

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 3

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. Розробити клас "Студент", який характеризується ПІБ студента, номером його групи, датою народження (у форматі ММ-ДД-РРРР). Створити масив об'єктів даного класу. Визначити найстаршого студента вказаної групи (на вказану дату).

## Код програми:

main.cpp:

```
#include "functions.h"
#include "student.h"
using namespace std;

int main() {
    int students_q = students_quantity();
    int size = students_q;
    student *students_array = array_create(size);
    array_write(students_array, size);
    array_read(students_array, size);
    oldest_search(students_array, size);
    array_delete(students_array);

    return 0;
}
```

student.h:

```
#pragma once
#include <iostream>
#include <istream>
#include <fstream>
#include <string>
//#include "student.cpp"
using namespace std;

class Data{
private:
    int day;
```

```

    int month;
    int year;
public:
    //Data();
    void setDay(int valueDay);
    void setMonth(int valueMonth);
    void setYear(int valueYear);
    int getDay() const;
    int getMonth() const;
    int getYear() const;

};

class student
{
private:
    string name;
    int group_num;
    Data birthday;
public:
    //student(string& value_name, Data value_birthday, int value_group_num);
    void setName(string valueName);
    void setGroup_num(const int);
    void setData(Data valueBirthday);
    string getName();
    int getGroup_num() const;
    Data getBirthday();
};

```

student.cpp:

```

#include <iostream>
#include <istream>
#include <fstream>
#include <string>
#include <utility>
#include "student.h"
//#include "functions.h"

```

```

using namespace std;

```

```

void student::setName(string valueName) {
    name = valueName;
}

void student::setGroup_num(const int valueGroup_num) {
    group_num = valueGroup_num;
}

void student::setData(Data valueBirthday) {
    birthday = valueBirthday;
}

string student::getName() {
    return name;
}

int student::getGroup_num() const {
    return group_num;
}

Data student::getBirthday() {
    return birthday;
}

void Data::setDay(int valueDay) {
    day = valueDay;
}

void Data::setMonth(int valueMonth) {
    month = valueMonth;
}

void Data::setYear(int valueYear) {
    year = valueYear;
}

int Data::getDay() const
{
    return day;
}

```

```
int Data::getMonth() const
{
    return month;
}
int Data::getYear() const
{
    return year;
}
```

functions.h:

```
#pragma once
#include <iostream>
#include <istream>
#include <fstream>
#include <string>
#include "student.h"
using namespace std;
int students_quantity();
string getName();
student* array_create(int students_q);
void array_delete(student* arr);
void array_write(student* array, int size);
void array_read(student* array, int size);
int day_check(int month, int year);
int month_check();
int year_check();
void oldest_search(student* array, int size);
int group_choose();
```

functions.cpp:

```
#include <iostream>
#include <istream>
#include <fstream>
#include <string>
#include "functions.h"
using namespace std;

int students_quantity()
```

```

{   int quantity;
    while(1)
    {
        cout<<"Please, enter the quantity of students: \n";
        cin>>quantity;
        if(cin.fail())
        {
            cin.clear();
            cin.sync();
            cout<<"Invalid input\nTry again please\n";
        }
        else {break;}
    }
    return quantity;
}

```

```

student* array_create(int students_q)
{
    auto* arr = new student[students_q];
    return arr;
}

```

```

void array_delete(student *arr)
{
    delete[] arr;
}

```

```

void array_write(student *array, int size)
{
    string tempStr;
    int tempGroup;
    Data tempData;
    int tempDay, tempMonth, tempYear;
    for(int i = 0; i < size; i++)
    {

        cin.ignore();
        student user;
        cout<<"Enter PIB\n";
        getline(cin, tempStr);
        user.setName(tempStr);
    }
}

```

```

        //::fflush(stdin);
        //cin.ignore();
        cout<<"Group number\n";
        cin>>tempGroup;
        user.setGroup_num(tempGroup);
        //::fflush(stdin);
        //cin.ignore();

        //::fflush(stdin);
        //cin.ignore();

        //::fflush(stdin);
        //cin.ignore();
        //cout<<"Year\n";
        //cin>>tempYear;
        tempYear = year_check();
        tempData.setYear(tempYear);
        //cout<<"Month\n";
        //cin>>tempMonth;
        tempMonth = month_check();
        tempData.setMonth(tempMonth);
        //cout<<"Day\n";
        //cin>>tempDay;
        tempDay = day_check(tempMonth, tempYear);
        tempData.setDay(tempDay);
        //::fflush(stdin);
        //cin.ignore();
        user.setData(tempData);
        array[i] = user;
        cout<<"success\n\n";
    }
}

```

```

void array_read(student *array, int size)
{
    string tempStr;
    int tempGroup;
    Data tempData{ };
    int tempDay, tempMonth, tempYear;
    for(int i = 0; i < size; i++)

```

```

{
    cout<<"-----\n";
    cout<<"Student "<<i+1<<": "<<endl;
    cout<<"-----\n";
    student user = array[i];
    tempStr = array[i].getName();
    cout<<"Pib: "<<tempStr<<endl;
    tempGroup = user.getGroup_num();
    cout<<"Group: "<<tempGroup<<endl;
    tempData = array[i].getBirthday();
    tempDay = tempData.getDay();
    tempMonth = tempData.getMonth();
    tempYear = tempData.getYear();
    cout<<"Birthday:
"<<tempDay<<": "<<tempMonth<<": "<<tempYear<<"\n"<<endl;

}
cout<<"*****\n\n";
}

int year_check()
{
    int year;
    while(true)
    {
        cout<<"Enter year:";
        cin>>year;
        if(year < 1923||year > 2023){cout<<"*****\nIncorrect year
value\nTry again\n*****\n";}
        else {break;}
    }
    return year;
}

int month_check()
{
    int month;
    while(true)
    {
        cout<<"Enter month:";

```



```

        cin>>month;
        if(month < 1||month > 12){cout<<"*****\nIncorrect month
value\nTry again\n*****\n";}
        else {break;}
    }
    return month;
}

```

```

int day_check(int month, int year)
{
    int day = 0;

    while(true) {
        int check = 0;
        cout << "Enter day:";
        cin >> day;
        if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
month == 10 || month == 12) {
            if (day > 31) {
                cout << "*****\nIncorrect month value\nTry
again\n*****\n";
                check = 1;
            }
        }
        if (month == 4 || month == 6 || month == 9 || month == 11) {
            if (day > 30) {
                cout << "*****\nIncorrect month value\nTry
again\n*****\n";
                check = 1;
            }
        }
        if (month == 2) {
            if (year % 4 == 0 && day > 29) {
                cout << "*****\nIncorrect month value\nTry
again\n*****\n";
                check = 1;
            }
            if (year % 4 != 0 && day > 28) {
                cout << "*****\nIncorrect month value\nTry
again\n*****\n";

```

```

        check = 1;
    }

    }
    if(check == 0) {break;}
}
return day;
}

void oldest_search(student* array, int size)
{
    student oldest_student;
    string tempStr, oldest_student_name;
    int chosen_group, minDay, minMonth, minYear, minGroup, tempDay,
tempMonth, tempYear, tempGroup, check = 0;
    Data tempData{ }, minData{ };
    oldest_student = array[0];
    chosen_group = group_choose();
    for(int i = 0; i < size; i++)
    {

        student user = array[i];
        //minGroup = oldest_student.getGroup_num();
        minData = oldest_student.getBirthday();
        minDay = minData.getDay();
        minMonth = minData.getMonth();
        minYear = minData.getYear();
        tempGroup = array[i].getGroup_num();
        tempData = array[i].getBirthday();
        tempDay = tempData.getDay();
        tempMonth = tempData.getMonth();
        tempYear = tempData.getYear();
        if(chosen_group == tempGroup)
        {
            check = 1;
            if(double(minYear)*365.25 + double(minMonth)*30.56 +
double(minDay) > double(tempYear)*365.25 + double(tempMonth)*30.56 +
double(tempDay))
            {
                oldest_student = array[i];
            }
        }
    }
}

```

```

        }
    }

}
if(check == 1)
{
    oldest_student_name = oldest_student.getName();
    cout << "Oldest student is: ";
    cout << oldest_student_name;
}
if(check == 0)
{
    cout<<"-----\nChosen group does not exist\n-----
-----";
}
}

int group_choose()
{
    cout<<"Please, enter group number:\n";
    int group_num;
    cin>>group_num;
    return group_num;
}

```

Тестування:

```
Please, enter the quantity of students:
2
Enter PIB
Grygorenko
Group number
22
Enter year:2004
Enter month:05
Enter day:11
success

Enter PIB
Mykolaichuk
Group number
22
Enter year:2003
Enter month:03
Enter day:22
success
```

```
-----
Student 1:
-----
Pib: Grygorenko
Group: 22
Birthday: 11:5:2004

-----
Student 2:
-----
Pib: Mykolaichuk
Group: 22
Birthday: 22:3:2003

*****

Please, enter group number:
22
Oldest student is: Mykolaichuk
Process finished with exit code 0
```