

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 №5 з дисципліни

«Основи програмування

2. Модульне програмування»

«Успадкування та поліморфізм»

Виконав студент ІП-14 Нікулін Павло Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірів Вітковська Ірина Іванівна
(прізвище, ім'я, по батькові)

Лабораторна робота №5

Успадкування та поліморфізм

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

Хід роботи

Задача: 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 дату продукту. Визначити загальну вартість харчових продуктів, термін зберігання яких закінчився, і загальну вартість промислових товарів, які знаходяться на складі.

Розв'язання

1. **Постановка задачі:** Створимо клас з атрибутами та методами та два нащадки з додатковими атрибутами. Реалізуємо функції створення товарів, виведення списку з них та обчислення необхідних величин. Результатом роботи мають бути результати останньої функції.

2. Побудова **математичної моделі**. Складемо таблицю імен змінних для двох мов.

C++

Змінна	Тип	Ім'я	Призначення
Хедер до файлу з функціями	Файловий	FUNCS.H	Файл
Файл з функціями	Файловий	FUNCS.CPP	Файл
		GOODS.H	Файл
		MGOODS.H	Файл
		MGOODS.CPP	Файл
		FGOODS.H	Файл
		FGOODS.CPP	Файл
Створення товарів	Універсальний	INIT_GOODS	Функція
Виведення товарів	Універсальний	OUTPUT_GOODS	Функція

Обчислення	Універсальний	CALCULATIONS	Функція
Список промислових товарів	Класовий	MANUFACTURE_GOODS/ MG	Початкове дане
Список харчових продуктів	Класовий	FOOD_GOODS/FG	Початкове дане
Кількість типів товарів	Цілий	N/M	Початкове дане
Змінні для відповідних атрибутів класу	Рядковий	ITEM_NAME/DATE/ TRANSPORTATION/ LOCATION/ END_DATE	Проміжне дане
Змінні для відповідних атрибутів класу	Цілий	ITEM_PRICE/ AMOUNT	Проміжне дане
Об'єкт класу	Класовий	M_OBJ/F_OBJ	Проміжне дане
Загальна вартість	Цілий	TOTAL_PRICE	Результат

Класи C++

Ім'я	Тип	Опис
GOODS	Клас	Клас-предок
GOODS	Конструктор Goods	Конструктор класу
SUM_PRICE	Метод Goods	Обчислення загальної суми
OUTPUT	Віртуальний метод Goods	Виведення товару
NAME	Атрибут Goods/Рядковий	Ім'я товару
DATE	Атрибут Goods/Рядковий	Дата виготовлення
PRICE	Атрибут Goods/Цілий	Ціна товару
AMOUNT	Атрибут Goods/Цілий	Кількість товару
MGOODS	Клас	Клас-нащадок
MGOODS	Конструктор MGoods	Конструктор класу
OUTPUT	Метод MGoods	Виведення товару
GET_STATUS	Метод MGoods	Доступ до статусу
TRANSPORTING	Атрибут MGoods/Рядковий	Метод транспортування
STATUS	Атрибут MGoods/Рядковий	Місце знаходження товару
FGOODS	Клас	Клас-нащадок
FGOODS	Конструктор FGoods	Конструктор класу
OVERDUE_CHECK	Метод FGoods	Перевірка дати придатності

OUTPUT	Метод FGoods	Виведення товару
END_DATE	Атрибут FGoods/Рядковий	Дата придатності

Python

Змінна	Тип	Ім'я	Призначення
Файл з функціями	Файловий	FUNCS.PY	Файл
Створення товарів	Мультикласовий	INIT_GOODS	Функція
Виведення товарів	Універсальний	OUTPUT_GOODS	Функція
Обчислення	Універсальний	CALCULATIONS	Функція
Список промислових товарів	Класовий	MANUFACTURE_GOODS/ MG	Початкове дане
Список харчових продуктів	Класовий	FOOD_GOODS/FG	Початкове дане
Кількість типів товарів	Цілий	N/M	Початкове дане
Товар у списку	Класовий	ITEM	Проміжне дане
Поточна дата	Цілий	CURR_DATE	Проміжне дане
Загальна вартість	Цілий	TOTAL_PRICE	Результат

Класи Python

Ім'я	Тип	Опис
GOODS	Клас	Клас-предок
__INIT__	Конструктор	Конструктор класів
SUM_PRICE	Метод Goods	Обчислення загальної суми
OUTPUT	Метод Goods	Виведення товару
NAME	Атрибут Goods	Ім'я товару
DATE	Атрибут Goods	Дата виготовлення
PRICE	Атрибут Goods	Ціна товару
AMOUNT	Атрибут Goods	Кількість товару
MGOODS	Клас	Клас-нащадок
TRANSPORTING	Атрибут MGoods	Метод транспортування
STATUS	Атрибут MGoods	Місце знаходження товару
FGOODS	Клас	Клас-нащадок

OVERDUE_CHECK	Метод FGoods	Перевірка дати придатності
END_DATE	Атрибут FGoods	Дата придатності

Випробування коду

C++

Код

lab1.cpp

funcs.h

```

1  #include <iostream>
2  #include <cstring>
3  #include <vector>
4  #include "funcs.h"
5  using namespace std;
6
7  int main()
8  {
9      vector<MGoods> manufacture_goods;
10     vector<FGoods> food_goods;
11
12     init_goods(manufacture_goods, food_goods);
13     output_goods(manufacture_goods, food_goods);
14     calculations(manufacture_goods, food_goods);
15
16     cout << endl;
17 }
```

```

1  #pragma once
2  #include <vector>
3  #include "MGoods.h"
4  #include "FGoods.h"
5
6  void init_goods(vector<MGoods>&, vector<FGoods>&);
7  void output_goods(vector<MGoods>&, vector<FGoods>&);
8  void calculations(vector<MGoods>&, vector<FGoods>&);
```

funcs.cpp

```

1  #include <iostream>
2  #include <cstring>
3  #include <vector>
4  #include "MGoods.h"
5  #include "FGoods.h"
6  using namespace std;
7
8  void init_goods(vector<MGoods>& mg, vector<FGoods>& fg)
9  {
10     string item_name, item_date, item_transportation, item_location, item_end_date;
11     int item_price, item_amount;
12
13     int n;
14     cout << "Amount of manufacture goods: ";
15     cin >> n;
16
17     for (int i = 0; i < n; i++)
18     {
19         cout << "Item name: ";
20         cin >> item_name;
21         cout << "Item manufacture date (dd.mm.yyyy): ";
22         cin >> item_date;
23         cout << "Item price: ";
24         cin >> item_price;
25         cout << "Item amount: ";
26         cin >> item_amount;
27         cout << "Item transportation method: ";
28         cin >> item_transportation;
29         cout << "Item location (warehouse, store): ";
30         cin >> item_location;
31
32         MGoods m_obj(item_name, item_date, item_price, item_amount, item_transportation, item_location);
33         mg.push_back(m_obj);
34
35         cout << endl;
36     }
37
38     int m;
39     cout << "Amount of food goods: ";
40     cin >> m;
41
42     for (int i = 0; i < m; i++)
43     {
44         cout << "Item name: ";
45         cin >> item_name;
46         cout << "Item manufacture date (dd.mm.yyyy): ";
47         cin >> item_date;
48         cout << "Item price: ";
49         cin >> item_price;
50         cout << "Item amount: ";
51         cin >> item_amount;
```

```

52         cout << "item end date (dd.mm.yyyy): ";
53         cin >> item_end_date;
54
55         FGoods f_obj(item_name, item_date, item_price, item_amount, item_end_date);
56         fg.push_back(f_obj);
57
58         cout << endl;
59     }
60 }
61
62 void output_goods(vector<MGoods>& mg, vector<FGoods>& fg)
63 {
64     cout << "Manufacture goods:\n";
65     for (int i = 0; i < mg.size(); i++)
66     {
67         mg[i].output();
68         cout << endl;
69     }
70
71     cout << "\nFood goods:\n";
72     for (int i = 0; i < fg.size(); i++)
73     {
74         fg[i].output();
75         cout << endl;
76     }
77 }
78
79 void calculations(vector<MGoods>& mg, vector<FGoods>& fg)
80 {
81     int total_price = 0;
82     for (int i = 0; i < fg.size(); i++)
83     {
84         if (fg[i].overdue_check() == 1)
85         {
86             total_price += fg[i].sum_price();
87         }
88     }
89
90     printf("\ntotal price of expired food goods: %d$", total_price);
91
92     total_price = 0;
93     for (int i = 0; i < mg.size(); i++)
94     {
95         if (mg[i].get_status() == "warehouse")
96         {
97             total_price += mg[i].sum_price();
98         }
99     }
100     printf("\ntotal price of manufacture goods in warehouse: %d$", total_price);
101 }

```

Goods.h

```

1     #pragma once
2     #include <iostream>
3     #include <cstring>
4     #include <sstream>
5     #include <vector>
6     #include <windows.h>
7     using namespace std;
8
9     class Goods
10    {
11    public:
12        int sum_price()
13        {
14            return price * amount;
15        }
16
17    protected:
18        string name;
19        string date;
20        int price;
21        int amount;
22
23        virtual void output() = 0;
24    };

```

MGoods.h

```
1 #pragma once
2 #include "Goods.h"
3
4 class MGoods : public Goods
5 {
6 private:
7     string transporting;
8     string status;
9 public:
10     MGoods(string c_name, string c_date, int c_price, int c_amount, string c_transporting, string c_status);
11     void output();
12     string get_status();
13 };
```

MGoods.cpp

```
1 #include "MGoods.h"
2
3 MGoods::MGoods(string c_name, string c_date, int c_price, int c_amount, string c_transporting, string c_status)
4 {
5     name = c_name;
6     date = c_date;
7     price = c_price;
8     amount = c_amount;
9     transporting = c_transporting;
10    status = c_status;
11 }
12
13 void MGoods::output()
14 {
15     printf("%s %s %d$ %d %s", name.c_str(), date.c_str(), price, amount, transporting.c_str(), status.c_str());
16 }
17
18 string MGoods::get_status()
19 {
20     return status;
21 }
```

FGoods.h

```
1 #pragma once
2 #include "Goods.h"
3 using namespace std;
4
5 class FGoods : public Goods
6 {
7 private:
8     string end_date;
9 public:
10     FGoods(string c_name, string c_date, int c_price, int c_amount, string c_end_date);
11     void output();
12     int overdue_check();
13 };
```

FGoods.cpp

```
1 #include "FGoods.h"
2
3 FGoods::FGoods(string c_name, string c_date, int c_price, int c_amount, string c_end_date)
4 {
5     name = c_name;
6     date = c_date;
7     price = c_price;
8     amount = c_amount;
9     end_date = c_end_date;
10 }
11
12 void FGoods::output()
13 {
14     printf("%s %s %d$ %d %s", name.c_str(), date.c_str(), price, amount, end_date.c_str());
15 }
16
17 int FGoods::overdue_check()
18 {
19     istringstream date_stream(end_date);
20     vector<int> ed;
21     string date_el;
22     while (getline(date_stream, date_el, '.')) {
23         if (!date_el.empty()) {
24             ed.push_back(stoi(date_el));
25         }
26     }
27
28     SYSTEMTIME curr_date;
29     GetLocalTime(&curr_date);
30
31     if (ed[2] < curr_date.wYear)
32     {
33         return 1;
34     }
35     else if (ed[2] == curr_date.wYear)
36     {
37         if (ed[1] < curr_date.wMonth)
38         {
39             return 1;
40         }
41         else if (ed[1] == curr_date.wMonth)
42         {
43             if (ed[0] <= curr_date.wDay)
44             {
45                 return 1;
46             }
47             else
48             {
49                 return 0;
50             }
51         }
52         else
53         {
44             return 0;
55         }
56     }
57     else
58     {
59         return 0;
60     }
61 }
62 }
```

Результат

```
Amount of manufacture goods: 3
Item name: axe
Item manufacture date (dd.mm.yyyy): 10.10.2010
Item price: 10
Item amount: 10
Item transportation method: truck
Item location (warehouse, store): warehouse

Item name: shovel
Item manufacture date (dd.mm.yyyy): 21.03.2015
Item price: 13
Item amount: 7
Item transportation method: truck
Item location (warehouse, store): store

Item name: hummer
Item manufacture date (dd.mm.yyyy): 15.09.2013
Item price: 9
Item amount: 15
Item transportation method: plane
Item location (warehouse, store): warehouse

Amount of food goods: 2
Item name: cake
Item manufacture date (dd.mm.yyyy): 27.05.2022
Item price: 5
Item amount: 5
Item end date (dd.mm.yyyy): 31.05.2022

Item name: salad
Item manufacture date (dd.mm.yyyy): 25.05.2022
Item price: 3
Item amount: 8
Item end date (dd.mm.yyyy): 28.05.2022

Manufacture goods:
axe 10.10.2010 10$ 10 truck warehouse
shovel 21.03.2015 13$ 7 truck store
hummer 15.09.2013 9$ 15 plane warehouse

Food goods:
cake 27.05.2022 5$ 5 31.05.2022
salad 25.05.2022 3$ 8 28.05.2022

Total price of expired food goods: 24$
Total price of manufacture goods in warehouse: 235$
```


Python

Kood

funcs.py

```
1  from datetime import datetime
2
3  class Goods:
4      def __init__(self, name, date, price, amount):
5          self.name = name
6          self.date = date
7          self.price = price
8          self.amount = amount
9
10     def sum_price(self):
11         return self.price * self.amount
12
13     def output(self):
14         if isinstance(self, MGoods):
15             print("%s %s %d$ %d %s %s" % (self.name, '.'.join(self.date), self.price, self.amount, self.transporting, self.status))
16         elif isinstance(self, FGoods):
17             print("%s %s %d$ %d %s" % (self.name, '.'.join(self.date), self.price, self.amount, '.'.join(self.end_date)))
18
19     class MGoods(Goods):
20         def __init__(self, name, date, price, amount, transporting, status):
21             super().__init__(name, date, price, amount)
22             self.transporting = transporting
23             self.status = status
24
25     class FGoods(Goods):
26         def __init__(self, name, date, price, amount, end_date):
27             super().__init__(name, date, price, amount)
28             self.end_date = end_date
29
30     def overdue_check(self, cd):
31         if self.end_date[2] < cd[2]:
32             return 1
33         elif self.end_date[2] == cd[2]:
34             if self.end_date[1] < cd[1]:
35                 return 1
36             elif self.end_date[1] == cd[1]:
37                 if self.end_date[0] <= cd[0]:
38                     return 1
39                 else:
40                     return 0
41             else:
42                 return 0
43         else:
```

```

    return 0

def init_goods():
    m = int(input("Amount of manufacture goods: "))
    mg = []
    for _ in range(m):
        mg.append(MGoods(input("Item name: "), [x for x in input("Item manufacture date (dd.mm.yyyy): ").split('.')] , int(input("Item price: ")), int(input("Item amount: ")), input("Item transportation method: "), input("Item location (warehouse, store): ")))
        print()

    f = int(input("Amount of food goods: "))
    fg = []
    for _ in range(f):
        fg.append(FGoods(input("Item name: "), [x for x in input("Item manufacture date (dd.mm.yyyy): ").split('.')] , int(input("Item price: ")), int(input("Item amount: ")), [x for x in input("Item end date (dd.mm.yyyy): ").split('.')]))
        print()

    return mg, fg

def output_goods(mg, fg):
    print("Manufacture goods:")
    for item in mg:
        item.output()

    print("\nFood goods:")
    for item in fg:
        item.output()

def calculations(mg, fg):
    curr_date = [x for x in datetime.today().strftime('%d.%m.%Y').split('.')]
    total_price = 0

    for item in fg:
        if item.overdue_check(curr_date) == 1:
            total_price += item.sum_price()

    print("\nTotal price of expired food goods: " + str(total_price) + "$")

    total_price = 0
    for item in mg:
        if item.status == "warehouse":
            total_price += item.sum_price()

    print("Total price of manufacture goods in warehouse: " + str(total_price) + "$")

```

main.py

```

1  from funcs import *
2
3  manufacture_goods, food_goods = init_goods()
4  output_goods(manufacture_goods, food_goods)
5  calculations(manufacture_goods, food_goods)
6

```

Результат

```

Amount of manufacture goods: 3
Item name: axe
Item manufacture date (dd.mm.yyyy): 10.10.2010
Item price: 10
Item amount: 10
Item transportation method: truck
Item location (warehouse, store): warehouse

Item name: shovel
Item manufacture date (dd.mm.yyyy): 21.03.2015
Item price: 13
Item amount: 7
Item transportation method: truck
Item location (warehouse, store): store

Item name: hummer
Item manufacture date (dd.mm.yyyy): 15.09.2013
Item price: 9
Item amount: 15
Item transportation method: plane
Item location (warehouse, store): warehouse

Amount of food goods: 3
Item name: cake
Item manufacture date (dd.mm.yyyy): 27.05.2022
Item price: 5
Item amount: 5
Item end date (dd.mm.yyyy): 31.05.2022

Item name: salad
Item manufacture date (dd.mm.yyyy): 25.05.2022
Item price: 8
Item amount: 3
Item end date (dd.mm.yyyy): 28.05.2022

Manufacture goods:
axe 10.10.2010 10$ 10 truck warehouse
shovel 21.03.2015 13$ 7 truck store
hummer 15.09.2013 9$ 15 plane warehouse

Food goods:
cake 27.05.2022 5$ 5 31.05.2022
salad 25.05.2022 8$ 3 28.05.2022

Total price of expired food goods: 24$
Total price of manufacture goods in warehouse: 235$

```

Висновок

Під час виконання лабораторної роботи було досліджено механізми створення і використання класів та об'єктів. Було створено клас *Goods* з атрибутами та методами та два класи-нащадки *MGoods* та *FGoods* з додатковими атрибутами та методами. Опис класів реалізовано у окремих хедерах та *.cpp* файлах у мові *C++*. Існує три функції: для створення товарів, для виведення списків товарів та для обчислень. Роботу виконано на двох мовах програмування (*Python* та *C++*). Програма працює, результат є вірним при будь-яких вхідних значеннях, що відповідають умові задачі.