Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни
«Основи програмування-2.
Методології програмування.»

«Класи та об’єкти»

Варіант 28

Виконав студент ІП-11 Сідак Кирил Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №3

Мета: вивчити механізми створення і використання класів та об’єктів.

Варіант 28

Розробити клас «Продукт», який характеризується найменуванням, датою випуску, кінцевим терміном придатності(у форматі ММ-ДД-РРРР). Створити масив об’єктів даного класу. Визначити продукти, термін яких закінчився(на вказану дату).

Постановка задачі: За умовою задачі треба розробити клас «Продукт» із атрибутами найменування рядкового типу, датою випуску рядкового типу або власного створеного типу та кінцевим терміном придатності цього ж типу. Введення дат повинно відбуватися у форматі ММ-ДД-РРРР. На основі введених продуктів треба створити їх масив та серед продуктів визначити ті, термін яких закінчився, тобто термін придатності вже вийшов на основі введеної користувачем дати.

Програма на С++:

main.cpp

```
#include "product.h"
#include "product_operations.h"
int main() {
    int n;
    string current_date;
    cout << "Enter the number of products: ";
    cin >> n;
    vector<Product> product_list = get_product_list(n);
    cout << "Products:" << endl;
    output_products(product_list);
    cout << "Enter current date:" << endl;
    getline(cin, current_date);
    while (!is_valid_date(current_date)) {
        cout << "Incorrect date. Please enter a valid date:"
<< endl;
        getline(cin, current_date);
    }
    vector <Product> expired_products =
get_expired_products(current_date, product_list);
    cout << "Expired products:" << endl;
    output_products(expired_products);
    return 0;
}
```

product_operations.h

```
#ifndef LAB_3_PRODUCT_OPERATIONS_H
#define LAB_3_PRODUCT_OPERATIONS_H
#include "product.h"
#include <iostream>
#include <vector>
using namespace std;
```

```
vector<Product> get_product_list(int);
bool is_number(string);
bool is_valid_date(const string&);
vector<string> split(string, char);
vector<Product> get_expired_products(const string&, const
vector<Product>&);
void output_products(const vector<Product>&);
bool is_earlier(const Date&, const Date&);
#endif
```

product_operations.cpp

```
#include "product_operations.h"

vector<Product> get_product_list(int n) {
    string name, date;
    vector<Product> product_list;
    cin.ignore();
    for (int i = 0; i < n; ++i) {
        cout << "Enter the product name: ";
        getline(cin, name);
        cout << "Enter the release date in such format MM-DD-
YYYY:" << endl;
        getline(cin, date);
        while (!is_valid_date(date)) {
            cout << "Incorrect input for the date. Please
enter a valid date:" << endl;
            getline(cin, date);
        }
        Date release_date(date);
        cout << "Enter the expire date in such format MM-DD-
YYYY:" << endl;
        getline(cin, date);
        while (!is_valid_date(date)) {
            cout << "Incorrect input for the date. Please
enter a valid date:" << endl;
            getline(cin, date);
        }
        Date expire_date(date);
        while (!is_earlier(release_date, expire_date)) {
            cout << "Expire date should be later than release
date. Please enter a valid date:" << endl;
            getline(cin, date);
            expire_date = Date(date);
        }
        product_list.emplace_back(Product(name, release_date,
expire_date));
    }
    return product_list;
}
```

```
vector<Product> get_expired_products(const string& date,
const vector<Product>& products) {
    vector<Product> exp_products;
    Date current_date(date);
    for (const Product& pr: products) {
        Date expire_date = pr.get_expire_date();
        if (is_earlier(expire_date, current_date)) {
            exp_products.push_back(pr);
        }
    }
    return exp_products;
}



bool is_valid_date(const string& s) {
    bool is_valid = false;
    char delim = '.';
    if (s.length() == 10) {
        if (s[2] == s[5] && s[2] == delim) {
            vector <string> strings = split(s, delim);
            int i = 0;
            bool is_num = true;
            while (i < strings.size() && is_num) {
                is_num = is_number(strings[i]);
                i++;
            }
            if (is_num) {
                int month = stoi(strings[0]);
                int year = stoi(strings[2]);
                bool is_month = month > 0 && month <= 12;
                bool is_year = year >= 1000 and year < 10000;
                if (is_month && is_year) {
                    int day = stoi(strings[1]);
                    if (day > 0 and day <= 31) {
                        vector <int> days_month = {31, 28,
31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
                        bool is_fourth = !(year % 4);
                        if (is_fourth) {
                            days_month[1] = 29;
                        }
                        if (day <= days_month[month - 1]) {
                            is_valid = true;
                        }
                    }
                }
            }
        }

    }
    return is_valid;
```

```
}


bool is_number(string str) {
    int i = 0;
    bool is_num = true;
    while (i < str.length() && is_num) {
        if (!isdigit(str[i])) is_num = false;
        i++;
    }
    return is_num;
}

vector<string> split(string str, char delim=' ') {
    vector<string> words;
    string temp_str;
    for (int i = 0; i < str.length(); ++i) {
        if (str[i] == delim || i == str.length() - 1) {
            if (str[i] == delim) {
                words.push_back(temp_str);
                temp_str = "";
            } else {
                temp_str += str[i];
                if (!temp_str.empty())
words.push_back(temp_str);
            }
        } else {
            temp_str += str[i];
        }
    }
    return words;
}

void output_products(const vector<Product>& products) {
    if (products.empty()) cout << "The list is empty.";
    for (Product pr: products) {
        pr.print();
    }
}

bool is_earlier (const Date& date_1, const Date& date_2) {
    bool check = true;
    int days_1 = date_1.get_day();
    int days_2 = date_2.get_day();
    int months_1 = date_1.date_in_months();
    int months_2 = date_2.date_in_months();
    if (months_1 > months_2 || (months_1 == months_2 &&
days_1 > days_2)) {
        check = false;
    }
```

```
    return check;
}
```

product.h

```
#ifndef LAB_3_PRODUCT_H
#define LAB_3_PRODUCT_H
#include "date.h"
#include <string>
#include <iostream>
using namespace std;
class Product {
    string name;
    Date release_date;
    Date expire_date;
public:
    Product(string, Date, Date);
    Date get_release_date() const { return release_date; }
    Date get_expire_date() const { return expire_date; }
    void print() const;
};
#endif
```

product.cpp

```
#include "product.h"
Product::Product(string name, Date release_date, Date
expire_date) {
    this -> name = name;
    this -> release_date = release_date;
    this -> expire_date = expire_date;
}

void Product::print() const {
    string r_date = release_date.get_date_str();
    string e_date = expire_date.get_date_str();
    cout << "Product name: " << name << "; release date: " <<
r_date << "; expire date: " << e_date << endl;
}
```

date.h

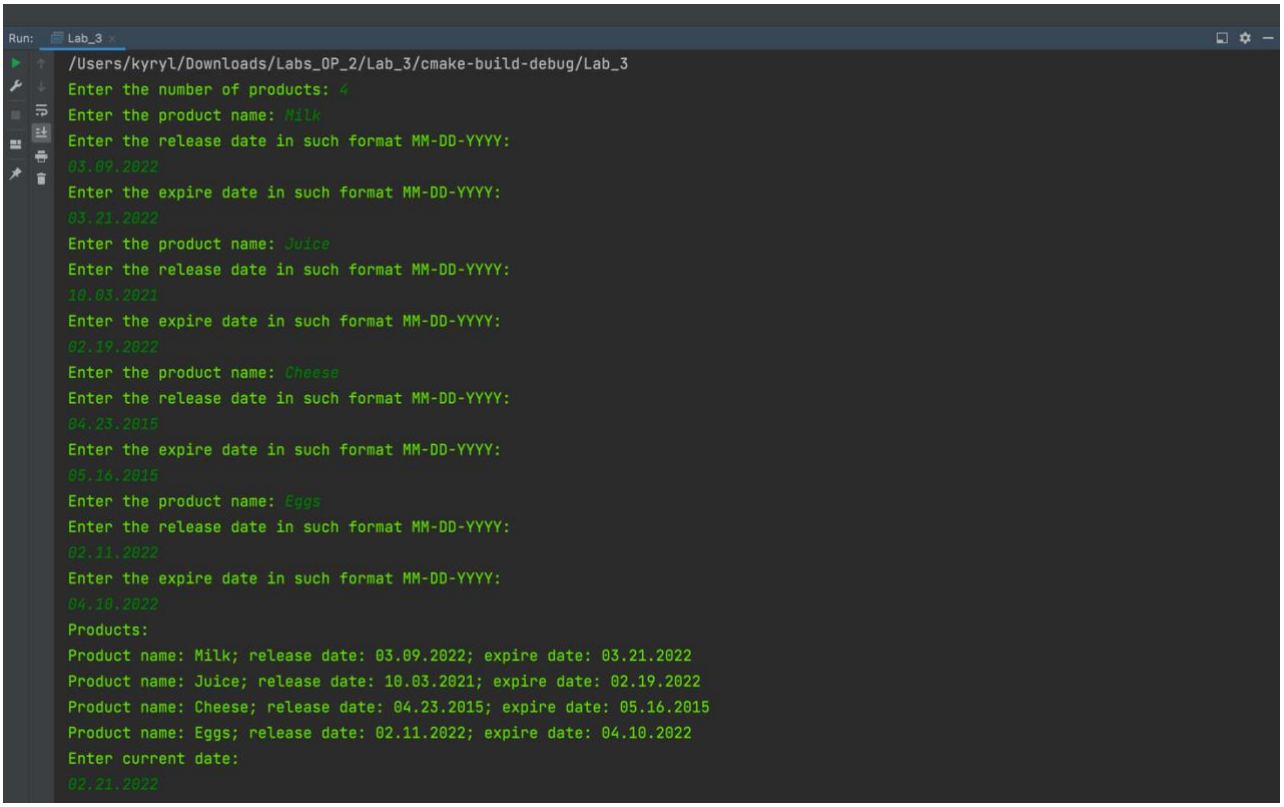
```
#ifndef LAB_3_DATE_H
#define LAB_3_DATE_H
#include <string>
#include <iostream>
using namespace std;
class Date {
    int day, month, year;
    string date_str;
public:
    Date() = default;
```

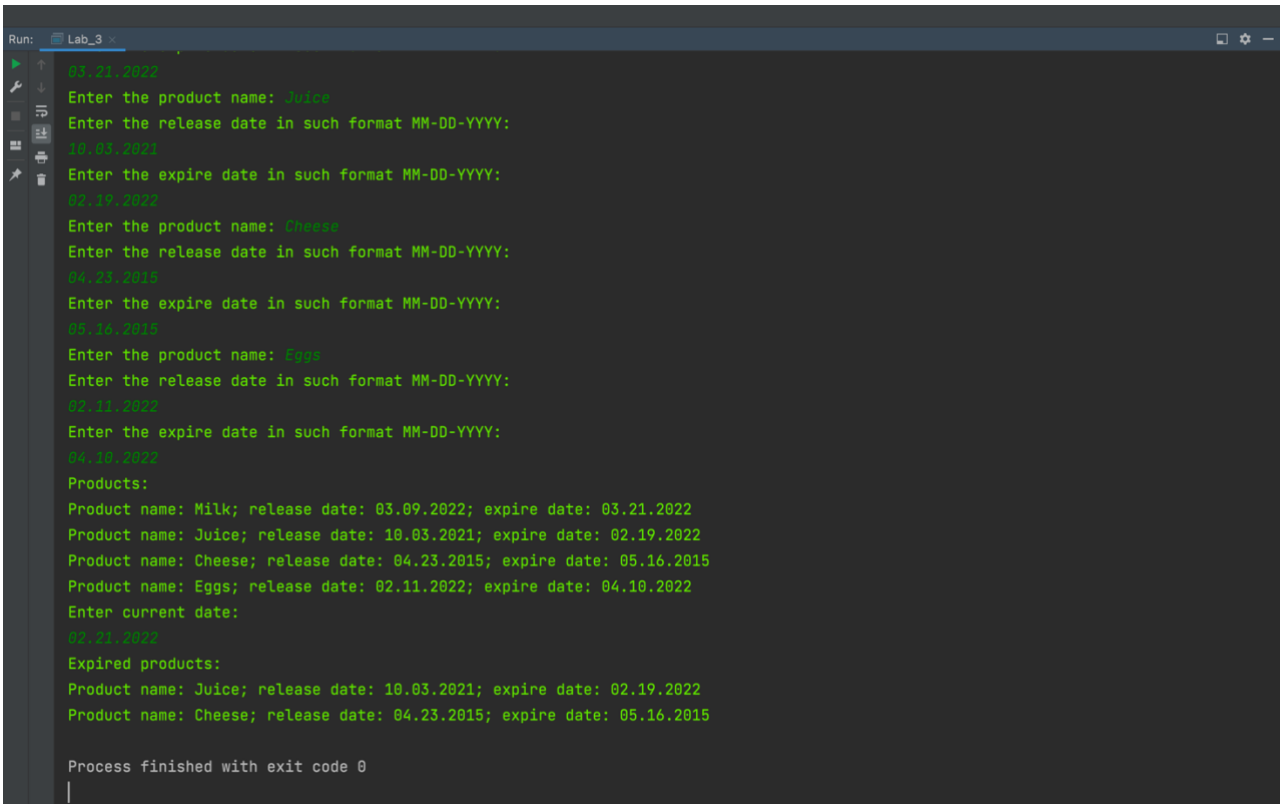
```
    Date(const string&);
    string get_date_str() const { return date_str;}
    int get_day() const { return day; }
    int get_month() const { return month; }
    int get_year() const { return year; }
    int date_in_months() const { return 12 * year + month; }
};
#endif
```

date.cpp

```
#include "date.h"
Date::Date(const string& s)
{
    this -> month = stoi(s.substr(0, 2));
    this -> day = stoi(s.substr(3, 2));
    this -> year = stoi(s.substr(6, 4));
    date_str = s;
}
```

Результат на C++:





Висновок

Отже, я вивчив механізми створення і використання класів та об’єктів на прикладі мови C++, створивши клас продукт та дата, які містили приватні атрибути різних типів, публічні методи та конструктори, зокрема, клас дата містив як конструктор за замовчуванням, так і перевантажений конструктор з параметрами з метою ініціалізації атрибутів. Створивши список продуктів та список продуктів, у яких термін придатності вже закінчився, я отримав коректний результат.