea": r = a.Area.CompareTo(b.Area); break;

case "colRooms": r = a.Rooms.CompareTo(b.Rooms); break;

case "colYear": r = a.YearBuilt.CompareTo(b.YearBuilt); break;

case "colPrice": r = a.Price.CompareTo(b.Price); break;

case "colCurrency": r = string.Compare(a.Currency, b.Currency, StringComparison.OrdinalIgnoreCase); break;

case "colStatus": r = string.Compare(a.Status, b.Status, StringComparison.OrdinalIgnoreCase); break;

}

if (!lastSortAsc) r = -r;

return r;

};

properties.Sort(cmp);

ApplyView();

}

}

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

Вміст на основі ШІ може бути неправильним. **Виводим список з бінарного файлу**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, Комп’ютерна піктограма

Вміст на основі ШІ може бути неправильним. **Результат виводу**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Вміст на основі ШІ може бути неправильним. **Додаєм новий обєкт**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Вміст на основі ШІ може бути неправильним. **Результат додавання**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

Вміст на основі ШІ може бути неправильним. **Пошук**Зображення, що містить текст, знімок екрана, програмне забезпечення, Комп’ютерна піктограма

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. **Фільтрація  
  
Висновок**: я навчилась працювати з класами на мовах с++ та с#.