

# Лабораторна робота №1

## Класи й об'єкти в C++

**Мета:** Одержати практичні навички реалізації класів на C++ та C# в WinForm.

**Основний зміст роботи.**

- Написати програму, в якій створюються і руйнуються об'єкти, визначеного користувачем класу. Виконати дослідження викликів конструкторів і деструкторів.

### Завдання 1

Опису членів-даних класів користувача

#### 4.ВИРІБ

ім'я- char\*

шифр – char\*

кількість- int

### Код програми

#### Product.h

```
#pragma once
#ifndef PRODUCT_H
#define PRODUCT_H

#include <string>
class Product
{
    std::string _name;
    std::string _chipper;
    int _count;

public:
    Product();
    Product(std::string name, std::string chipper, int count);
    void Show();

    ~Product();
};
```

```
#endif // !PRODUCT_H
```

#### Product.cpp

```
#include "Product.h"
#include <iostream>

Product::Product() : _name("NoName"), _chipper("NoChipper"), _count(0)
{
    std::cout << "Default constructor worked!" << std::endl;
}

Product::Product(std::string name, std::string chipper, int count) : _name(name), _chipper(chipper), _count(count)
{
    std::cout << "Parametrized constructor worked!" << std::endl;
}

Product::~~Product()
{
    std::cout << "Destructor worked!" << std::endl;
}

void Product::Show(){
```

|                      |             |          |        |       |
|----------------------|-------------|----------|--------|-------|
| ЛР.ОК.19.ПІ231.01.04 |             |          |        |       |
| Змін.                | Аркуш       | № докум. | Підпис | Дата  |
| Розробив             | Дар'єв Д.О. |          |        |       |
| Перевірів            | Жереб Д.В.  |          |        |       |
| Н.контр.             |             |          |        |       |
| Затвер.              |             |          |        |       |
|                      |             |          | Лім    | Аркуш |
|                      |             |          | 1      | 1     |
|                      |             |          | ХПК    |       |

```

std::cout << "Name: " << _name << std::endl;
std::cout << "Chiper: " << _chiper << std::endl;
std::cout << "Count: " << _count << std::endl;
}
Lab_1_1_Daryev.cpp
#include "Product.h"

int main()
{
    {
        Product prodOne = Product("Bread", "12345", 10);
        prodOne.Show();
    }
    {
        Product* prodTwo = new Product("Milk", "67890", 20);
        prodTwo->Show();
    }
    return 0;
}

```

## Результат

```

Parametrized constructor worked!
Name: Bread
Chiper: 12345
Count: 10
Destructor worked!
Parametrized constructor worked!
Name: Milk
Chiper: 67890
Count: 20

```

## Завдання 2

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

- **Поля:**
  - ☐ int first; //номінал 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 програми

### Money.h

```

#ifndef MONEY_H
#define MONEY_H

class Money
{
    int _faceValue;
    int _count;
}

```

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
|      |      |          |        |      |                      | 2    |
| Вим. | Арк. | № докум. | Підпис | Дата |                      |      |

```

        int _sum;

public:
    Money();
    Money(int faceValue, int count);
    ~Money();

    void Show();
    bool whetherThereeMoney(int sum);
    int howManyBuy(int price);

    int getSum() { return _sum; }
    int getFaceValue() { return _faceValue; }
    int getCount() { return _count; }

    void setCount(int count) {
        _count = count;
        _sum = _faceValue * _count;
    }
    void setFaceValue(int faceValue) {
        _faceValue = faceValue;
        _sum = _faceValue * _count;
    }
};
#endif // !MONEY_H

```

## Money.cpp

```

#include <iostream>
#include "Money.h"

```

```

Money::Money() : _faceValue(0), _count(0), _sum(0)
{
    std::cout << "Default constructor called" << std::endl;
}
Money::Money(int faceValue, int count) : _faceValue(faceValue), _count(count), _sum(faceValue* count)
{
    std::cout << "Parametrized constructor called" << std::endl;
}
Money::~~Money()
{
    std::cout << "Destructor called" << std::endl;
}

void Money::Show()
{
    std::cout << "Face value: " << _faceValue << ", Count: " << _count << ", Sum: " << _sum << std::endl;
}
bool Money::whetherThereeMoney(int sum)
{
    return _sum >= sum;
}
int Money::howManyBuy(int price)
{
    return (int)(_sum / price);
}

```

## Lab\_1\_2\_Daryev.cpp

```

#include "Money.h"
#include <iostream>

```

```

void wheterYouHaveEnoughMoneyToBuy(Money wallet)
{
    int sum;
    std::cout << "Enter sum to check: ";
    std::cin >> sum;
    if (wallet.whetherThereeMoney(sum)) {
        std::cout << "There is enough money in the wallet." << std::endl;
    }
}

```

```

        else {
            std::cout << "There is not enough money in the wallet." << std::endl;
        }
    }
    void howMuchYouCanBuy(Money wallet)
    {
        int price;
        std::cout << "Enter product price: ";
        std::cin >> price;
        int canBuy = wallet.howManyBuy(price);
        std::cout << "You can buy " << canBuy << " items with the price of " << price << "." << std::endl;
    }
    int main()
    {
        {
            int n;
            std::cout << "Enter number of different money types in wallet: ";
            std::cin >> n;
            Money *wallet=new Money[n];

            for (int i = 0; i < n; i++)
            {
                int somenumber;
                std::cout << "Enter face value money from wallet " << i + 1 << ": ";
                std::cin >> somenumber;
                wallet[i].setFaceValue(somenumber) ;
                std::cout << "Enter count money from wallet " << i + 1 << ": ";
                std::cin >> somenumber;
                wallet[i].setCount(somenumber);
            }
            delete[] wallet;
        }
    }
}

```

## Результат

```

Enter number of different money types in wallet: 2
Default constructor called
Default constructor called
Enter face value money from wallet 1: 3
Enter count money from wallet 1: 12
Enter face value money from wallet 2: 43
Enter count money from wallet 2: 2
Destructor called
Destructor called

```

## Завдання 3

Створення форми з кнопками WinForm

## Код програми

### Form1.Designer.cs

```

namespace OOP_Lab_1_Daryev
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
    }
}

```

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                      | 4    |

```

/// </summary>
/// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 15.75F, System.Drawing.FontStyle.Italic,
System.Drawing.GraphicsUnit.Point, ((byte)(0)));
    this.label1.Location = new System.Drawing.Point(12, 9);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(760, 48);
    this.label1.TabIndex = 0;
    this.label1.Text = "Тихіше їдеш - далі будеш!";
    this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
    this.label1.Click += new System.EventHandler(this.label1_Click);
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleModeMode = System.Windows.Forms.AutoScaleModeMode.Font;
    this.BackColor = System.Drawing.SystemColors.ActiveCaption;
    this.ClientSize = new System.Drawing.Size(784, 461);
    this.Controls.Add(this.label1);
    this.Name = "Form1";
    this.Text = "Form1";
    this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.Label label1;
}

```

## Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace OOP_Lab_1_Daryev
{
    public partial class Form1 : Form
    {
        int _Width = 800;
        int _Height = 500;
        public Form1()
        {
            InitializeComponent();
            Button[] buttons = new Button[6];

```

|                      |      |          |        |      |      |
|----------------------|------|----------|--------|------|------|
| ЛР.ОК.19.ПІ231.01.04 |      |          |        |      | Арк. |
| Вим.                 | Арк. | № докум. | Підпис | Дата | 5    |

```

for (int i = 0; i < buttons.Length; i++)
{
    buttons[i] = new Button();
    buttons[i].Size = new Size(300 - i * 20, 50);
    buttons[i].Location = new Point((_Width / 6) + (100 + i * 10), 60 + i * 60);
    if (i % 2 == 0)
    {
        buttons[i].Font = new Font("Comic Sans MS", 16, FontStyle.Italic);
        buttons[i].BackColor = Color.LightBlue;
    }
    else
    {
        buttons[i].Font = new Font("Arial", 16, FontStyle.Italic);
        buttons[i].BackColor = Color.Aquamarine;
    }
    buttons[i].Text = $"History {i + 1}";
    this.Controls.Add(buttons[i]);
    if (i == buttons.Length - 1)
    {
        buttons[i].Click += new System.EventHandler(this.closeWindow);
    }
    else
    {
        buttons[i].Click += new System.EventHandler(this.someButtonClick);
    }
}

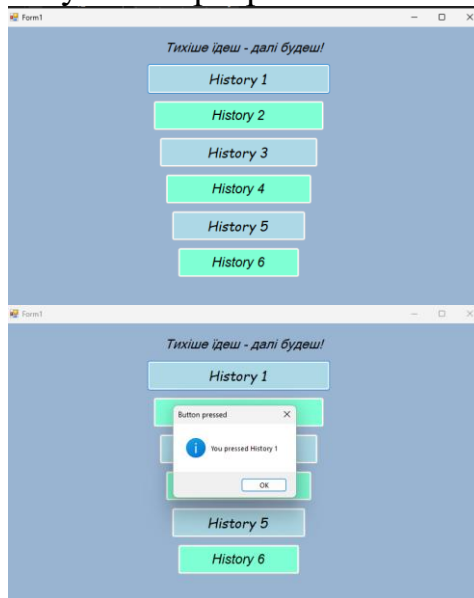
void closeWindow(object sender, EventArgs e)
{
    Application.Exit();
}

void someButtonClick(object sender, EventArgs e)
{
    Button btn = sender as Button;
    MessageBox.Show($"You pressed {btn.Text}", "Button pressed", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void label1_Click(object sender, EventArgs e)
{
    MessageBox.Show("THIS IS NOT BUTTON", "THIS IS NOT BUTTON", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}

```

## Результат програми



## Завдання 4

Створіть додаток-форму Windows. Опишіть в модулі форми під класом форми базовий клас згідно варіанту, передбачивши в ньому не менше трьох полів, двох

конструкторів різних видів, двох методів для виконання обчислень. Створіть також у цьому класі метод Info для виводу інформації.

## Код програми

### Form1.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Windows.Forms;

namespace OOPLab_1_4_Daryev
{
    public partial class MainForm : Form
    {
        private readonly List<PhoneClass> phoneList;
        private readonly DataTable dt;

        public MainForm()
        {
            InitializeComponent();

            phoneList = new List<PhoneClass>();
            dt = new DataTable();

            dt.Columns.Add("Company", typeof(string));
            dt.Columns.Add("Phone model", typeof(string));
            dt.Columns.Add("Phone year", typeof(DateTime));
            dt.Columns.Add("Price", typeof(int));

            dataGridView1.DataSource = dt;
            UpdateCounter();
        }

        private void createWithoutParButton_Click(object sender, EventArgs e)
        {
            var phone = new PhoneClass();
            phoneList.Add(phone);
            dt.Rows.Add(phone.Company, phone.PhoneModel, phone.PhoneYear, phone.Price);
            UpdateCounter();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (!ValidateInputs(out string company, out string model, out DateTime year, out int price))
                return;

            var phone = new PhoneClass(company, model, price, year);
            phoneList.Add(phone);
            dt.Rows.Add(phone.Company, phone.PhoneModel, phone.PhoneYear, phone.Price);
            UpdateCounter();
        }

        private bool ValidateInputs(out string company, out string model, out DateTime year, out int price)
        {
            company = textBox1.Text.Trim();
            model = textBox2.Text.Trim();
            year = DateTime.MinValue;
            price = 0;

            if (string.IsNullOrEmpty(company) ||
                string.IsNullOrEmpty(model) ||
                string.IsNullOrEmpty(maskedTextBox1.Text) ||
                string.IsNullOrEmpty(textBox3.Text))
            {
                MessageBox.Show("Please, fill all fields", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }

            if (!DateTime.TryParse(maskedTextBox1.Text, out year) || year < new DateTime(1900, 01, 01) || year >
                DateTime.Now)
            {
                MessageBox.Show("Year is invalid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }

            if (!int.TryParse(textBox3.Text, out price) || price <= 0)
            {
            }
        }
    }
}
```

|      |      |          |        |      |                      |           |
|------|------|----------|--------|------|----------------------|-----------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк.<br>7 |
| Вим. | Арк. | № докум. | Підпис | Дата |                      |           |

```

        MessageBox.Show("Price must be a positive number", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return false;
    }

    return true;
}

private void UpdateCounter()
{
    classCountLabel.Text = phoneList.Count.ToString();
}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar))
        e.Handled = true;
}

private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsLetterOrDigit(e.KeyChar) && e.KeyChar != '-' && e.KeyChar !=
    ' ')
        e.Handled = true;
}

private void maskedTextBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsNumber(e.KeyChar))
        e.Handled = true;
}

private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!char.IsControl(e.KeyChar) && !char.IsNumber(e.KeyChar))
        e.Handled = true;
}
}
}

```

## PhoneClass.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OOPLab_1_4_Daryev
{
    internal class PhoneClass
    {
        static public int classCounter { get; set; }=0;
        public string Company { get; set; }
        public string PhoneModel { get; set; }
        public int Price { get; set; }
        public DateTime PhoneYear { get; set; }

        public PhoneClass()
        {
            Company = "NoName";
            PhoneModel = "NoModel";
            Price = 0;
            PhoneYear = new DateTime(1950, 01, 01);
            ++classCounter;
        }

        public PhoneClass(string _company, string _phoneModel, int _price, DateTime _phoneYear)
        {
            Company = _company;
            PhoneModel = _phoneModel;
            Price = _price;
            PhoneYear = new DateTime();
            PhoneYear = _phoneYear;
            ++classCounter;
        }

        public override string ToString()
    }
}

```



```

    {
        return $"Company: {Company}, Phone model: {PhoneModel}, Price: {Price}, Phone year:
{PhoneYear.ToShortDateString()}";
    }
}
}

```

## Результат програми

### 1 – Форма без даних

Lab 1 Daryev

Without parameters From parameters

Create phone without parameters

| Company | Phone model | Phone year | Price |
|---------|-------------|------------|-------|
|---------|-------------|------------|-------|

Class count: 0

### 2 – Форма з одним об'єктом за замовчуванням

Lab 1 Daryev

Without parameters From parameters

Create phone without parameters

| Company | Phone model | Phone year | Price |
|---------|-------------|------------|-------|
| NoName  | NoModel     | 1/1/1950   | 0     |

Class count: 1

### 3 – Форма з об'єктом за замовчуванням та об'єктом з параметрами

Lab 1 Daryev

Without parameters From parameters

Company: Nokia  
Model: 3310  
Year: 12.12.1985  
Price: 1000

Create phone from parameters

| Company | Phone model | Phone year | Price |
|---------|-------------|------------|-------|
| NoName  | NoModel     | 1/1/1950   | 0     |
| Nokia   | 3310        | 12/12/1985 | 1000  |

Class count: 2

**Висновок:** на лабораторній роботі було одержано практичні навички реалізації класів на C++ та C# в WinForm.

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                      | 10   |

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
|      |      |          |        |      |                      | 11   |
| Вим. | Арк. | № докум. | Підпис | Дата |                      |      |

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
|      |      |          |        |      |                      | 12   |
| Вим. | Арк. | № докум. | Підпис | Дата |                      |      |

|      |      |          |        |      |                      |      |
|------|------|----------|--------|------|----------------------|------|
|      |      |          |        |      | ЛР.ОК.19.ПІ231.01.04 | Арк. |
|      |      |          |        |      |                      | 13   |
| Вим. | Арк. | № докум. | Підпис | Дата |                      |      |

