

Sofia University
Department of Mathematics and Informatics

Course : OO Programming C#.NET

Date: January 9, 2019

Student Name:

Lab No. 12

Submit the all C# .NET files developed to solve the problems listed below. Use comments and Modified-Hungarian notation.

Problem No.1

Open the sample code for Lecture 12 (*Lab12P1SampleCode.rar*) and make project

SortedListSample the **Startup project**. Complete the following tasks:

- Modify the Constructor for `_SortedList<string, string> openWith` to create a `SortedList`, where the keys will be sorted in descending order. Add a group of commands to make sure the Keys "txt", ",", "rtf" are stored in this order.
- Sort the Values of `openWith` `_SortedList<string, string> openWith` in descending order. Add a group of commands to make sure the Values "notepad.exe", ",", "wordpad.exe" are stored in this order. Make the sorting work using method `Sort()` of class `List` and also using method `OrderBy()` of class `List`

Problem No.2

Open the sample code for lab 12 (*Lab12SampleCode.rar*) and make project `DictionaryDemo` the

Startup project. Complete the following tasks:

- Replace `Dictionary<Person, BankAccount>` with `SortedDictionary<Person, BankAccount>`. This causes a minor problem. Identify the problem, and fix it
- Създайте

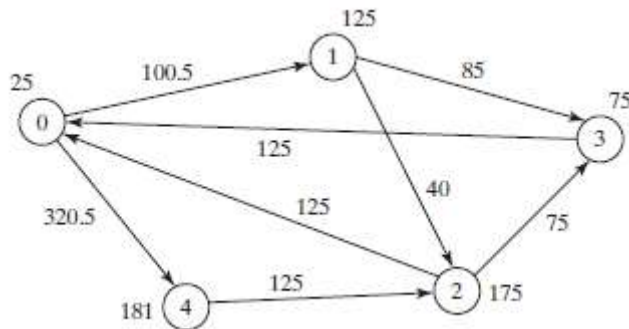
```
public class SalaryEmployee:Person{
    private double salary;
    public SalaryEmployee(string name, double salary):base(name)
    {
        Salary = salary;
    }
    public double Salary
    {
        get { return salary; }
        set { salary = value; }
    }
    public override string ToString()
    {
        return base.ToString();
    }
}
```

- }
- Напишете `class NewPersonComparer : IComparer<Person>` и добавете
- `IDictionary<SalaryEmployee, BankAccount> bankMapSalary =`
- `new SortedDictionary<SalaryEmployee, BankAccount>(new NewPersonComparer());`

Обяснете защо обект от `NewPersonComparer()` може да се подаде за параметър в конструктора на `SortedDictionary<SalaryEmployee, BankAccount>()`, който очаква обект от `Comparer< SalaryEmployee >`

Problem No.3

Banks lend money to each other. In tough economic times, if a bank goes bankrupt, it may not be able to pay back the loan. A bank's total assets are its current balance plus its loans to other banks. The diagram below shows five banks.



The banks' current balances are 25, 125, 175, 75, and 181 million dollars, respectively. The directed edge from node 1 to node 2 indicates that bank 1 lends 40 million dollars to bank 2. If a bank's total assets are under a certain limit, the bank is unsafe. The money it borrowed cannot be returned to the lender, and the lender cannot count the loan in its total assets. Consequently, the lender may also be unsafe, if its total assets are under the limit.

Write a program to find all the unsafe banks. Your program **reads the input** as follows. It first **reads two integers** `n` and `limit`, where `n` indicates **the number of banks** and `limit` is the **minimum total assets** for keeping a bank safe. It **then reads `n` lines** that **describe the information** for `n` **banks with IDs from 0 to `n-1`**.

The **first number** in the line is the bank's **balance**, the **second number** indicates the number of banks that borrowed money from the bank, and the rest are pairs of two numbers. Each pair describes a borrower. The **first number** in the pair is the **borrower's ID** and the **second** is the **amount borrowed**. For example, the **input for the five banks shown above** is as follows (note that the **limit** is 201):

```

5 201
25 2 1 100.5 4 320.5
125 2 2 40 3 85
175 2 0 125 3 75
75 1 0 125
181 1 2 125

```

The total assets of bank 3 are (75 + 125), which is under 201, so bank 3 is unsafe. After bank 3 becomes unsafe, the total assets of bank 1 fall below (125 + 40). Thus, bank 1 is also unsafe. The output of the program should be **Unsafe banks are 3 1**

(Hint: Use a **two-dimensional array** `borrowers` to represent loans. `borrowers[i][j]` indicates the loan that bank `i` loans to bank `j`. Once bank `j` becomes unsafe, `borrowers[i][j]` should be set to 0.)

Problem No.4

Study the [Graph representation in C#](#) and use it to solve problem 1a, where **the data is read from a text file**, located of the file system with a DialogBox. **Validate the data** input with regular expressions.

Hint:

Remember

```
string[] lines = File.ReadAllLines(txtProxyListPath.Text);  
List<string> listLines = new List<string>(lines);  
Parallel.ForEach(listLines, line => { //Your stuff });
```