

Додаток 1
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»

«Класи та об'єкти»

Варіант 12

Виконав студент: ІП-13 Дойчев Костянтин Миколайович

Перевірила: Вечерковська Анастасія Сергіївна

Київ 2021

Лабораторна робота №2

Тема: Класи та об'єкти

Постановка задачі

Розробити клас “Абонент”, який характеризується ПІБ абонента, його адресою і номером телефону (у форматі XXX-XXX-XX-XX). Створити масив об'єктів даного класу. Визначити абонента, у якого найбільша сума цифр телефону.

Код:

Файл - main.cpp

```
#include <iostream>
#include <vector>
#include "Subscriber.h"
#include "File.h"

int main() {
    std::vector<Subscriber> subscribers;
    try {
        File file("../data.csv");
        std::vector<std::vector<std::string>> data = file.getCsvContent();
        /**
         * @description: starting from the second line of the csv file
         */
        for (int i = 1; i < data.size(); i++) {
            Subscriber subscriber(data[i][0], data[i][1], data[i][2]);
            subscribers.push_back(subscriber);
        }
    } catch (std::exception &e) {
        std::cout << e.what() << std::endl;
        return 1;
    }

    if (subscribers.empty()) {
```

```

        cout << "No subscribers" << endl;
        return 1;
    }

    Subscriber subscriberWithTheBiggestPhoneNumberSum = subscribers[0];
    for (Subscriber subscriber: subscribers) {
        int sum = subscriber.getDigitsSum();
        if (sum > subscriberWithTheBiggestPhoneNumberSum.getDigitsSum()) {
            subscriberWithTheBiggestPhoneNumberSum = subscriber;
        }
    }

    std::cout << subscriberWithTheBiggestPhoneNumberSum.fullName
        << "'s phone number has the biggest sum of digits, which is "
        << subscriberWithTheBiggestPhoneNumberSum.getDigitsSum() <<
std::endl;

    return 0;
}

```

Файл: File.cpp

```
1      #include "File.h"
2
3      File::File(string fileName){
4          this->name = fileName;
5      }
6
7      vector<vector<string>> File::getCsvContent() {
8          vector<vector<string>> content;
9          vector<string> row;
10
11          string line, word;
12
13          fstream file( s: this->name);
14          if (!file.is_open()){
15              throw runtime_error("An error has occurred while changing the file");
16          }
17
18          while(getline( &: file, &: line)){
19              row.clear();
20
21              stringstream ss( s: line);
22              while(getline( &: ss, &: word, dlm: ',')){
23                  row.push_back(word);
24              }
25              content.push_back(row);
26          }
27          return content;
28      }
```

Файл:PhoneNumber.cpp

```
#include "PhoneNumber.h"
#include "StringUtilities.h"

PhoneNumber::PhoneNumber() {
    this->number = "000-00-00-00";
}

PhoneNumber::PhoneNumber(std::string number) {
    this->number = number;
    if (!this->isValid()) {
        throw std::invalid_argument(
            number + " is not a valid phone number. Make sure you match the
            format XXX-XXX-XX-XX");
    }
}
```

```

    }
}

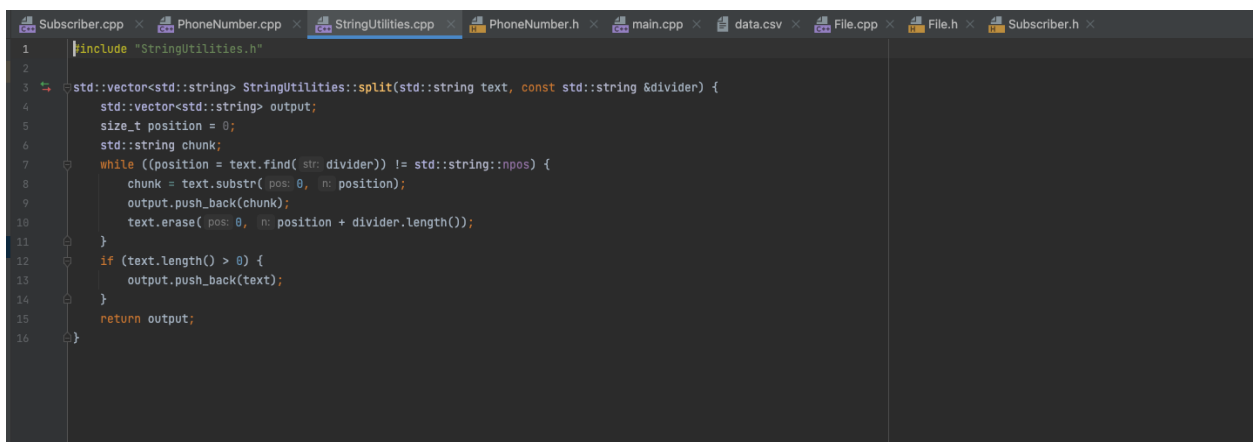
bool PhoneNumber::isValid() {
    if (this->number.length() != 13) {
        return false;
    }
    std::vector<std::string> parts = StringUtilities::split(this->number, "-");
    if (parts.size() != 4) {
        return false;
    }

    /**
     * @description: i < 4 as the we have already checked that length
     */
    for (int i = 0; i < 4; i++) {
        size_t partLength = parts[i].length();
        if (i <= 1 && partLength != 3) {
            return false;
        } else if (i > 1 && partLength != 2) {
            return false;
        }
    }
    return true;
}

std::string PhoneNumber::getNumber() {
    return this->number;
}

```

Файл:StringUtilities.cpp



```

1  #include "StringUtilities.h"
2
3  std::vector<std::string> StringUtilities::split(std::string text, const std::string &divider) {
4      std::vector<std::string> output;
5      size_t position = 0;
6      std::string chunk;
7      while ((position = text.find(divider)) != std::string::npos) {
8          chunk = text.substr(position, text.length() - position);
9          output.push_back(chunk);
10         text.erase(position, position + divider.length());
11     }
12     if (text.length() > 0) {
13         output.push_back(text);
14     }
15     return output;
16 }

```

Файл: Subscriber.cpp

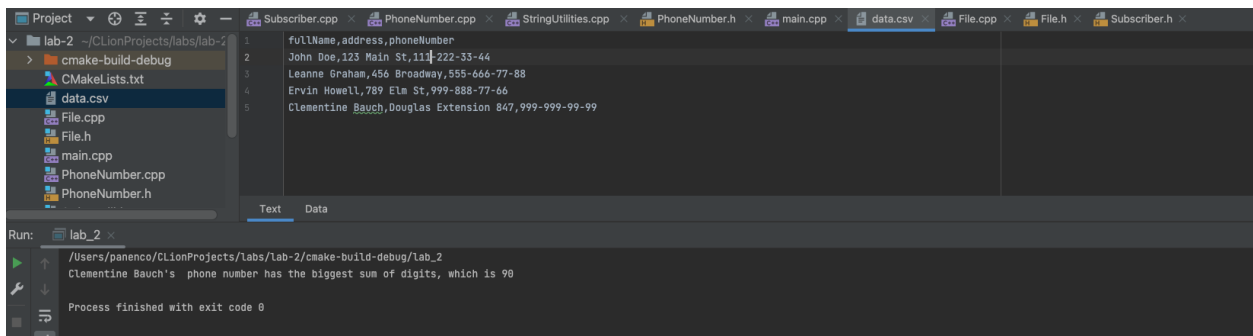
```
#include "Subscriber.h"

Subscriber::Subscriber() {}

Subscriber::Subscriber(std::string fullName, std::string address, std::string phoneNumber) {
    this->fullName = fullName;
    this->address = address;
    try {
        this->phoneNumber = PhoneNumber( number: phoneNumber);
    } catch (const std::invalid_argument& exception) {
        throw std::invalid_argument(s: fullName+ " has invalid phone number. " + exception.what());
    }
}

int Subscriber::getDigitsSum() {
    std::vector<std::string> parts = StringUtilities::split( string: this->phoneNumber.getNumber(), divider: "-");
    int sum = 0;
    for (const std::string& part : parts){
        for (char c : part){
            /**
             * @description c - '0' is to cast char to int. a - '0' is equivalent to ((int)a) - ((int)'0')
             */
            sum += c - '0';
        }
    }
    return sum;
}
```

Дані і консоль:



The screenshot shows a C++ IDE with a project named 'lab-2'. The project files include 'Subscriber.cpp', 'PhoneNumber.cpp', 'StringUtilities.cpp', 'PhoneNumber.h', 'main.cpp', 'data.csv', 'File.cpp', 'File.h', and 'Subscriber.h'. The 'data.csv' file is open, showing a list of subscribers with their full names, addresses, and phone numbers. The console output shows the results of the program, indicating that Clementine Bauch's phone number has the highest sum of digits (90).

fullName	address	phoneNumber
John Doe	123 Main St	111-222-33-44
Leanne Graham	456 Broadway	555-666-77-88
Ervin Howell	789 Elm St	999-888-77-66
Clementine Bauch	Douglas Extension	847,999-999-99-99

Run: lab_2

/Users/panenco/CLionProjects/Labs/Lab-2/cmake-build-debug/lab_2

Clementine Bauch's phone number has the biggest sum of digits, which is 90

Process finished with exit code 0