

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 № 2 з дисципліни
«Основи програмування-2.
Методології програмування»

«Файли даних. Бінарні файли»

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 №1

Мета: вивчити особливості створення і обробки бінарних файлів.

Варіант 28

Створити файл із списком автомобілів автосалону: назва, дата випуску, дата надходження у продаж. Створити список автомобілів, що надійшли у продаж за останній місяць. Вивести інформацію про вживані автомобілі(які були випущені більш ніж за рік до надходження у продаж).

Постановка задачі:

За умовою задачі треба створити файл зі список автомобілів автосалону, ввівши з клавіатури їхню кількість та назву, дату випуску та дату надходження у продаж для кожного авто.

Потім, зчитавши цей файл, потрібно створити список автомобілів, що надійшли у продаж за останній місяць та записати ці дані у новий файл. Знову, зчитавши початковий файл, треба вивести інформацію про вживані автомобілі(які були випущені більш ніж за рік до надходження у продаж).

Програма на C++:

main.cpp

```
#include "file_operations.h"

int main() {
    int n, month, year;
    string file_1 =
"/Users/kyryl/Downloads/Labs_OP_2/Lab_2/Lab_2_C++/cars_1.txt"
;
    string file_2 =
"/Users/kyryl/Downloads/Labs_OP_2/Lab_2/Lab_2_C++/cars_2.txt"
;

    string file_mode = enter_file_mode();
    cout << "Enter the number of cars:" << endl;
    cin >> n;
    create_fist_file(file_1, n, file_mode);
    cout << "List of cars(first file):" << endl;
    output_file(file_1);
    cout << "Enter the month number:" << endl;
    cin >> month;
    cout << "Enter the year:" << endl;
    cin >> year;
    create_list_recent_cars(file_1, file_2, month, year);
    cout << "List of cars which started selling this
month(second file):" << endl;
    output_file(file_2);
    output_worn_cars(file_1);
    return 0;
}
```

file_operations.h

```
#ifndef LAB_2_C___FILE_OPERATIONS_H
#define LAB_2_C___FILE_OPERATIONS_H
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
using namespace std;

struct Car {
    string model;
    string release_date;
    string sell_date;
};

string enter_file_mode();
void create_fist_file(string, int, string);
void output_file(string);
void create_list_recent_cars(string, string, int, int);
void output_worn_cars(string);
void output_car(Car);
vector<string> split_date(string);
#endif
```

file_operations.cpp

```
#include "file_operations.h"

string enter_file_mode() {
    string file_mode;
    cout << "Do you want to overwrite the file or append  
input to it? Enter w or a:" << endl;
    cin >> file_mode;
    while (file_mode != "w" and file_mode != "a") {
        cout << "Incorrect input. Enter 'w' or 'a'." << endl;
        cout << "Do you want to overwrite the file or append  
input to it? Enter w or a:" << endl;
        cin >> file_mode;
    }
    return file_mode;
}

void create_fist_file(string file_name, int n, string mode) {
    ofstream file;
    if (mode == "w") {
        file.open(file_name, ios::binary);
    }
    else {
        file.open(file_name, ios::binary | ios::app);
    }
    Car car;
    cin.ignore();
}
```

```

        for (int i = 0; i < n; ++i) {
            cout << "Car " << i+1 << endl;
            cout << "Enter the model of the car: ";
            getline(cin, car.model);
            cout << "Enter the date when this car was released in
such format MM.YYYY : ";
            getline(cin, car.release_date);
            cout << "Enter the sale start date in such format
MM.YYYY : ";
            getline(cin, car.sell_date);
            file.write((char*)&car, sizeof(Car));
        }
        file.close();
    }

void output_file(string file_name) {
    Car car;
    ifstream file(file_name, ios::binary);
    cout << "-----" << endl;
    while (file.read((char*)&car, sizeof(Car))) {
        output_car(car);
    }
    cout << "-----" << endl;
    file.close();
}

void create_list_recent_cars(string file_name1, string
file_name2, int current_month, int current_year) {
    Car car;
    vector<Car> car_list;
    ifstream file_1(file_name1, ios::binary);
    while (file_1.read((char*)&car, sizeof(Car))) {
        car_list.push_back(car);
    }
    file_1.close();
    vector<string> month_year;
    vector<Car> recent_cars;
    int sell_month, sell_year;
    for(Car c: car_list) {
        month_year = split_date(c.sell_date);
        sell_month = stoi(month_year[0]);
        sell_year = stoi(month_year[1]);
        if (sell_month == current_month && sell_year ==
current_year) {
            recent_cars.push_back(c);
        }
    }
    ofstream file_2(file_name2, ios::binary);
    for (Car c: recent_cars) {

```

```

        file_2.write((char*)&c, sizeof(Car));
    }
    file_2.close();
}

void output_worn_cars(string file_name) {
    Car car;
    vector<Car> car_list;
    ifstream file(file_name, ios::binary);
    while (file.read((char*)&car, sizeof(Car))) {
        car_list.push_back(car);
    }
    file.close();
    vector<Car> worn_cars;
    vector<string> month_year_sell, month_year_release;
    int sell_month, sell_year, release_month, release_year,
    delta_year, delta_month;
    for (Car c: car_list) {
        month_year_sell = split_date(c.sell_date);
        sell_month = stoi(month_year_sell[0]);
        sell_year = stoi(month_year_sell[1]);
        month_year_release = split_date(c.release_date);
        release_month = stoi(month_year_release[0]);
        release_year = stoi(month_year_release[1]);
        delta_year = sell_year - release_year;
        delta_month = sell_month - release_month;
        if (delta_year > 1 || (delta_month > 0 && delta_year
== 1)) {
            worn_cars.push_back(c);
        }
    }
    cout << "Worn cars (more than one year):" << endl;
    cout << "-----" << endl;
    for (Car c: worn_cars) {
        output_car(c);
    }
    cout << "-----" << endl;
    cout << "-----" << endl;
}

void output_car(Car car) {
    string info = "Car model: " + car.model + ", release
date: " + car.release_date + ", sell start date: " +
car.sell_date;
    cout << info << endl;
}

vector<string> split_date(string str) {
    vector<string> month_year;

```

```

string temp;
for (int i = 0; i < str.length(); ++i) {
    if (str[i] == '.') {
        month_year.push_back(temp);
        temp = "";
    }
    else {
        temp += str[i];
    }
}
month_year.push_back(temp);
return month_year;
}

```

Програма на Python:

main.py

```

from file_operations import *

def main():
    file_1 = 'cars_1.txt'
    file_2 = 'cars_2.txt'
    file_mode = enter_file_mode()
    n = int(input('Enter the number of cars: '))
    create_fist_file(file_1, n, file_mode)
    print('List of cars(first file):')
    output_file(file_1)
    month = int(input('Enter the month number: '))
    year = int(input('Enter the year: '))
    create_list_recent_cars(file_1, file_2, month, year)
    print(f'List of cars which started selling this
month(second file): ')
    output_file(file_2)
    output_worn_cars(file_1)

if __name__ == '__main__':
    main()

```

file_operations.py

```

import pickle

class Car:
    def __init__(self, model: str, release_date: str,
sell_date: str):
        self.model = model

```

```

        self.release_date = release_date
        self.sell_date = sell_date

    def __str__(self):
        info = f'Car model: {self.model}, release date: {self.release_date}, sell start date: {self.sell_date}'
        return info

def enter_file_mode():
    file_mode = input('Do you want to overwrite the file or append input to it? Enter w or a: ')
    while file_mode != 'w' and file_mode != 'a':
        print("Incorrect input. Enter 'w' or 'a'.")
        file_mode = input('Do you want to overwrite the file or append input to it? Enter w or a: ')
    return file_mode

def create_fist_file(file_name: str, n: int, mode: str):
    if mode == 'w':
        file_mode = 'wb'
    else:
        file_mode = 'ab'
    car_list = list()
    for i in range(n):
        print(f'Car {i+1}:')
        model = input('Enter the model of the car: ')
        release_date = input('Enter the date when this car was released in such format '
                               'MM.YYYY : ')
        sell_date = input('Enter the sale start date in such format MM.YYYY : ')
        car = Car(model, release_date, sell_date)
        car_list.append(car)
    with open(file_name, file_mode) as file:
        pickle.dump(car_list, file)

def create_list_recent_cars(file_name_1: str, file_name_2: str, current_month: int, current_year: int):
    with open(file_name_1, 'rb') as file:
        car_list = pickle.load(file)
    recent_cars = list()
    for car in car_list:
        month_year = car.sell_date.split('.')
        sell_month = int(month_year[0])
        sell_year = int(month_year[1])
        if sell_month == current_month and sell_year == current_year:
            recent_cars.append(car)

```

```

with open(file_name_2, 'wb') as file:
    pickle.dump(recent_cars, file)

def output_worn_cars(file_name: str):
    with open(file_name, 'rb') as file:
        car_list = pickle.load(file)
    worn_cars = list()
    for car in car_list:
        month_year_sell = car.sell_date.split('.')
        sell_month = int(month_year_sell[0])
        sell_year = int(month_year_sell[1])
        month_year_release = car.release_date.split('.')
        release_month = int(month_year_release[0])
        release_year = int(month_year_release[1])
        delta_year = sell_year - release_year
        delta_month = sell_month - release_month
        if delta_year > 1 or (delta_month > 0 and delta_year
== 1):
            worn_cars.append(car)
    print('Worn cars (more than one year):')
    print('-' * 70)
    for car in worn_cars:
        print(car)
    print('-' * 70)

def output_file(file_name: str):
    with open(file_name, 'rb') as file:
        car_list = pickle.load(file)
    print('-' * 70)
    for car in car_list:
        print(car)
    print('-' * 70)

```


Результат на C++:

```
Run: Lab_2_C_...
/Users/kyryl/Downloads/Labs_OP_2/Lab_2/Lab_2_C++/cmake-build-debug/Lab_2_C_...
Do you want to overwrite the file or append input to it? Enter w or a:
w
Enter the number of cars:
4
Car 1
Enter the model of the car: Hundai
Enter the date when this car was released in such format MM.YYYY : 02.2021
Enter the sale start date in such format MM.YYYY : 04.2022
Car 2
Enter the model of the car: Mercedes
Enter the date when this car was released in such format MM.YYYY : 09.2019
Enter the sale start date in such format MM.YYYY : 01.2022
Car 3
Enter the model of the car: Honda
Enter the date when this car was released in such format MM.YYYY : 05.2018
Enter the sale start date in such format MM.YYYY : 02.2019
Car 4
Enter the model of the car: BMW
Enter the date when this car was released in such format MM.YYYY : 01.2022
Enter the sale start date in such format MM.YYYY : 04.2022
List of cars(first file):
-----
Car model: Hundai, release date: 02.2021, sell start date: 04.2022
Car model: Mercedes, release date: 09.2019, sell start date: 01.2022
Car model: Honda, release date: 05.2018, sell start date: 02.2019
Car model: BMW, release date: 01.2022, sell start date: 04.2022
-----
Enter the month number:
```

```
Run: Lab_2_C_...
Enter the sale start date in such format MM.YYYY : 02.2019
Car 4
Enter the model of the car: BMW
Enter the date when this car was released in such format MM.YYYY : 01.2022
Enter the sale start date in such format MM.YYYY : 04.2022
List of cars(first file):
-----
Car model: Hundai, release date: 02.2021, sell start date: 04.2022
Car model: Mercedes, release date: 09.2019, sell start date: 01.2022
Car model: Honda, release date: 05.2018, sell start date: 02.2019
Car model: BMW, release date: 01.2022, sell start date: 04.2022
-----
Enter the month number:
4
Enter the year:
2022
List of cars which started selling this month(second file):
-----
Car model: Hundai, release date: 02.2021, sell start date: 04.2022
Car model: BMW, release date: 01.2022, sell start date: 04.2022
-----
Worn cars(more than one year):
-----
Car model: Hundai, release date: 02.2021, sell start date: 04.2022
Car model: Mercedes, release date: 09.2019, sell start date: 01.2022
-----
Process finished with exit code 0
```

Результат на Python:

```
Run: main
/Users/kyryl/Downloads/Labs_second_term/Lab_2/Lab_2_Python/venv/bin/python
/Users/kyryl/Downloads/Labs_second_term/Lab_2/Lab_2_Python/main.py
Do you want to overwrite the file or append input to it? Enter w or a: w
Enter the number of cars: 4
Car 1:
Enter the model of the car: BMW
Enter the date when this car was released in such format MM.YYYY : 05.2021
Enter the sale start date in such format MM.YYYY : 04.2022
Car 2:
Enter the model of the car: Nissan
Enter the date when this car was released in such format MM.YYYY : 09.2019
Enter the sale start date in such format MM.YYYY : 04.2022
Car 3:
Enter the model of the car: Hyundai
Enter the date when this car was released in such format MM.YYYY : 10.2012
Enter the sale start date in such format MM.YYYY : 11.2013
Car 4:
Enter the model of the car: Audi
Enter the date when this car was released in such format MM.YYYY : 01.2022
Enter the sale start date in such format MM.YYYY : 03.2022
List of cars(first file):
-----
Car model: BMW, release date: 05.2021, sell start date: 04.2022
Car model: Nissan, release date: 09.2019, sell start date: 04.2022
Car model: Hyundai, release date: 10.2012, sell start date: 11.2013
Car model: Audi, release date: 01.2022, sell start date: 03.2022
-----
Enter the month number: 4
Enter the year: 2022
```

```
Run: main
Enter the model of the car: Hyundai
Enter the date when this car was released in such format MM.YYYY : 10.2012
Enter the sale start date in such format MM.YYYY : 11.2013
Car 4:
Enter the model of the car: Audi
Enter the date when this car was released in such format MM.YYYY : 01.2022
Enter the sale start date in such format MM.YYYY : 03.2022
List of cars(first file):
-----
Car model: BMW, release date: 05.2021, sell start date: 04.2022
Car model: Nissan, release date: 09.2019, sell start date: 04.2022
Car model: Hyundai, release date: 10.2012, sell start date: 11.2013
Car model: Audi, release date: 01.2022, sell start date: 03.2022
-----
Enter the month number: 4
Enter the year: 2022
List of cars which started selling this month(second file):
-----
Car model: BMW, release date: 05.2021, sell start date: 04.2022
Car model: Nissan, release date: 09.2019, sell start date: 04.2022
-----
Worn cars(more than one year):
-----
Car model: Nissan, release date: 09.2019, sell start date: 04.2022
Car model: Hyundai, release date: 10.2012, sell start date: 11.2013
-----
Process finished with exit code 0
```

Висновок:

Отже, я вивчив особливості створення і обробки бінарних файлів(операції читання з файлу та запис у файл) на прикладі створення файлу зі списком машин(їхньою назвою, датою виготовлення та датою надходження у продаж), зчитування даних із нього, їхню подальшу обробку, опираючись на постановку задачі, та створення нового файлу із списку машин, що надійшли у продаж за останній місяць. Вивівши отримані дані, я отримав коректний результат.