

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 5

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 22

Виконав:

студент 1-го курсу

гр. ІІІ-22 ФІОТ

Підпанюк Віталій Андрійович

Київ 2023

Завдання:

22. Створити клас “Товар”, який містить назву, дату виготовлення, ціну, кількість одиниць, а також методи порівняння дат та обчислення сумарної вартості товару. На його основі створити класи-нащадки “Промисловий товар”, що додатково зберігає умови транспортування, місце знаходження товару (на складі, в торговому залі) та “Харчовий продукт”, який додатково містить термін зберігання дати продукту. Створити n номенклатур промислових товарів та m номенклатур харчових продуктів. Визначити загальну вартість харчових продуктів, термін зберігання яких закінчився, і загальну вартість промислових товарів, які знаходяться на складі.

Код програми:

main.cpp:

```
#include <iostream>
#include "product.h"
#include "function.h"
using namespace std;

int main() {
    double total_gr_price = 0, total_ind_price = 0;
    int n, m;
    cout<<"Enter quantity of industrial products"<<endl;
    cin>>n;
    cout<<"Enter quantity of grocery products"<<endl;
    cin>>m;
    cout<<"Creating of industrial products..."<<endl;
    product** ind_array = array_creation(n, '1',
total_ind_price);
    cout<<"Creating of grocery products..."<<endl;
    product** gr_array = array_creation(m, '0',
total_gr_price);
    clear_memory(ind_array, n);
    clear_memory(gr_array, m);
    cout<<"Total price of overdue products equals
"<<total_gr_price<<endl;
    cout<<"Total price of industrial products in warehouse
equals "<<total_ind_price<<endl;
    return 0;
```

```
}
```

function.cpp:

```
#include "function.h"
```

```
#include "industrial_product.h"
```

```
void clear_memory(product** arr, int n)
```

```
{
```

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

```
    {
```

```
        delete arr[i];
```

```
    }
```

```
    delete[] arr;
```

```
}
```

```
product** array_creation(int n, char ch, double  
&total_price)
```

```
{
```

```
    auto** arr = new product*[n];
```

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

```
    {
```

```
        product base_info = product_creation();
```

```
        if(ch == '0')
```

```
        {
```

```
            Data exp_date = enter_exp_date();
```

```
            arr[i] = new grocery_product(base_info,  
exp_date);
```

```
            total_price += arr[i]->summary_price_check();
```

```
        }
```

```
        if(ch == '1')
```

```
        {
```

```
            string transport = enter_transport();
```

```
            string storage = enter_storage();
```

```
            arr[i] = new industrial_product(base_info,  
transport, storage);
```

```
            total_price+=arr[i]->summary_price_check();
```

```
        }
```

```
    }
```

```

        return arr;
    }

    product product_creation()
    {
        int base_day, base_month, base_year;
        string base_name;
        cout<<"Enter product`s name:\n";
        cin>>base_name;
        cout<<"Enter the date of manufacture in format dd mm
yyyy:\n";
        cin>>base_day>>base_month>>base_year;
        Data base_date_of_manufacture(base_day, base_month,
base_year);
        double base_price;
        cout<<"Enter product`s price (per unit):\n";
        cin>>base_price;
        int base_amount;
        cout<<"Enter amount of product\n";
        cin>>base_amount;
        product base_product(base_name,
base_date_of_manufacture, base_price,base_amount);
        return base_product;
    }

```

```

Data enter_exp_date()
{
    int day, month, year;
    cout<<"Enter the expiration date in format dd mm
yyyy\n";
    cin>>day>>month>>year;
    Data expiration_date(day, month, year);
    return expiration_date;
}

```

```

string enter_transport()
{
    string transporting;
    char ch;

```

```

        cout<<"Choose type of transporting:\n(0) by cargo
van\n(1) by plane\n(2) by train\n(3) by ship\n";
        cin>>ch;
        if(ch == '0')
        {
            transporting = "Cargo van";
        }
        if(ch == '1')
        {
            transporting = "Plane";
        }
        if(ch == '2')
        {
            transporting = "Train";
        }
        if(ch == '3')
        {
            transporting = "Ship";
        }
        ch = '\0';
        return transporting;
    }

string enter_storage()
{
    string storage;
    char ch;
    cout<<"Choose storage\n(0)-warehouse\n(1)-hall"<<endl;
    cin>>ch;
    if(ch=='0')
    {
        storage = "warehouse";
    }
    if(ch == '1')
    {
        storage = "hall";
    }
    return storage;
}

```

function.h:

```
#pragma once
#include <iostream>
#include "product.h"
#include "grocery_product.h"
#include "industrial_product.h"

product** array_creation(int n, char ch, double&);
void clear_memory(product** arr, int n);
product product_creation();
grocery_product create_grocery(product);
industrial_product create_industrial(product);
Data enter_exp_date();
string enter_transport();
string enter_storage();
```

product.cpp:

```
#include <iostream>
#include "product.h"
#include <ctime>

double product::summary_price_check()
{
    return amount * price;
}

int Data::get_day() const {return day;}
int Data::get_month() const {return month;}
int Data::get_year() const {return year;}

Data product::time_since_made()
{
    Data time_since_made;
    Data current_data = Data::push_current_data();
    time_since_made = current_data - date_of_manufacture;
    return time_since_made;
}

Data Data::operator-(Data temp_data) const
{

```

```

    Data push_data;
    push_data.day = day - temp_data.day;
    push_data.month = month - temp_data.month;
    push_data.year = year - temp_data.year;
    return push_data;
}

bool Data::operator<(Data temp_data) const
{
    bool result;
    Data push_data = push_current_data();
    if((push_data.day + push_data.month*30 +
push_data.year*365)>(day + month*30 + year*365))
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

```

Data Data::push_current_data()
{
    time_t currentTime = time(nullptr);

    // Перетворити час в локальну структуру часу
    tm* localTime = localtime(&currentTime);

    // Отримати дату зі структури часу
    int t_year = localTime->tm_year + 1900; // Додати 1900
до отриманого значення
    int t_month = localTime->tm_mon + 1; // Додати 1 до
отриманого значення (місяці відлічуються з 0)
    int t_day = localTime->tm_mday;
    Data current_data(t_day, t_month, t_year);
    return current_data;
}

```

product.h:

```
#pragma once
#include <iostream>

using namespace std;

class Data {
    int day,
        month,
        year;
public:
    int get_day() const;
    int get_month() const;
    int get_year() const;
    Data(): day(0), month(0), year(0) {} //конструктор за
замовчуванням
    static Data push_current_data(); //метод виклику
поточної дати
    Data(int temp_day, int temp_month, int temp_year) :
day(temp_day), month(temp_month), year(temp_year){}
//список ініціалізації для об'єкта типу дата
    Data operator-(Data) const; //перевантаження оператора
"-" для об'єктів типу дата
    bool operator<(Data) const;
};
```

```
class product {
protected:
    string name;
    Data date_of_manufacture;
    double price;
    int amount;

public:
    //product& operator=(const product other);
    virtual double summary_price_check();
    Data time_since_made();
    product(string &t_name, Data t_date_of_manufacture,
```



```

double t_price, int t_amount):name(t_name),
date_of_manufacture(t_date_of_manufacture),
price(t_price),

amount(t_amount){}
    product()
    {
        name = "empty";
        date_of_manufacture = Data(0, 0, 0);
        price = 0;
        amount = 0;
    }
};

```

grocery_product.cpp:

```

#include "grocery_product.h"

double grocery_product::summary_price_check()
{
    double total_price = 0;
    Data time_made = time_since_made();
    if(expiration_date < time_made)
    {
        total_price = amount * price;
    }
    return total_price;
    //return 1;
}

```

grocery_product.h:

```

#pragma once
#include "product.h"

class grocery_product:public product{
private:
    Data expiration_date;

```

```

public:
    double summary_price_check() override;
    grocery_product(product &base_product, Data
t_expiration_data) : product(base_product),
expiration_date(t_expiration_data){}

};

```

industrial_product.cpp:

```

#include "industrial_product.h"

double industrial_product::summary_price_check(){
    double total_price = 0;
    string temp = "warehouse"; //string check
    if(storage==temp)
    {
        total_price = price * amount;
    }
    else
    {
        cout<<"Product is not in the warehouse"<<endl;
    }
    return total_price;
}

void industrial_product::set_transport(string
&t_transport)
{
    transportation = t_transport;
}

void industrial_product::set_storage(string &t_storage)
{
    storage = t_storage;
}

industrial_producth:

```

```
#pragma once
#include "product.h"

class industrial_product:public product {
private:
    string transportation;
    string storage;

public:
    double summary_price_check() override;
    industrial_product(product &base_product, string
&t_transportation, string &t_storage)
:product(base_product),

transportation(t_transportation),

storage(t_storage){}
    void set_transport(string &t_transport);
    void set_storage(string &t_storage);
};
```

Тестування:

```
Enter quantity of industrial products
1
Enter quantity of grocery products
1
Creating of industrial products...
Enter product`s name:
Dress
Enter the date of manufacture in format dd mm yyyy:
15
5
2023
Enter product`s price (per unit):
150
Enter amount of product
3
Choose type of transporting:
(0) by cargo van
(1) by plane
(2) by train
(3) by ship
0
Choose storage
(0)-warehouse
(1)-hall
0
```

```
Creating of grocery products...
Enter product`s name:
Milk
Enter the date of manufacture in format dd mm yyyy:
10
3
2023
Enter product`s price (per unit):
25
Enter amount of product
6
Enter the expiration date in format dd mm yyyy
10
0
0
Total price of overdue products equals 150
Total price of industrial products in warehouse equals 450
```