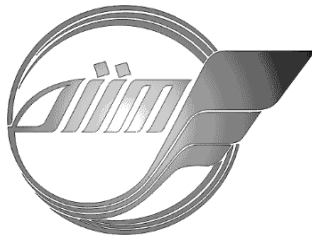


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

**Лабораторна робота №4**  
**з дисципліни «Групова динаміка і комунікації»**

**на тему: «Програмування в парі»**

Виконав:  
студент гр.ПЗ1911  
Сафонов Д.Є.  
Прийняв:  
Литвиненко К.В.

Дніпро, 2020

Тема. Програмування в парі.

Мета. Отримати навички роботи в малій групі (парі) в процесі рішення спеціалізованої задачі й проаналізувати одержаний ефект.

### **Завдання.**

Створіть текстовий файл, що містить інформацію про студентів: П.І.Б., рік народження, ознака отримання стипендії (роздільник між елементами в рядку –кома). Вивести на екран список студентів, які не отримують стипендію.

## Опис формату текстового файлу.

Як найпростіший для обробки був обраний формат \*.txt .

Структура студента у файлі:

- `([A-Za-z]{3,}\s[A-Za-z]{3,}\s[A-Za-z]{3,})\,(19\d{2}|20[01]\d|2020)\,([01])`
- Три групи розділені комами ПІБ, рік народження та статус стипендії(отримує чи ні).
- ПІБ — три послідовності латинських букв не коротше трьох поділені пробілами.
- Рік народження — чотиризначне число від 1900 до 2020
- Статус стипендії 1 або 0.

The screenshot shows a regex testing interface. At the top, the regular expression `/([A-Za-z]{3,}\s[A-Za-z]{3,}\s[A-Za-z]{3,})\,(19\d{2}|20[01]\d|2020)\,([01])/g` is entered. Below it, a text file content is displayed with several lines of student data. The matches for the regex are highlighted in blue. The matches are:

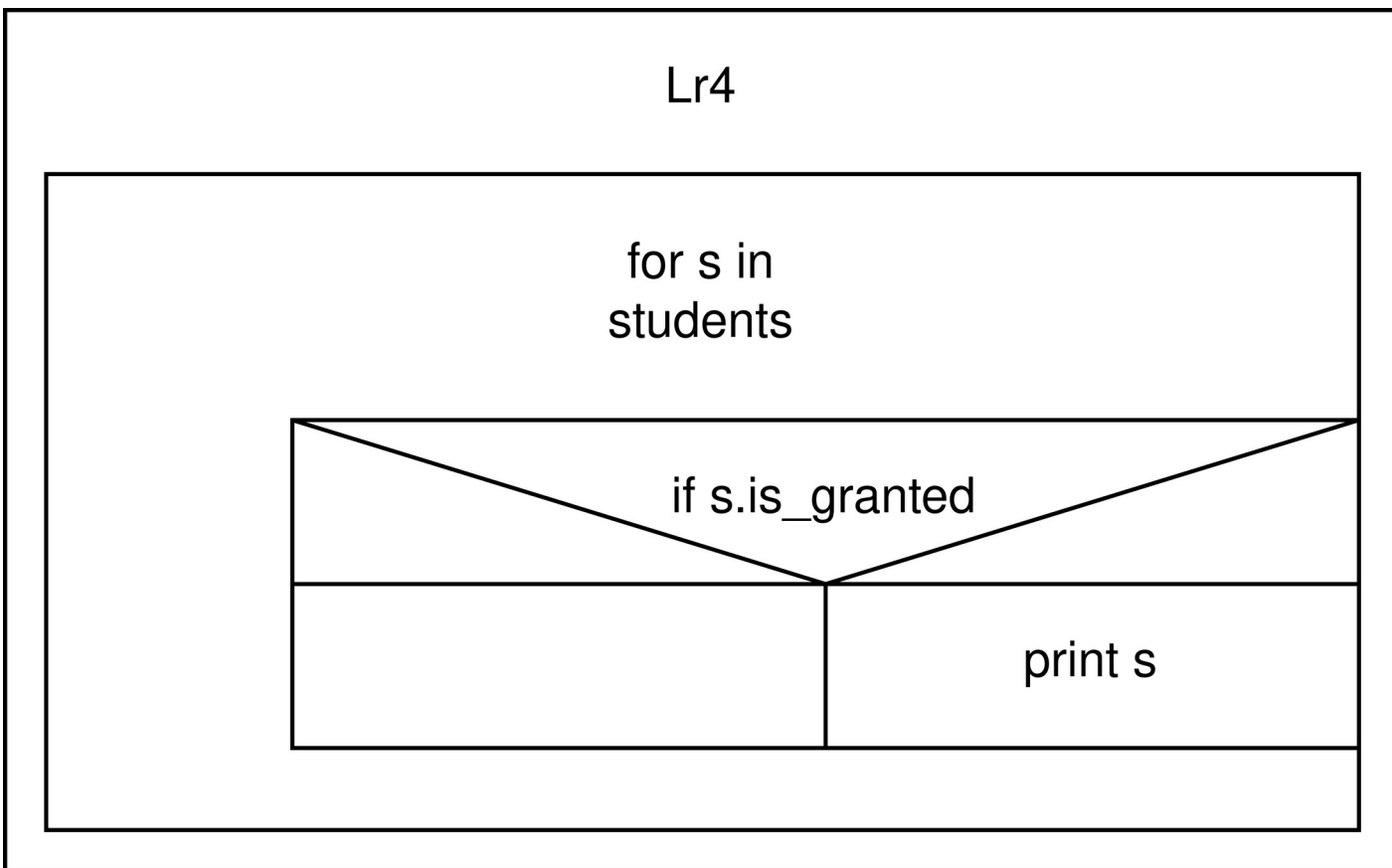
- `tdnehir.knek.nkei,1800,0`
- `lidthrkne.iltkanvsr.nitev,1900,1`
- `tnerhskb.thnrsvbe.tervhnsb,1900,1`
- `tivknehknivt.nkitve.intevks,1900,1`
- `aisvtkuner.alvsnkriue.vluaskir,2100,0`
- `tlniuaskr.sarimnvtetk.tsvarmnei,2020,1`
- `nkhtsrv.tvkhnsr.tvshrnk,2019,0`

At the bottom, the 'Tools' section shows a detailed view of the third match. It indicates that the match is at positions 205-234. The match details are:

Match	Start	End	Match
Match 2	205	234	nkhtsrv tvkhnsr tvshrnk,2019,0
Group 1	205	227	nkhtsrv tvkhnsr tvshrnk
Group 2	229	232	2019
Group 3	234	234	0

Рисунок 1(Приклад роботи регулярного виразу)

## Алгоритм обработки данных.



*Рисунок 2*

## Лістинг програми з коментарями. “main.cpp”

```
#include "menu.h"
int main() {
    Lr4::menu();
}
```

## “lr4.h”

```
#ifndef LR4_H
#define LR4_H

#include <string>
#include <functional>
#include <vector>
#include <iostream>
#include <regex>
#include <fstream>

namespace Lr4 {
    class Option {
    public:
        std::string menu_str;
        std::function<void()> action;
    };

    class Student {
    public:
        Student(const char* name, size_t year, bool is_granted) : year{year}, is_granted{is_granted}
        {memcpy(&(this->name), name, 256);}
        char name[256];//can't read/write std::string to files
        size_t year;
        bool is_granted;
        friend std::ostream& operator<<(std::ostream& os, const Student& s);//to cout
    };

    std::ostream& operator<<(std::ostream& os, const Student& s);
    std::fstream open_file(std::ios_base::openmode mode = std::ios_base::in);
    std::vector<Student> students_str_to_vec(std::string str);//regex matches students from string to vector
(attributes separated by coma)
}
```

```

    std::string students_vec_to_bin(std::vector<Student>& students);//writes students to binary file with
unique name
    std::vector<Student> students_txt_to_vec(std::fstream& file);//reads text file with students into vector
    std::vector<Student> students_bin_to_vec(std::fstream& file);//reads binary file with students into vector

    const std::vector<Option> options {//menu options
        {"Read txt.", [](){
            std::fstream file = open_file();
            std::vector<Student> students = students_txt_to_vec(file);
            std::cout << "\n";
            for(auto& s : students)
                if(s.is_granted) std::cout << s << "\n";
            std::cout << "Students were saved to " << students_vec_to_bin(students) << "\n";
        }},
        {"Read binary.", [](){
            std::fstream file = open_file(std::ios_base::binary | std::ios_base::in);
            std::vector<Student> students = students_bin_to_vec(file);
            for(auto& s : students)
                if(s.is_granted) std::cout << s << "\n";
        }},
    };

    bool try_str_to_option_index(const std::string& str, size_t& result, size_t max_val);//tries to convert
string to appropriate int
    size_t read_choice();//reads menu choice
    void menu();//whole app actually
};

#endif

```

## “lr4.cpp”

```
#include "lr4.h"
bool Lr4::try_str_to_option_index(const std::string& str, size_t& result, size_t max_val) {
    result = atol(str.c_str());
    return result != 0 && result <= max_val && std::to_string(result) == str;
}

size_t Lr4::read_choice() {
    size_t input;
    bool good = true;
    do {
        for(size_t i = 0; i < options.size(); i++)
            std::cout << i + 1 << ". " << options[i].menu_str << "\n";
        std::cout << options.size() + 1 << ". Exit.\n";
        if(!good) std::cout << "Invalid choice! Try again.\n";
        std::string choice;
        std::cin >> choice;
        good = try_str_to_option_index(choice, input, options.size() + 1);
    } while (!good);
    return input - 1;
}

void Lr4::menu() {
    size_t choice;
    while((choice = read_choice()) < options.size()) {
        options[choice].action();
        std::cout << "Press any key to continue...\n";
        std::cin.get();
        system("clear");
    }
}

std::ostream& Lr4::operator<<(std::ostream& os, const Student& s) {
    os << "Name: " << s.name << "\n";
    os << "Year: " << s.year << "\n";
    os << "Grant: " << std::boolalpha << s.is_granted << "\n";
    return os;
}

std::fstream Lr4::open_file(std::ios_base::openmode mode) {
    char filename[256];
    std::fstream file;
    do {
        std::fill_n(filename, 256, 0); //0 = '\0'
        std::cout << "input file address: " << std::endl;
    }
```

```

    std::cin.get();
    std::cin.getline(filename, 255);
    file.open(filename, mode);
    if (!file) std::cout << "Can't open that file!" << std::endl;
} while (!file);
return file;
}

std::vector<Lr4::Student> Lr4::students_str_to_vec(std::string str){
    std::smatch match;
    std::regex expr("([A-Za-z]{3,}\\s[A-Za-z]{3,}\\s[A-Za-z]{3,})\\,(19\\d{2}|20[01]\\d|2020)\\,([01])");
    std::vector<Student> students;
    for (; std::regex_search(str, match, expr); str = match.suffix().str())
        students.push_back(Student(static_cast<std::string>(match[1]).c_str(),
                                   std::stoi(match[2]),
                                   (bool)(static_cast<std::string>(match[3])[0] - 48)));
    return students;
}

std::string Lr4::students_vec_to_bin(std::vector<Student>& students) {
    char filename_out[] = "test_XXXXXX";
    if (mkstemp(filename_out) != -1) {//actually shouldn't happen
        std::ofstream bin(filename_out, std::ios_base::binary | std::ios_base::trunc | std::ios_base::out);
        for (auto& s : students) bin.write(reinterpret_cast<char*>(&s), sizeof(Student));
    }
    return filename_out;
}

std::vector<Lr4::Student> Lr4::students_txt_to_vec(std::fstream& file) {
    std::string str((std::istreambuf_iterator<char>(file)), (std::istreambuf_iterator<char>()));
    return students_str_to_vec(str);
}

std::vector<Lr4::Student> Lr4::students_bin_to_vec(std::fstream& file) {
    file.seekg(0, std::ios_base::end);
    int length = file.tellg() / sizeof(Student);
    file.seekg(0, std::ios_base::beg);
    std::vector<Student> students;
    Student s((char*)"", 0, 0);
    for (int i = 0; i < length; i++) {
        file.read(reinterpret_cast<char*>(&s), sizeof(Student));
        students.push_back(s);
    }
    return students;
}

```



## Результати роботи програми (первинний файл та результат)

Текстовий файл: “strineo sratnie rstaonie,1900,0  
ahrntisenhtesairhrtse rsaethni nrseathi,1800,7”

otsairne rtsnieoa artisone,2020,1

Результат програми при читанні текстового файлу:

```
1. Read txt.  
2. Read binary.  
3. Exit.  
1  
input file adress: /test.txt  
  
Name: otsairne rtsnieoa artisone  
Year: 2020  
Grant: true  
  
Students were saved to test_Sf0cZk  
Press any key to continue...  
█
```

Рисунок 3

Результат програми при читанні бінарного файлу створеного з текстового:

```
1. Read txt.  
2. Read binary.  
3. Exit.  
2  
input file adress: /test3.bin  
  
Name: otsairne rtsnieoa artisone  
Year: 2020  
Grant: true  
  
Press any key to continue...  
█
```

Рисунок 4