

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»

«Бінарні файли»

Варіант №4

Виконав студент ІП-14 Берковський Максим Юрійович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №2

Тема: Бінарні файли

Мета – вивчити особливості створення і обробки бінарних файлів.

Хід роботи

Задача.

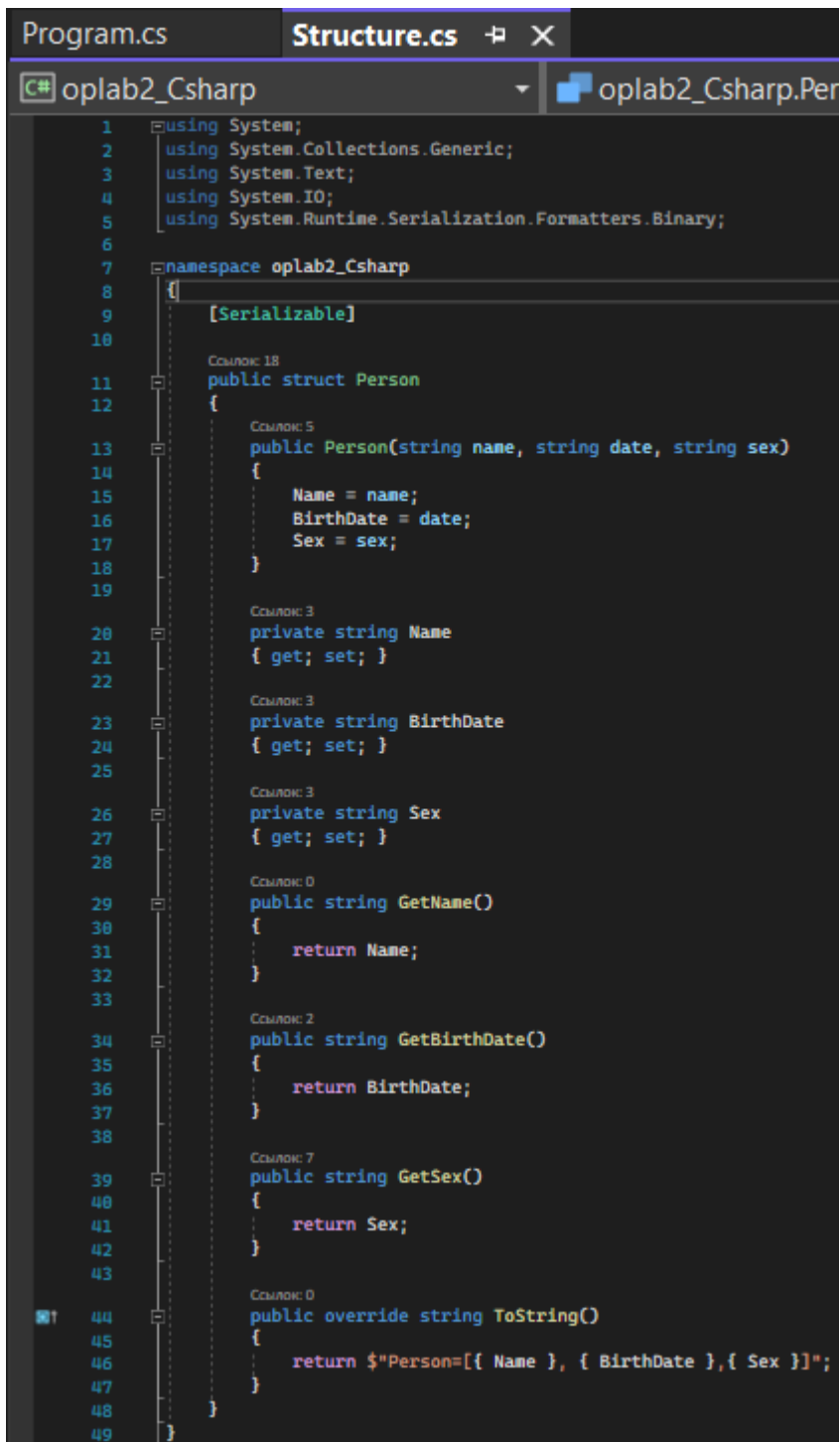
4. Створити файл з інформацією про абітурієнтів: його ПІБ, дата народження (ДД.ММ.РРРР), стать (чоловік / жінка). Видалити зі списку абітурієнтів старше 35 років. Створити новий файл, до якого занести інформацію про юнаків призовного віку (від 18 до 27 років).

Постановка задачі.

Для виконання цього завдання нам потрібно створити структуру, яка буде вміщати в собі інформацію про абітурієнта. Він буде вміщати в собі ім'я, дату народження та стать. Для занесення інформації в файл та взяття її з файлу ми будемо використовувати серіалізацію. Для розрахунку віку абітурієнта на мові C# знадобиться підставляти дату народження в різні проміжки часу. На мові python з розрахунком віку абітурієнтів нам допоможе модуль datetime призначений для роботи з датами та часом.

Випробування коду на C#:

Код:



```
Program.cs | Structure.cs | oplab2_Csharp | oplab2_Csharp.Per
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using System.IO;
5  using System.Runtime.Serialization.Formatters.Binary;
6
7  namespace oplab2_Csharp
8  {
9      [Serializable]
10
11      public struct Person
12      {
13          public Person(string name, string date, string sex)
14          {
15              Name = name;
16              BirthDate = date;
17              Sex = sex;
18          }
19
20          private string Name
21          { get; set; }
22
23          private string BirthDate
24          { get; set; }
25
26          private string Sex
27          { get; set; }
28
29          public string GetName()
30          {
31              return Name;
32          }
33
34          public string GetBirthDate()
35          {
36              return BirthDate;
37          }
38
39          public string GetSex()
40          {
41              return Sex;
42          }
43
44          public override string ToString()
45          {
46              return $"Person=[{ Name }, { BirthDate },{ Sex }]";
47          }
48      }
49  }
```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4  using System.IO;
5  using System.Runtime.Serialization.Formatters.Binary;
6
7  namespace oplab2_Csharp
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             //створення назв файлів
14             const string inputPath = "input.txt";
15             const string del35Path = "del35.txt";
16             const string VoenskomPath = "voenskom.txt";
17             //створення файла зі списком абітурієнтів
18             InitEntities(inputPath);
19             var people = GetFromFile(inputPath);
20             OutputPeople(people, "Here are people:");
21             //видалення зі списку абітурієнтів старших 35 років та занесення до нового файлу
22             var del35people = Del35(people);
23             OutputPeople(del35people, "Here are people under 35:");
24
25             WriteToFile(del35people, del35Path, FileMode.OpenOrCreate);
26
27             var del35people_fromfile = GetFromFile(del35Path);
28             //створення файлу зі списком абітурієнтів призовного віку
29             var voenskom = Voenskom(del35people_fromfile, VoenskomPath);
30             OutputPeople(voenskom, "Here are people who should go to voenskom:");
31             WriteToFile(voenskom, VoenskomPath, FileMode.OpenOrCreate);
32
33             //перевірка файлів
34             Console.WriteLine("Checking files...\n");
35             var del35newlist = GetFromFile(del35Path);
36             OutputPeople(del35newlist, "From file del35:");
37
38             var voenskomnewlist = GetFromFile(VoenskomPath);
39             OutputPeople(voenskomnewlist, "From file voenskom:");
40         }
41     }

```

```
42     private static void OutputPeople(List<Person> people, string prompt)
43     {
44         Console.WriteLine(prompt);
45
46         foreach (var person in people)
47         {
48             Console.WriteLine(person);
49         }
50         Console.WriteLine();
51     }
52
53     private static void InitEntities(string path)
54     {
55         var people = new List<Person>()
56         {
57             new Person( "Ivan36", "11.04.1986", "male"),
58             new Person( "Ivan34-35", "04.04.1987", "male"),
59             new Person( "Lucy22", "12.07.2000", "female"),
60             new Person( "Nik18+", "04.04.2004", "male"),
61             new Person( "Max17", "17.05.2004", "male")
62         };
63
64         WriteToFile(people, path, FileMode.OpenOrCreate);
65     }
66
67     private static List<Person> Voenkom(List<Person> old, string path)
68     {
69         var people= new List<Person>();
70
71         foreach (var person in old)
72         {
73             var date = person.GetBirthDate().Split('.');
74             int year = Convert.ToInt32(date[2]);
75             int month;
76             int day;
```

```
76     int day;
77
78     if(date[1].StartsWith("0"))
79     {
80         month = Convert.ToInt32(date[1].Substring(1));
81     }
82     else
83     {
84         month = Convert.ToInt32(date[1]);
85     }
86
87     if (date[0].StartsWith("0"))
88     {
89         day = Convert.ToInt32(date[0].Substring(1));
90     }
91     else
92     {
93         day = Convert.ToInt32(date[0]);
94     }
95
96     if (year > 1995 && year < 2004 && string.Equals(person.GetSex(), "male"))
97     {
98         people.Add(person);
99     }
100
101     if (month > 4 && year == 1995 && string.Equals(person.GetSex(), "male"))
102     {
103         people.Add(person);
104     }
105     if (day > 4 && month == 4 && year == 1995 && string.Equals(person.GetSex(), "male"))
106     {
107         people.Add(person);
108     }
109
110     if (month < 4 && year == 2004 && string.Equals(person.GetSex(), "male"))
111     {
112         people.Add(person);
113     }
114     if (day <= 4 && month == 4 && year == 2004 && string.Equals(person.GetSex(), "male"))
115     {
116         people.Add(person);
117     }
118 }
119
120 return people;
121 }
```

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

Ссылка: 1

```
private static List<Person> Del35(List<Person> old)
{
```

```
    var people = new List<Person>();
```

```
    foreach (var person in old)
```

```
    {
```

```
        var date = person.GetBirthDate().Split('.');
```

```
        int year = Convert.ToInt32(date[2]);
```

```
        int month;
```

```
        int day;
```

```
        if (date[1].StartsWith("0"))
```

```
        {
```

```
            month = Convert.ToInt32(date[1].Substring(1));
```

```
        }
```

```
        else
```

```
        {
```

```
            month = Convert.ToInt32(date[1]);
```

```
        }
```

```
        if (date[0].StartsWith("0"))
```

```
        {
```

```
            day = Convert.ToInt32(date[0].Substring(1));
```

```
        }
```

```
        else
```

```
        {
```

```
            day = Convert.ToInt32(date[0]);
```

```
        }
```

```
        if (year > 1987)
```

```
        {
```

```
            people.Add(person);
```

```
        }
```

```
        if (year == 1987 && month > 4 && string.Equals(person.GetSex(), "male"))
```

```
        {
```

```
            people.Add(person);
```

```
        }
```

```
        if (year == 1987 && day > 4 && month == 4 && string.Equals(person.GetSex(), "male"))
```

```
        {
```

```
            people.Add(person);
```

```
        }
```

```

167     }
168     return people;
169 }
170
171
172 Ссылка: 3
173 private static void WriteToFile(List<Person> people, string outputPath, FileMode fileMode)
174 {
175     var stream = File.Open(outputPath, FileMode.Create);
176
177     var binaryFormatter = new BinaryFormatter();
178
179     foreach (var person in people)
180     {
181         binaryFormatter.Serialize(stream, person);
182     }
183
184     stream.Close();
185 }
186
187 Ссылка: 4
188 private static List<Person> GetFromFile(string outputPath)
189 {
190     var people = new List<Person>();
191     var stream = File.Open(outputPath, FileMode.Open);
192     var binaryFormatter = new BinaryFormatter();
193
194     while (stream.Position < stream.Length)
195     {
196         people.Add((Person)binaryFormatter.Deserialize(stream));
197     }
198
199     stream.Close();
200
201     return people;
202 }

```

Результат:

```

Here are people:
Person=[Ivan36, 11.04.1986,male]
Person=[Ivan34-35, 04.04.1987,male]
Person=[Lucy22, 12.07.2000,female]
Person=[Nik18+, 04.04.2004,male]
Person=[Max17, 17.05.2004,male]

Here are people under 35:
Person=[Lucy22, 12.07.2000,female]
Person=[Nik18+, 04.04.2004,male]
Person=[Max17, 17.05.2004,male]

Here are people who should go to voenkom:
Person=[Nik18+, 04.04.2004,male]

Checking files...

From file del35:
Person=[Lucy22, 12.07.2000,female]
Person=[Nik18+, 04.04.2004,male]
Person=[Max17, 17.05.2004,male]

From file voenkom:
Person=[Nik18+, 04.04.2004,male]

```


Випробування коду на Python:

Код:

```
main.py × modules.py × Person.py ×
1  from modules import *
2  #створення назв файлів
3  input_path = "input.txt"
4  voenkom_path = "voenkom.txt"
5  del35_path = "del35.txt"
6  # створення файла зі списком абітурієнтів
7  InitEntities(input_path)
8  people = GetFromFile(input_path)
9  OutputPeople(people, "Here are people from input file:")
10 #видалення зі списку абітурієнтів старших 35 років та занесення до нового файлу
11 peopleunder35 = Del35(people)
12 OutputPeople(peopleunder35, "Here are people under 35:")
13 WritePeopleToFile(peopleunder35, del35_path, "wb")
14 #створення файлу зі списком абітурієнтів призовного віку
15 peoplevoenkom = Voenkom(peopleunder35)
16 OutputPeople(peoplevoenkom, "Here are people who needs to go to voenkom:")
17 WritePeopleToFile(peoplevoenkom, voenkom_path, "wb")
18 #перевірка
19 print("Checking files...\n")
20 peopleunder35new = GetFromFile(del35_path)
21 OutputPeople(peopleunder35new, "From file del35:")
22 peoplevoenkomnew = GetFromFile(voenkom_path)
23 OutputPeople(peoplevoenkomnew, "From file voenkom:")
```

```
main.py × modules.py × Person.py ×
1  import pickle
2      from datetime import date
3  from Person import Person
4
5  def OutputPeople(people: list, prompt: str):
6      print(prompt)
7
8      for person in people:
9          print("Person:", person.name, person.birthdate, person.sex)
10
11     print()
12
13  def WritePeopleToFile(people: list, path: str, mode: str):
14      with open(path, mode) as file:
15          for person in people:
16              pickle.dump(person, file)
17
18  def GetFromFile(path: str):
19      people = []
```

```
20
21     with open(path, "rb") as file:
22         while True:
23             try:
24                 people.append(pickle.load(file))
25             except EOFError:
26                 break
27
28     return people
29
30 def InitEntities(path: str):
31     people = [
32         Person("Ivan36", "1986-04-11", "male"),
33         Person("Ivan34-35", "1987-04-05", "male"),
34         Person("Nik", "2004-04-04", "male"),
35         Person("Max17", "2004-05-17", "male"),
36         Person("Den34", "1988-04-11", "male"),
37         Person("Lucy22", "2000-04-11", "female")
38     ]
39     WritePeopleToFile(people, path, "wb")
40
```

```

40
41 def Del35(people: list):
42     newpeople = []
43     for person in people:
44         personbirthdate_list = person.birthdate.split('-')
45         personbirthdate = date(int(personbirthdate_list[0]), int(personbirthdate_list[1]), int(personbirthdate_list[2]))
46         age = (date.today() - personbirthdate) / 365.242199
47         age = age.days
48         if age < 35:
49             newpeople.append(person)
50
51     return newpeople
52
53 def Voenkom(people: list):
54     newpeople = []
55     for person in people:
56         personbirthdate_list = person.birthdate.split('-')
57         personbirthdate = date(int(personbirthdate_list[0]), int(personbirthdate_list[1]), int(personbirthdate_list[2]))
58         age_temp = (date.today() - personbirthdate)
59         age_temp = age_temp / 365.2
60         age = age_temp.days
61         if age > 17 and age < 28 and person.GetSex() == "male":
62             newpeople.append(person)
63     return newpeople
64

```

```

main.py x modules.py x Person.py x
1 class Person:
2
3     def __init__(self, name, birthdate, sex):
4         self.name = name
5         self.birthdate = birthdate
6         self.sex = sex
7
8     def GetName(self):
9         return self.name
10
11    def GetBirthDate(self):
12        return self.birthdate
13
14    def GetSex(self):
15        return self.sex

```

Результат:

Here are people from input file:

Person: Ivan36 1986-04-11 male

Person: Ivan34-35 1987-04-05 male

Person: Nik 2004-04-04 male

Person: Max17 2004-05-17 male

Person: Den34 1988-04-11 male

Person: Lucy22 2000-04-11 female

Here are people under 35:

Person: Ivan34-35 1987-04-05 male

Person: Nik 2004-04-04 male

Person: Max17 2004-05-17 male

Person: Den34 1988-04-11 male

Person: Lucy22 2000-04-11 female

Here are people who needs to go to voenkom:

Person: Nik 2004-04-04 male

Checking files...

From file del35:

Person: Ivan34-35 1987-04-05 male

Person: Nik 2004-04-04 male

Person: Max17 2004-05-17 male

Person: Den34 1988-04-11 male

Person: Lucy22 2000-04-11 female

From file voenkom:

Person: Nik 2004-04-04 male

Висновок:

- ознайомився з бінарними файлами та навчився розробляти алгоритми і програми з їх застосуванням;
- перевінив роботу програми, випробувавши її у відладчику середовища розробки Visual Studio та PyCharm, і отримав задовільний результат.