## Лабораторна робота №1. Класи й об'єкти в С++

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

**Основний зміст роботи.**

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

**Завдання 1 (Варіант 10 цех)**

**Умова:**

|  |
| --- |
| **10.ЦЕХ** |
| ім'я- char\* |
| начальник – char\* |
| кількість працюючих- int |

Код програми:

|  |  |
| --- | --- |
| Factory.h  #pragma once  #include <iostream>  #include <string>  using namespace std;  class Factory  {  private:  string name;  string ceo;  int employers;  public:  Factory();  Factory(string n, string c, int empl);  Factory(const Factory& other);  ~Factory();  string getName();  string getCeo();  int getEmployers();  void setFactory(string n, string c, int empl);  void setName(string n);  void setCeo(string c);  void setEmployers(int empl);  void printFactory();  };  Factory.cpp  #include "Factory.h"  #include <string>  Factory::Factory(){}  Factory::Factory(string n, string c, int empl) {  }  Factory::Factory(const Factory& other)  {  name = other.name;  ceo = other.ceo;  employers = other.employers;  }  Factory::~Factory() {  }  string Factory::getName()  {  return name;  }  string Factory::getCeo()  {  return ceo;  }  int Factory::getEmployers()  {  return employers;  } | void Factory::setFactory(string n, string c, int empl)  {  name = n;  ceo = c;  employers = empl;  }  void Factory::setName(string n)  {  name = n;  }  void Factory::setCeo(string c)  {  ceo = c;  }  void Factory::setEmployers(int empl)  {  employers = empl;  }  void Factory::printFactory()  {  cout << "Factory name: " << (name.length() != 0 ? name : "NULL")  << ", CEO: " << (ceo.length() ? ceo : "NULL")  << ", Employers: " << employers << endl;  }  LR1\_01.cpp  #include "Factory.h"  #include <tchar.h>  #include <iostream>  #include <conio.h>  using namespace std;  void print(Factory ob)  {  cout << "Drue z funkcii: ";  ob.printFactory();  }  int \_tmain(int argc, \_TCHAR\* argv[])  {  Factory chel;  Factory \*chel1;  chel1= new Factory;  Factory Spis[2];  chel.setFactory("Butcher","Bill", 10);  chel1->setFactory("Metal","Billy", 20);  chel1->printFactory();  chel.printFactory();  chel1->~Factory();  chel.~Factory();  /\*Spis[0].~Factory();  Spis[1].~Factory();  Spis[2].~Factory();\*/  \_getch();  return 0;  } |

**Робота програми:**

**Factory name: Metal, CEO: Billy, Employers: 20**

**Factory name: Butcher, CEO: Bill, Employers: 10**

**Опис програми:**

Програма демонструє роботу з класом Factory, що зберігає дані про цех (назва, директор, кількість працівників) та дозволяє їх змінювати і виводити.

**Завдання 2 (Варіант 10 DataTime)**

**Умова:**

1. Створити клас для роботи з датою. Розробити наступні елементи класу:
   * Поле DataTime data.
   * Конструктори, що дозволяють встановити:
     + задану дату
     + дату 1.01.2009
   * Методи, що дозволяють:
     + обчислити дату попереднього дня;
     + обчислити дату наступного дня;
     + визначити скільки днів залишилося до кінця місяця.
   * Властивості:
     + які надають можливість встановити або отримати значення поля класу (доступно для читання і запису)
     + які надають можливість визначити рік высокосным (доступно тільки для читання)

**Код програми:**

|  |  |
| --- | --- |
| **DataTimes.h:**  #pragma once  class DataTimes  {  public:  unsigned short int dd;  unsigned short int mm;  unsigned int yyyy;  public:  DataTimes();  DataTimes(unsigned short int d, unsigned short int m, unsigned int y);  ~DataTimes();  void setAll(unsigned short int d, unsigned short int m, unsigned int y);  int getYear();  int getMounth();  int getDay();  DataTimes Input();  void Print();  DataTimes NextDay(DataTimes ob);  DataTimes PrevDay(DataTimes ob);  int CountDay();  };  **LR1\_02:**  #include "DataTimes.h"  #include "tchar.h"  #include <iostream>  using namespace std;  int \_tmain(int argc, \_TCHAR\* argv[])  {  int n;  setlocale(LC\_ALL, "");  cout << "Введiть кiлькiсть об'єктiв:";  cin >> n;  DataTimes ob[100];  for (int i = 0; i < n; i++)  {  cout << "Введiть об'єкт №" << i + 1 << "\n";  ob[i].Input();  cout << "Кiлькiсть днiв до кiнця мiсяця = " << ob[i].CountDay() << "\n";  DataTimes tmp;  tmp.PrevDay(ob[i]);  cout << "Дата попереднього дня:";  tmp.Print();  tmp.NextDay(ob[i]);  cout << "Дата наступного дня:";  tmp.Print();  }  setlocale(LC\_ALL, "");  system("pause");  return 0;  } | **DataTimes.cpp:**  #include "DataTimes.h"  #include <iostream>  using namespace std;  DataTimes::DataTimes() :dd(1), mm(1), yyyy(2009) {};  DataTimes::DataTimes(unsigned short int d = 1, unsigned short int m = 1, unsigned int y = 2009)  {  }  DataTimes::~DataTimes(void)  {  }  void DataTimes::setAll(unsigned short int d, unsigned short int m, unsigned int y)  {  dd = d;  mm = m;  yyyy = y;  }  int DataTimes::getYear()  {  return yyyy;  }  int DataTimes::getMounth()  {  return mm;  }  int DataTimes::getDay()  {  return dd;  }  DataTimes DataTimes::Input()  {  setlocale(0, "UKR");  cout << "Введiть рiк:";  cin >> yyyy;  cout << "Введiть мiсяць:";  cin >> mm;  if (mm <= 12 && mm > 0)  {  cout << "Введiть день:";  cin >> dd;  if (dd > 0 && dd <= 31)  {  setlocale(0, "");  return DataTimes(dd, mm, yyyy);  }  else  {  cout << "Введено не коректний день!"<<endl;  exit(0);  }  }  else  {  cout << "Введено не коректний мiсяць!" << endl;  exit(0);  }  }  DataTimes DataTimes::NextDay(DataTimes ob)  {  this->dd = ob.dd;  this->mm = ob.mm;  this->yyyy = ob.yyyy;  this->dd++;  if ((mm == 1) || (mm == 3) || (mm == 5) || (mm == 7) || (mm == 8) || (mm == 10) || (mm == 12))  {  if (dd > 31) { dd = 1; mm++; }  }  else  {  if ((mm == 4) || (mm == 6) || (mm == 9) || (mm == 11))  {  if (dd > 30) { dd = 1; mm++; }  }  else  {  if (((yyyy % 4 == 0) && (yyyy % 100 != 0)) || ((yyyy % 4 == 0) && (yyyy % 100 != 0) && (yyyy % 400 == 0)))  {  if (dd > 29)  {  dd = 1; mm++;  }  }  else  {  if (dd > 28)  {  dd = 1; mm++;  }  }  }  }  if (mm > 12) {  mm = 1;  yyyy++;  }  return DataTimes(dd, mm, yyyy);  }  DataTimes DataTimes::PrevDay(DataTimes ob)  {  this->dd = ob.dd;  this->mm = ob.mm;  this->yyyy = ob.yyyy;  this->dd--;  if ((mm == 1) || (mm == 3) || (mm == 5) || (mm == 7) || (mm == 8) || (mm == 10) || (mm == 12))  {  if (dd < 0) {  dd = 30; mm--;  if (((yyyy % 4 == 0) && (yyyy % 100 != 0) && (mm == 2)) || ((yyyy % 4 == 0) && (yyyy % 100 != 0) && (yyyy % 400 == 0) && (mm == 2)))  {  if (dd < 1)  {  dd = 29;  }  }  else  {  if (dd < 1)  {  dd = 28;  }  }  }  }  else  {  if ((mm == 2) || (mm == 4) || (mm == 6) || (mm == 9) || (mm == 11))  {  if (dd < 0) { dd = 31; mm--; }  }  }  if ((dd < 1) && (mm == 1)) {  dd = 31;  mm = 12;  yyyy--;  }  return DataTimes(dd, mm, yyyy);  }  int DataTimes::CountDay()  {  switch (mm)  {  case 1:case 3:case 5:case 7:case 8:case 10:case 12: { return 31 - dd; break; };  case 4:case 6:case 9:case 11: { return 30 - dd; break; }  case 2:  {  if (((yyyy % 4 == 0) && (yyyy % 100 != 0) && (mm == 2)) || ((yyyy % 4 == 0) && (yyyy % 100 != 0) && (yyyy % 400 == 0) && (mm == 2)))  return 29 - dd; else return 28 - dd;  break;  }  default: break;  }  }  void DataTimes::Print() {  printf("%02d.%02d.%04d\n", dd, mm, yyyy);  } |

**Опис програми:**

Програма реалізує клас DataTimes для роботи з датами, що дозволяє вводити дату, виводити її, знаходити попередній і наступний день, підраховувати кількість днів до кінця місяця та перевіряти коректність введених даних.

**Результат роботи програми:**

**Введiть кiлькiсть об'єктiв:2**

**Введiть об'єкт №1**

**Введiть рiк:2000**

**Введiть мiсяць:12**

**Введiть день:23**

**Кiлькiсть днiв до кiнця мiсяця = 8**

**Дата попереднього дня:22.12.2000**

**Дата наступного дня:24.12.2000**

**Введiть об'єкт №2**

**Введiть рiк:2003**

**Введiть мiсяць:13**

**Введено не коректний мiсяць!**

**Завдання 3(Додаткове варіант 10 цукерки)**

**Умова:** 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.

|  |  |
| --- | --- |
| **№ Варіанту** | **Базовий клас** |
| 10. | Цукерки |

**Код програми:**

|  |  |
| --- | --- |
| **Form1.cs**  using System;  using System.Drawing;  using System.Windows.Forms;  namespace LR1.\_03  {  public partial class Form1 : Form  {  public Form1()  {  InitializeComponent();  label1.Text = "";  label2.Text = "";  label3.Text = "";  label4.Text = "";  label5.Text = "";  }  class Candy  {  private string name;  private double weight;  private double price;  public Candy()  {  name = "Карамелька";  weight = 20;  price = 5;  }  public Candy(string n, double w, double p)  {  name = n;  weight = w;  price = p;  }  public double PricePerKg()  {  return price / weight \* 1000;  }  public CandyControl GetCandyControl()  {  CandyControl candyControl = new CandyControl();  candyControl.CandyName = name;  candyControl.Weight = weight;  candyControl.Price = price;  candyControl.PricePerKg = PricePerKg();  return candyControl;  }  }  private void button1\_Click(object sender, EventArgs e)  {  Candy c1 = new Candy();  CandyControl control = c1.GetCandyControl();  flowLayoutPanel1.Controls.Add(control);  label1.Text = "Назва";  label2.Text = "Вага";  label3.Text = "Ціна";  label4.Text = "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  label5.Text = "Ціна за кг";  }  private void button2\_Click(object sender, EventArgs e)  {  using (CandyInputForm inputForm = new CandyInputForm())  {  if (inputForm.ShowDialog() == DialogResult.OK)  {  CandyControl candyControl = new CandyControl();  candyControl.CandyName = inputForm.CandyName;  candyControl.Weight = inputForm.Weight;  candyControl.Price = inputForm.Price;  candyControl.PricePerKg = inputForm.PricePerKg;  flowLayoutPanel1.Controls.Add(candyControl);  label1.Text = "Назва";  label2.Text = "Вага";  label3.Text = "Ціна";  label4.Text = "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  label5.Text = "Ціна за кг";  }  }  }  private void button3\_Click(object sender, EventArgs e)  {  this.Close();  }  }  } | **Candy.cs**  using System;  using System.Windows.Forms;  namespace LR1.\_03  {  public partial class CandyControl : UserControl  {  public string CandyName { get; set; }  public double Weight { get; set; }  public double Price { get; set; }  public double PricePerKg { get; set; }  public CandyControl()  {  InitializeComponent();  this.Load += CandyControl\_Load;  }  private void CandyControl\_Load(object sender, EventArgs e)  {  label1.Text = CandyName;  label2.Text = Weight.ToString() + " г";  label3.Text = Price.ToString() + " грн";  label4.Text = PricePerKg.ToString() + " грн";  }  }  }  **CandyInputForm.cs**  using System;  using System.Windows.Forms;  namespace LR1.\_03  {  public partial class CandyInputForm : Form  {  public string CandyName { get; private set; }  public double Weight { get; private set; }  public double Price { get; private set; }  public double PricePerKg { get; private set; }  public CandyInputForm()  {  InitializeComponent();  }  private void buttonOk\_Click(object sender, EventArgs e)  {  CandyName = textBoxName.Text;  if (!double.TryParse(textBoxWeight.Text, out double w))  {  MessageBox.Show("Введіть правильну вагу!");  return;  }  Weight = w;  if (!double.TryParse(textBoxPrice.Text, out double p))  {  MessageBox.Show("Введіть правильну ціну!");  return;  }  Price = p;  this.DialogResult = DialogResult.OK;  this.Close();  }  private void buttonCancel\_Click(object sender, EventArgs e)  {  this.DialogResult = DialogResult.Cancel;  this.Close();  }  }  } |

**Опис програми:**

Це Windows Forms програма для роботи з цукерками, яка містить клас Candy для зберігання даних (назва, вага, ціна) та обчислення ціни за кілограм, користувацький елемент CandyControl для відображення інформації про одну цукерку, форму введення CandyInputForm для додавання нових цукерок і головну форму Form1, де за допомогою кнопок можна додати цукерку за замовчуванням, ввести власні дані чи закрити програму, а всі цукерки відображаються у вигляді карток у FlowLayoutPanel.

**Результат роботи програми:**

****

**Висновок:** У ході виконання лабораторної роботи було закріплено знання з об’єктно-орієнтованого програмування, створено класи з конструкторами, методами та властивостями, а також розроблено Windows Forms застосунок, що демонструє практичне застосування інкапсуляції та взаємодії між класами, дозволяючи працювати з даними про цукерки у зручній графічній формі.