

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 10

Виконав студент ІП-15 Закірова Олександра Володимирівна
Перевірів Вечерковська Анастасія Сергіївна

Київ 2022

Лабораторна робота 2

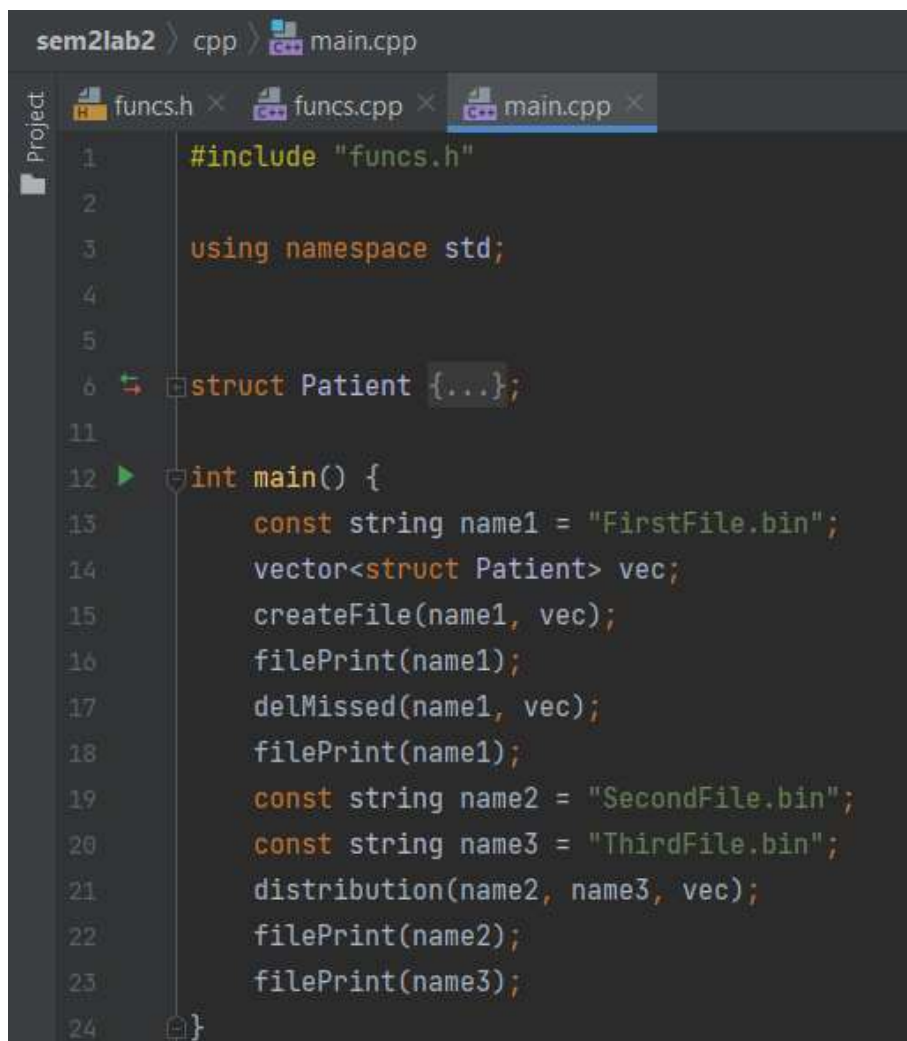
Мета – вивчити особливості створення і обробки бінарних файлів даних.

Індивідуальне завдання


Варіант 10

10. Створити файл із списком пацієнтів, записаних на прийом до лікаря: прізвище пацієнта, дата попереднього відвідування лікаря та час, на який пацієнт записаний. Видалити з файлу записи про пацієнтів, час прийому яких минув. Створити два нових файли: в один занести відомості про вторинних хворих (попереднє відвідування яких було протягом 10-ти останніх днів), а в другий – про решту пацієнтів.

Виконання на C++



```
sem2lab2 > cpp > main.cpp
Project
funcs.h x  funcs.cpp x  main.cpp x
1      #include "funcs.h"
2
3      using namespace std;
4
5
6      struct Patient {...};
11
12  int main() {
13      const string name1 = "FirstFile.bin";
14      vector<struct Patient> vec;
15      createFile(name1, vec);
16      filePrint(name1);
17      delMissed(name1, vec);
18      filePrint(name1);
19      const string name2 = "SecondFile.bin";
20      const string name3 = "ThirdFile.bin";
21      distribution(name2, name3, vec);
22      filePrint(name2);
23      filePrint(name3);
24  }
```

 File Edit View Navigate Code Refactor Build Run Tools VCS Window Help sem2lab2 - funcs.h
sem2lab2 > cpp > funcs.h
Project
funcs.h x funcs.cpp x main.cpp x
1 #include <iostream>
2 #include <iomanip>
3 #include <fstream>
4 #include <ctime>
5 #include <time.h>
6 #include <vector>
7 #include <string>
8
9 using namespace std;
10
11
12 struct Patient;
13 bool isNum(string text);
14 string getSurname();
15 time_t getDate();
16 time_t getTime();
17 Patient getData();
18 bool isEx(Patient person);
19 bool isMissed(Patient person);
20 void createFile(const string &name, vector <Patient> &vec);
21 void delMissed(const string &name, vector <Patient> &vec);
22 void distribution(const string &name1, const string &name2, vector <Patient> &vec);
23 void filePrint(const string &name);
24

Project

funcs.h x funcs.cpp x main.cpp x

```
1  #include "funcs.h"
2
3  using namespace std;
4
5
6  struct Patient {
7      char surname[20];
8      time_t date;
9      time_t time;
10 };
11
12 bool isNum(string text) {
13     for (char i: text) {
14         if (isdigit(i)) {
15             return true;
16         }
17     }
18     return false;
19 }
20
21 string getSurname() {
22     string surname;
23     cout << "Enter the surname of patient\n";
24     cin >> surname;
25     while (isNum(surname) || surname.size() > 20) {
26         cout << "Please, do not enter any numbers in surname. Also it should be less than 20 symbols.\nTry again: ";
27         cin >> surname;
28     }
29     return surname;
30 }
```

Structure

```
32 time_t getDate() {
33     tm tmRef = {0, 0, 0, 0, 0, 0};
34     cout << "Enter the date of previous appointment\n(USE FORMAT 18.02.2011)\n ";
35     cin >> get_time(&tmRef, "%d.%m.%Y");
36     while (cin.fail()) {
37         cin.clear();
38         cin.ignore(numeric_limits<streamsize>::max(), '\n');
39         cout << "Failed:\nTry again: ";
40         cin >> get_time(&tmRef, "%d.%m.%Y");
41     }
42     return mktime(&tmRef);
43 }
44
45 time_t getTime() {
46     time_t t = time(0);
47     tm tmRef = {0, 0, 0, localtime(&t)->tm_mday, localtime(&t)->tm_mon, localtime(&t)->tm_year};
48     cout << "Enter the time of appointment\n(USE FORMAT 17:30)\n";
49     cin >> get_time(&tmRef, "%H:%M");
50     while (cin.fail()) {
51         cin.clear();
52         cin.ignore(numeric_limits<streamsize>::max(), '\n');
53         cout << "Failed:\nTry again: ";
54         cin >> get_time(&tmRef, "%H:%M");
55     }
56     return mktime(&tmRef);
57 }
```

```
Project
funcs.h x
funcs.cpp x
main.cpp x

59 Patient getData() {
60     Patient person;
61     strcpy(person.surname, getSurname().c_str());
62     person.date = getDate();
63     person.time = getTime();
64     return person;
65 }
66
67 bool isEx(Patient person) {
68     time_t t1 = person.date;
69     time_t t2 = time(NULL);
70     double a = difftime(t2, t1);
71     if (a > 864000) {
72         return false;
73     }
74     return true;
75 }
76
77 bool isMissed(Patient person) {
78     time_t t1 = person.time;
79     time_t t2 = time(NULL);
80     double a = difftime(t2, t1);
81     if (a < 0) {
82         return false;
83     }
84     return true;
85 }
```

```
87 void createFile(const string &name, vector<Patient> &vec) {
88     ofstream file;
89     string answer;
90     file.open(name, ios::binary);
91     while (file.is_open()) {
92         cout << "Would you like to add a patient? Please, enter 'yes' or 'no':\n";
93         cin >> answer;
94         while (answer != "yes" && answer != "no") {
95             cout << "PLEASE, enter 'yes' or 'no':\n";
96             cin >> answer;
97         }
98         if (answer == "yes") {
99             Patient human = getData();
100             vec.push_back(human);
101             file.write((char *) &human, sizeof(Patient));
102         } else file.close();
103     }
104     if (vec.empty()) {
105         cout << "\nI can't complete the task because of you.";
106         exit(0);
107     }
108 }
```



```
110 void delMissed(const string &name, vector<Patient> &vec) {
111     cout << "DELETING records about those patients whose time has already passed...\n";
112     ofstream file;
113     file.open(name, ios::binary);
114     if (!file.is_open()) {
115         cout << "Error while opening file => i can't overwrite the file :( ";
116         exit(EXIT_FAILURE);
117     } else {
118         for (vector<Patient>::iterator i = vec.begin(); i != vec.end(); ++i) {
119             if (!isMissed(i.operator*())) {
120                 file.write((char *) &i.operator*(), sizeof(Patient));
121             }
122         }
123     }
124     file.close();
125     vector<struct Patient> newVec;
126     newVec.reserve(vec.size());
127     for (auto i = 0; i < vec.size(); ++i) {
128         if (!isMissed(vec[i])) {
129             newVec.emplace_back(std::move(vec[i]));
130         }
131     }
132     vec.swap(newVec);
133     cout << "Done!\n ";
134 }
```



```
136 void distribution(const string &name1, const string &name2, vector<Patient> &vec) {
137     if (vec.empty()) {
138         cout << "\nNo more patients. I can't complete the task because of you.";
139         exit(0);
140     }
141     cout << "I will distribute patients into two new files...\n";
142     ofstream f1;
143     f1.open(name1, ios::binary);
144     if (!f1.is_open()) {
145         cout << "Error opening second file :(";
146         exit(EXIT_FAILURE);
147     } else {
148         for (vector<Patient>::iterator i = vec.begin(); i != vec.end(); ++i) {
149             if (isEx(i.operator*())) {
150                 f1.write((char *) &i.operator*(), sizeof(Patient));
151             }
152         }
153         f1.close();
154     }
155     ofstream f2;
156     f2.open(name2, ios::binary);
157     if (!f2.is_open()) {
158         cout << "Error opening third file :(";
159         exit(EXIT_FAILURE);
160     }
161     for (vector<Patient>::iterator i = vec.begin(); i != vec.end(); ++i) {
162         if (!isEx(i.operator*())) {
163             f2.write((char *) &i.operator*(), sizeof(Patient));
164         }
165     }
166     f2.close();
167     cout << "Done!";
168 }
```

```

170 void filePrint(const string &name) {
171     cout << "\n" << setw(20) << name << " contains:\n";
172     Patient person{};
173     ifstream file(name, ios::binary);
174     while (file.read((char *) &person, sizeof(Patient))) {
175         string surname(person.surname);
176
177         time_t dt = person.date;
178         struct tm *dateinfo;
179         time(&dt);
180         dateinfo = localtime(&dt);
181         string date = asctime(dateinfo);
182
183         time_t tm = person.time;
184         struct tm *timeinfo;
185         time(&tm);
186         timeinfo = localtime(&tm);
187         string time = asctime(timeinfo);
188
189         cout << "Surname: " << surname << "\n" << "Last appointment date: " << date << "Time of sppappointment: " << time
190             << "\n";
191     }
192     file.close();
193 }

```



ENG



21:59

23.04.2022

```
Run: sem2lab2 x
C:\Users\Legion\CLionProjects\sem2lab2\cmake-build-debug\sem2lab2.exe
Would you like to add a patient? Please, enter 'yes' or 'no':
yes
Enter the surname of patient
Matsko
Enter the date of previous appointment
(USE FORMAT 18.02.2011)
15.04.2022
Enter the time of appointment
(USE FORMAT 17:30)
23:20
Would you like to add a patient? Please, enter 'yes' or 'no':
yes
Enter the surname of patient
Kizyuk
Enter the date of previous appointment
(USE FORMAT 18.02.2011)
12.07.2021
Enter the time of appointment
(USE FORMAT 17:30)
18:00
Would you like to add a patient? Please, enter 'yes' or 'no':
yes
Enter the surname of patient
Petrushevskaya
Enter the date of previous appointment
(USE FORMAT 18.02.2011)
17.09.2019
Enter the time of appointment
(USE FORMAT 17:30)
22:25
Would you like to add a patient? Please, enter 'yes' or 'no':
no
```

```
Run: sem2lab2 x

FirstFile.bin contains:
Surname: Matsko
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

Surname: Kizyuk
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

Surname: Petrushevskaya
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022
```

```
DELETING records about those patients whose time has already passed...
Done!

FirstFile.bin contains:
Surname: Matsko
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

Surname: Petrushevskaya
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

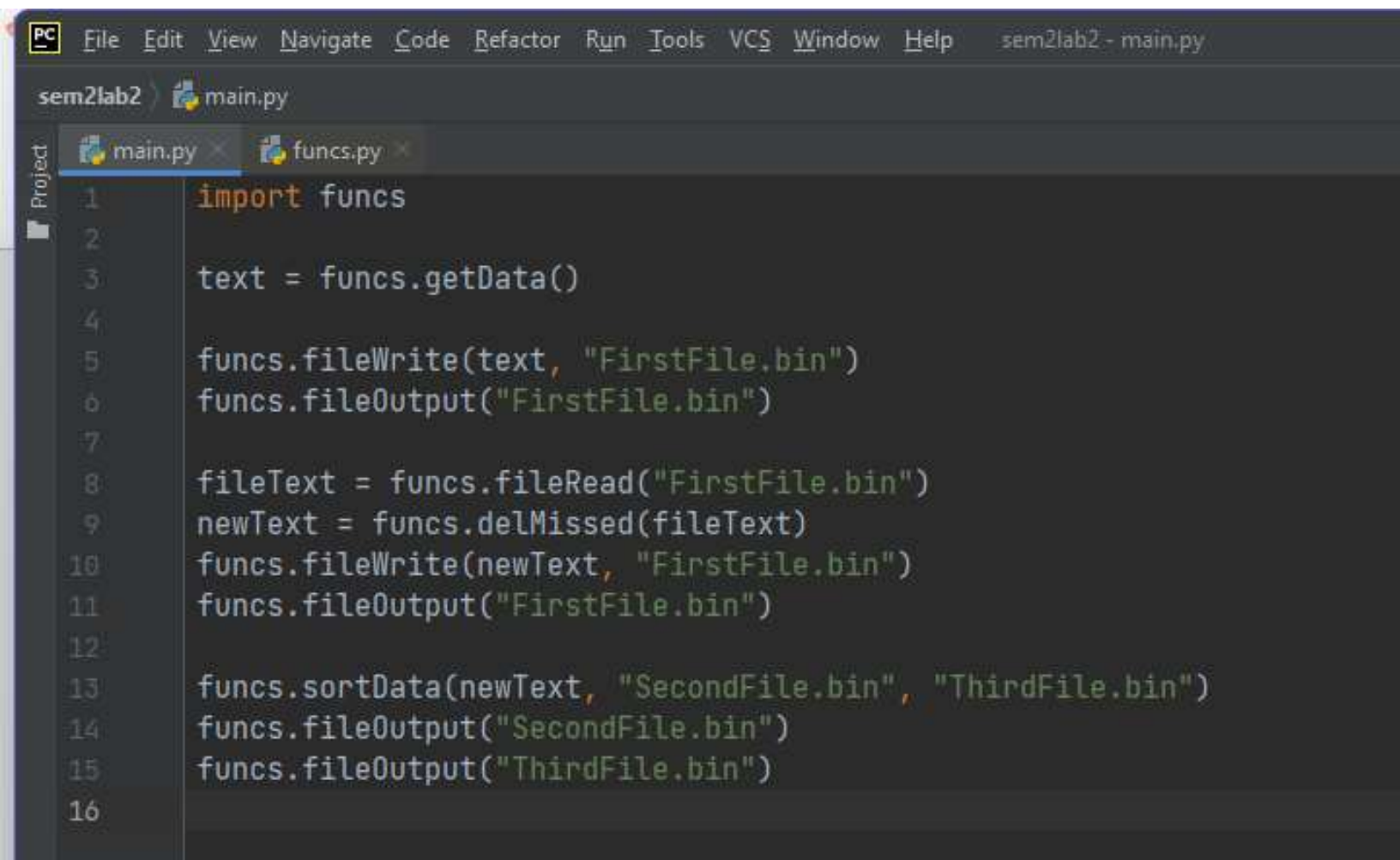
I will distribute patients into two new files...
Done!

SecondFile.bin contains:
Surname: Matsko
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

ThirdFile.bin contains:
Surname: Petrushevskaya
Last appointment date: Sat Apr 23 21:57:05 2022
Time of sppaintment: Sat Apr 23 21:57:05 2022

Process finished with exit code 0
```

Виконання на Python



The image shows a screenshot of an IDE window titled "sem2lab2 - main.py". The interface includes a menu bar with options: File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. Below the menu bar, the project name "sem2lab2" is displayed. The editor area shows two open files: "main.py" and "funcs.py". The "main.py" file is active, displaying the following Python code:

```
1 import funcs
2
3 text = funcs.getData()
4
5 funcs.fileWrite(text, "FirstFile.bin")
6 funcs.fileOutput("FirstFile.bin")
7
8 fileText = funcs.fileRead("FirstFile.bin")
9 newText = funcs.delMissed(fileText)
10 funcs.fileWrite(newText, "FirstFile.bin")
11 funcs.fileOutput("FirstFile.bin")
12
13 funcs.sortData(newText, "SecondFile.bin", "ThirdFile.bin")
14 funcs.fileOutput("SecondFile.bin")
15 funcs.fileOutput("ThirdFile.bin")
16
```



```

1  import re
2  import time
3  from datetime import datetime, date
4
5
6  def getData():
7      text = []
8      flag = 'yes'
9      while flag == 'yes':
10         surname = input("Enter a surname of the client: ")
11         surname = checkSurname(surname)
12
13         strtime = input("Enter the time of appointment(USE FORMAT 19:30):")
14         t = datetime.strptime(strtime, "%H:%M")
15         tm = datetime.time(t)
16         fulltime = datetime.combine(date.today(), tm)
17
18         strdate = input("Enter the date of previous appointment(USE FORMAT 22.07.2019):")
19         fulldate = datetime.strptime(strdate, '%d.%m.%Y')
20
21         line = surname + ',' + str(fulltime) + ',' + str(fulldate) + ';\n'
22         text.append(line)
23
24         flag = input("\nIf you want to add a patient print 'yes':")
25     return text
26
27
28 def checkSurname(text):
29     digits = re.findall('[0-9]', text)
30     while len(text) > 20 or len(text) <= 1 or len(digits) > 0:
31         text = input("Enter surname again: ")
32         digits = re.findall('[0-9]', text)
33     return text
34
35
36 def fileWrite(text, name):
37     file = open(name, "wb")
38     for i in range(len(text)):
39         line = text[i]
40         file.write(line.encode())
41     file.close()

```

```

44 def fileRead(name):
45     text = []
46     file = open(name, "rb")
47     while True:
48         line = file.readline().decode()
49         if len(line) < 1:
50             break
51         text.append(line)
52     file.close()
53     return text
54
55
56 def delMissed(text):
57     now = datetime.now().timetuple()
58     for line in text:
59         patient = line.split(',')
60         buf = datetime.strptime(patient[1], "%Y-%m-%d %H:%M:%S").timetuple()
61         tm1 = time.mktime(buf)
62         tm2 = time.mktime(now)
63         time_diff = tm2 - tm1
64         if time_diff > 0:
65             text.remove(line)
66     return ''.join(text)
67

```



```

67
68
69 def sortData(text, name1, name2):
70     data = date.today().timetuple()
71     text1 = []
72     text2 = []
73     lines = text.split(';\\n')
74     lines = lines[:-1]
75     for line in lines:
76         patient = line.split(',')
77         buf = datetime.strptime(patient[2], "%Y-%m-%d %H:%M:%S").timetuple()
78         dt1 = time.mktime(buf)
79         dt2 = time.mktime(data)
80         date_diff = dt2 - dt1
81         if date_diff <= 864000.0:
82             text1.append(line + '\\n')
83         else:
84             text2.append(line + '\\n')
85     fileWrite(text1, name1)
86     fileWrite(text2, name2)
87
88
89 def fileOutput(name):
90     file = open(name, "rb")
91     print('\\n' + name + ' contains:\\n')
92     while True:
93         line = file.readline().decode()
94         if len(line) < 1:
95             break
96         if len(line) > 2:
97             print(line, end='')
98     file.close()
99

```



ENG



18:52

23.04.2022

Run: main x

C:\Users\Legion\PycharmProjects\sem2lab2\venv\Scripts\python.exe C:/Users/L

Enter a surname of the client: *Matsko*

Enter the time of appointment(USE FORMAT 19:30):*20:00*

Enter the date of previous appointment(USE FORMAT 22.07.2019):*15.04.2022*

If you want to add a patient print 'yes':*yes*

Enter a surname of the client: *Kizyuk*

Enter the time of appointment(USE FORMAT 19:30):*18:00*

Enter the date of previous appointment(USE FORMAT 22.07.2019):*12.07.2021*

If you want to add a patient print 'yes':*yes*

Enter a surname of the client: *Petrushevskaya*

Enter the time of appointment(USE FORMAT 19:30):*20:35*

Enter the date of previous appointment(USE FORMAT 22.07.2019):*17.09.2019*

If you want to add a patient print 'yes':*no*

FirstFile.bin contains:

Matsko,2022-04-23 20:00:00,2022-04-15 00:00:00;

Kizyuk,2022-04-23 18:00:00,2021-07-12 00:00:00;

Petrushevskaya,2022-04-23 20:35:00,2019-09-17 00:00:00;

FirstFile.bin contains:

Matsko,2022-04-23 20:00:00,2022-04-15 00:00:00;

Petrushevskaya,2022-04-23 20:35:00,2019-09-17 00:00:00;

SecondFile.bin contains:

Matsko,2022-04-23 20:00:00,2022-04-15 00:00:00;

ThirdFile.bin contains:

Petrushevskaya,2022-04-23 20:35:00,2019-09-17 00:00:00;

Process finished with exit code 0

Висновки

Протягом лабораторної роботи було розглянуто роботу з бінарними файлами та використано отримані навички під час написання програм. В результаті роботи було створену програму, яка створює новий файл, що містить інформацію про пацієнтів, перевіряє на коректність введені дані, видаляє дані з файлу, якщо час прийому пацієнта вже минув, а також створює 2 нових файли — у першому записи про пацієнтів, попереднє відвідування яких було протягом 10-ти останніх днів, у другому — про решту пацієнтів.