LAB SESSION-5

ID NO:2100090162
Jaya Krishna

PRE-**LAB**

1.What are the types of Interfaces? Differentiate between Abstraction and Interfaces? How an Interface is used to implement an application?

Solution:

1. Types of interfaces: Marker interfaces (without any methods), Functional interfaces (single abstract method), and Normal interfaces (with multiple methods).
2. Abstraction is a concept where we hide complex implementation details and show only the necessary features. Interfaces, on the other hand, define a contract for classes to implement specific methods and properties, promoting code consistency and reusability.
3. Abstraction focuses on hiding implementation details, while interfaces focus on defining a contract for classes to adhere to.
4. Abstraction is achieved using abstract classes and methods, while interfaces provide a way to achieve abstraction through method signatures.
5. Interfaces are used in applications to define a set of methods and properties that classes must implement, allowing for polymorphism and loosely coupled code design.

IN-LAB:

TASK1: To create classes **Deposit** (bank account), **BaseDeposit** (regular deposit), **SpecialDeposit** (special deposit), **LongDeposit** (long-term deposit), **Client** (bank client) with set functionality.

1. To create abstract class **Deposit** and declare within it:

- Public money property only for reading **Amount** (deposit amount)
- Public integer property only for reading **Period** (time of deposit in months)
- Constructor (for calling in class-inheritor) with parameters **depositAmount** and **depositPeriod**, which creates object deposit with specified sum for specified period.
- Abstract method **Income**, which returns money value – amount of income from deposit. Income is the difference between sum, withdrawn from deposit upon expiration date and deposited sum.

2. To create classes that are inheritors of the class **Deposit**, which determine different options of deposit interest addition – class **BaseDeposit**, class **SpecialDeposit** and class **LongDeposit**. To implement in each class a constructor with parameters **amount** and **period**, which calls constructor of parent class.
3. For each inheritor class – to implement own interest addition scheme and accordingly profit margin definitions, overriding abstract method **Income** in each class.

BaseDeposit implies each month 5% of interest from current deposit sum. Each following month of income is calculated from the sum, which was received by adding to current income sum of the previous month and is rounded to hundredth.

Example: Base amount – 1000,00

In a month – 105,00; income amount – 50,00

In two months – 1102,50; income amount – 102,50

In three months – 1157,62; income amount – 157,62

SpecialDeposit implies income addition each month, amount of which (in percent) equals to deposit expiration period. If during the first month 1% is added, during the second month – 2% from the sum obtained after first month and so on.

Example: Base amount – 1000,00

In a month – 1010,00; income amount – 10,00

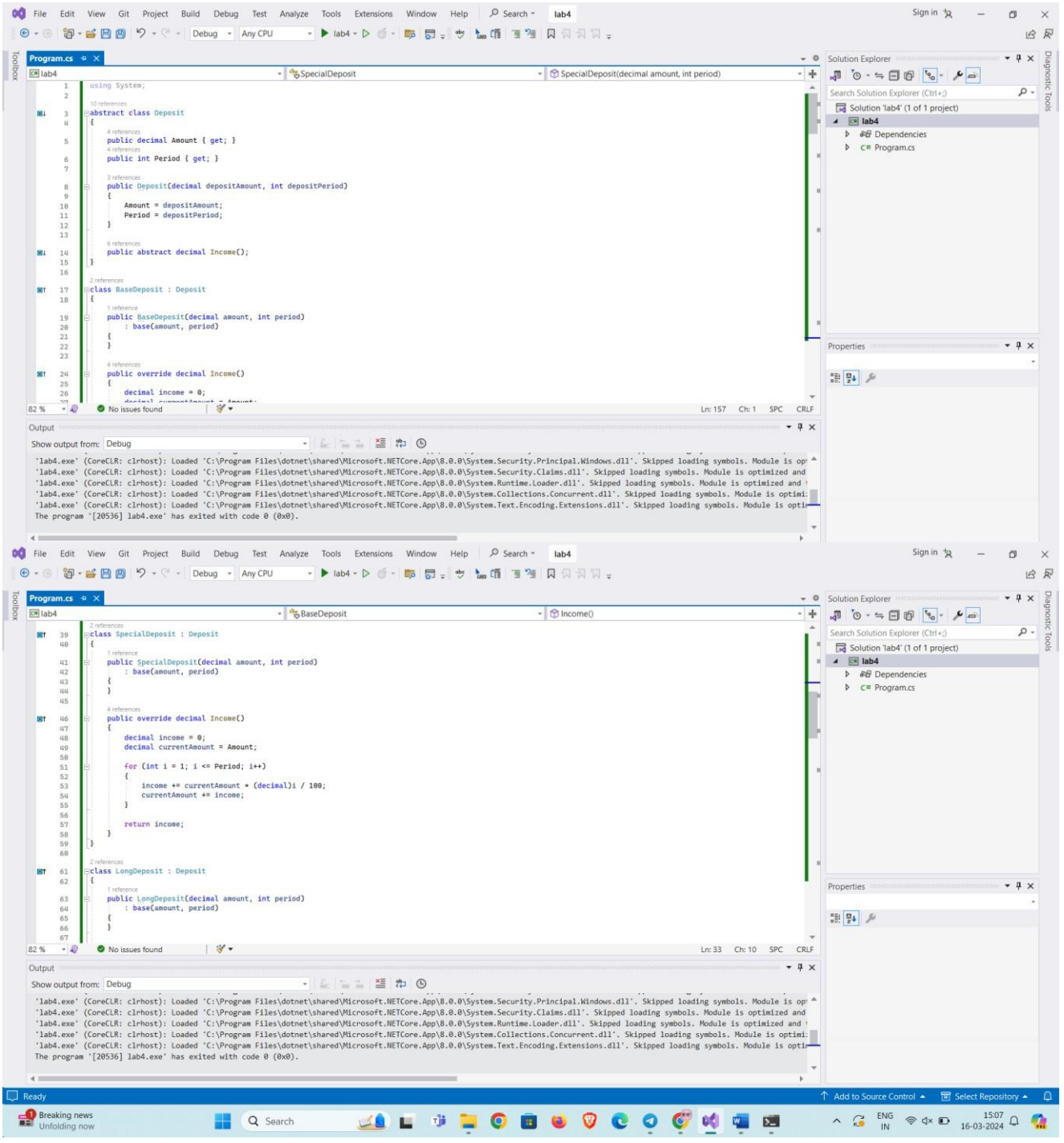
In two months – 1030,20; income amount – 30,20

LongDeposit implies that during first 6 months, no percent is added to client's deposit, but starting from 7th month, each month percent addition is 15% from current deposit sum, thus encouraging client to make long-term deposits.

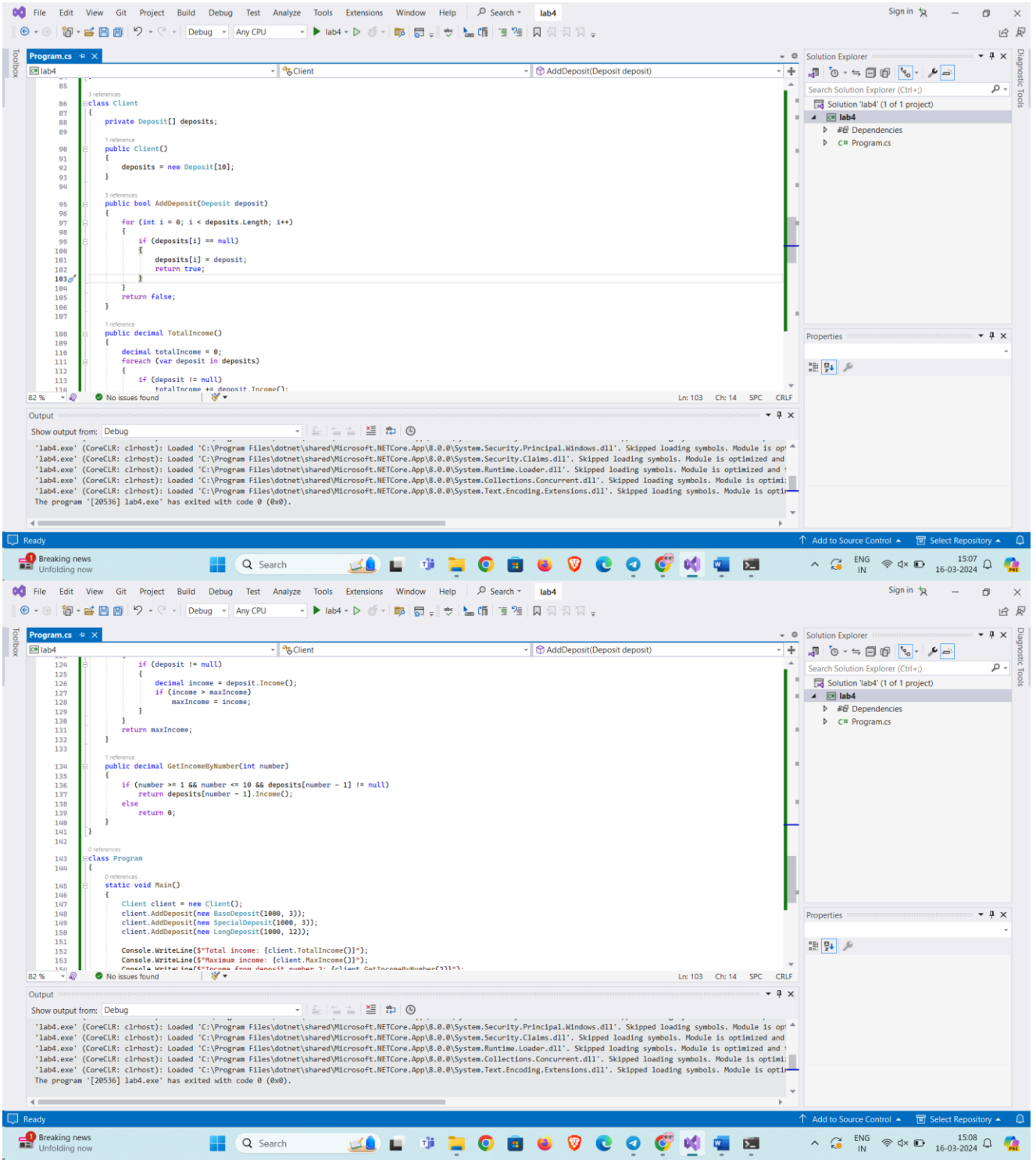
4. To create class **Client** (bank client) and declare within it:

- Private field **deposits** (client deposits) – objects array of type Deposit
- Constructor without parameters, which creates empty array deposits consisting of 10 elements
- Method **AddDeposit** with parameter **deposit** for adding regular, special or long-term account into array on the first empty spot and returning true, or returning false, if accounts number limit is depleted (no empty space in array).
- Method **TotalIncome**, returning total income amount based on all client's deposits upon deposits expiration.
- Method **MaxIncome**, returning maximum deposit income of all client's deposits upon deposits expiration.
- Method **GetIncomeByNumber** with integer parameter **number** (deposit number, which equals its index in array, increased by one), returning income from deposit with such number. If deposit with such number does not exist, method returns 0 value.

Solution:

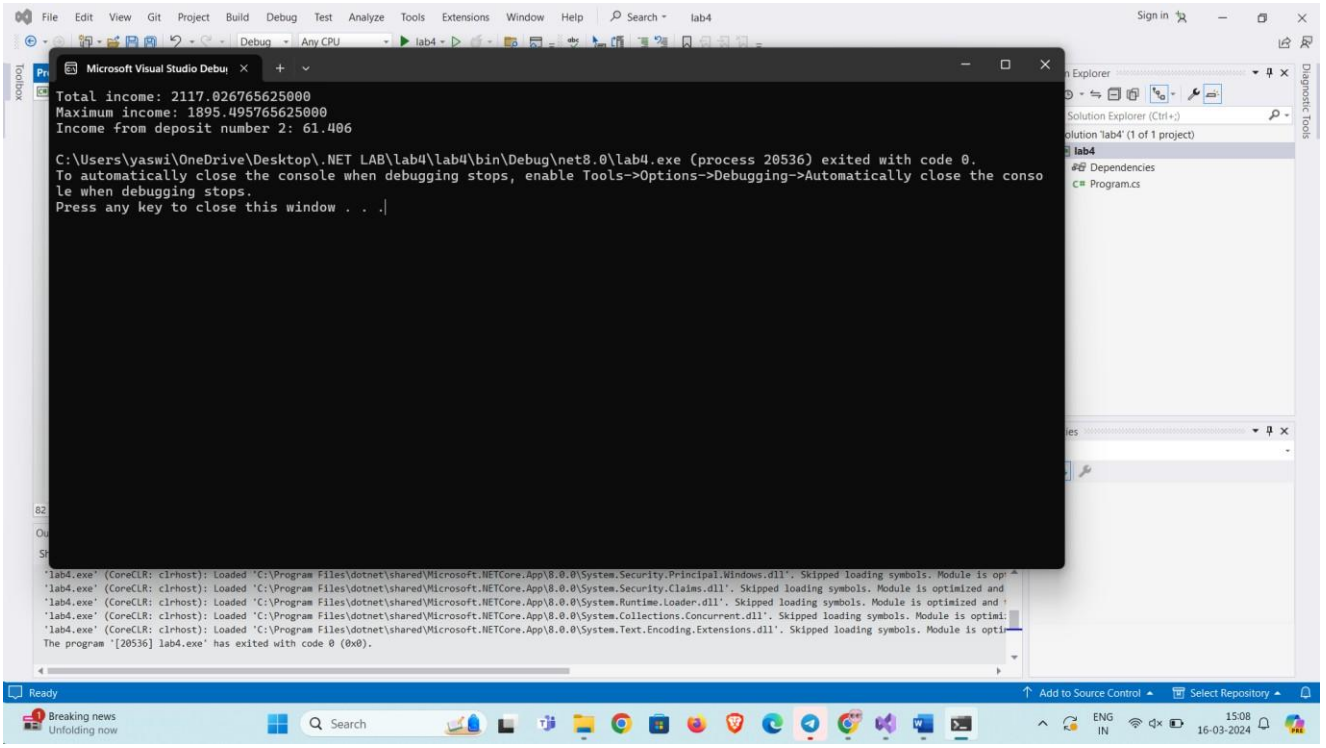

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
SpecialDeposit
SpecialDeposit(decimal amount, int period)
using System;
abstract class Deposit
{
public decimal Amount { get; }
public int Period { get; }
public Deposit(decimal depositAmount, int depositPeriod)
{
Amount = depositAmount;
Period = depositPeriod;
}
public abstract decimal Income();
}
class BaseDeposit : Deposit
{
public BaseDeposit(decimal amount, int period)
: base(amount, period)
{
}
public override decimal Income()
{
decimal income = 0;
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
No issues found
Ln: 157 Ch: 1 SPC CRLF
Output
Show output from: Debug
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is op
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Claims.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Runtime.Loader.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optimi
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opti
The program '[20536] lab4.exe' has exited with code 0 (0x0).

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
BaseDeposit
Income()
class SpecialDeposit : Deposit
{
public SpecialDeposit(decimal amount, int period)
: base(amount, period)
{
}
public override decimal Income()
{
decimal income = 0;
decimal currentAmount = Amount;
for (int i = 1; i <= Period; i++)
{
income += currentAmount * (decimal)i / 100;
currentAmount += income;
}
return income;
}
}
class LongDeposit : Deposit
{
public LongDeposit(decimal amount, int period)
: base(amount, period)
{
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
No issues found
Ln: 33 Ch: 10 SPC CRLF
Output
Show output from: Debug
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is op
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Claims.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Runtime.Loader.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optimi
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opti
The program '[20536] lab4.exe' has exited with code 0 (0x0).
Ready
Add to Source Control
Select Repository
Breaking news
Unfolding now
Search
ENG
IN
15:07
16-03-2024


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
Client
AddDeposit(Deposit deposit)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is op
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Claims.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Runtime.Loader.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optimi
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opti
The program '[20536] lab4.exe' has exited with code 0 (0x0).
No issues found
Ln: 103 Ch: 14 SPC CRLF
Ready
Add to Source Control
Select Repository
Breaking news
Unfolding now
Search
ENG
IN
15:07
16-03-2024

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
Client
AddDeposit(Deposit deposit)
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Principal.Windows.dll'. Skipped loading symbols. Module is op
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Security.Claims.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Runtime.Loader.dll'. Skipped loading symbols. Module is optimized and
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Collections.Concurrent.dll'. Skipped loading symbols. Module is optimi
'lab4.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\8.0.0\System.Text.Encoding.Extensions.dll'. Skipped loading symbols. Module is opti
The program '[20536] lab4.exe' has exited with code 0 (0x0).
No issues found
Ln: 103 Ch: 14 SPC CRLF
Ready
Add to Source Control
Select Repository
Breaking news
Unfolding now
Search
ENG
IN
15:08
16-03-2024

OUTPUT:

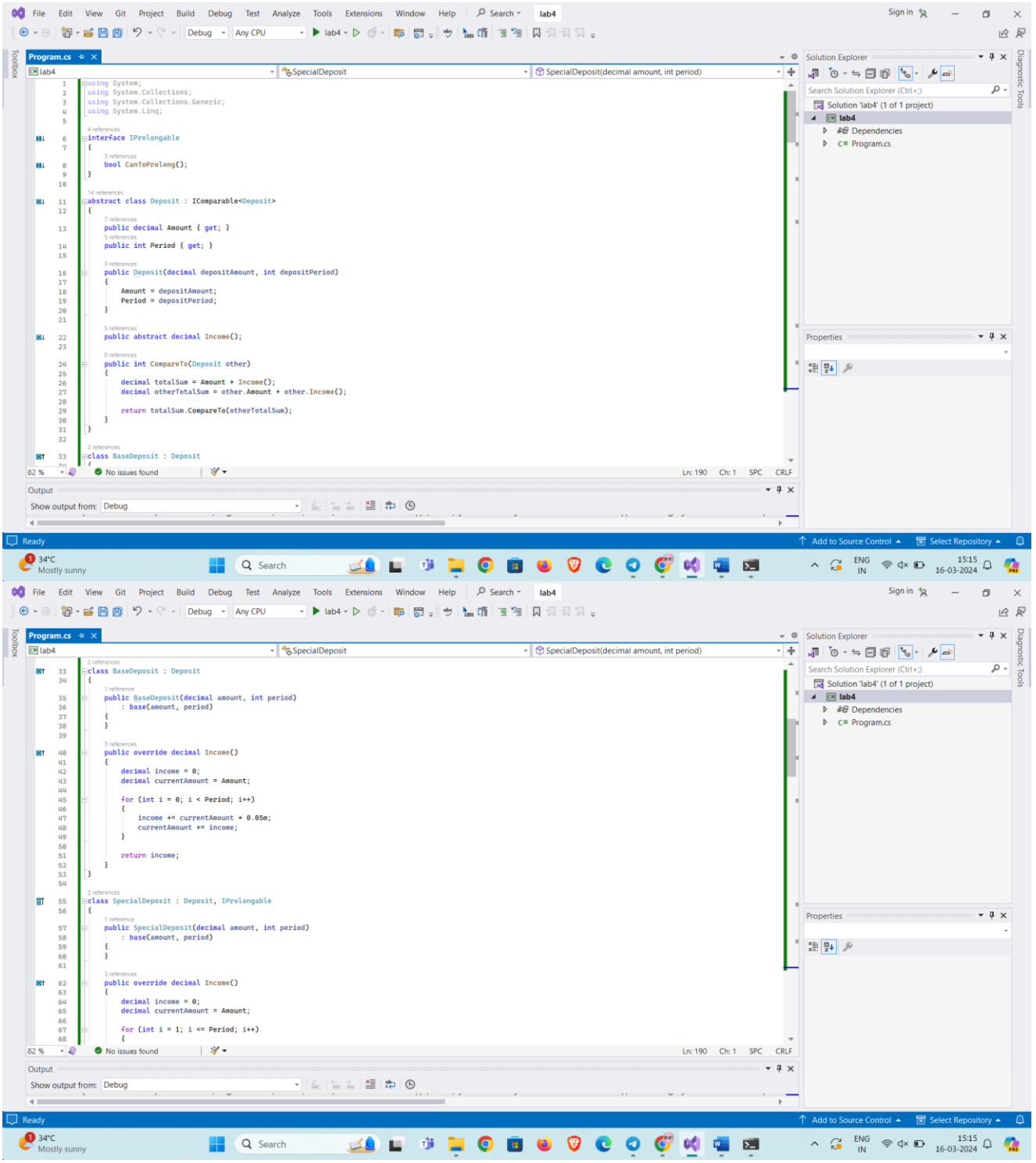


TASK 2: To add the following new functionalities to the project created in task Aggregation:

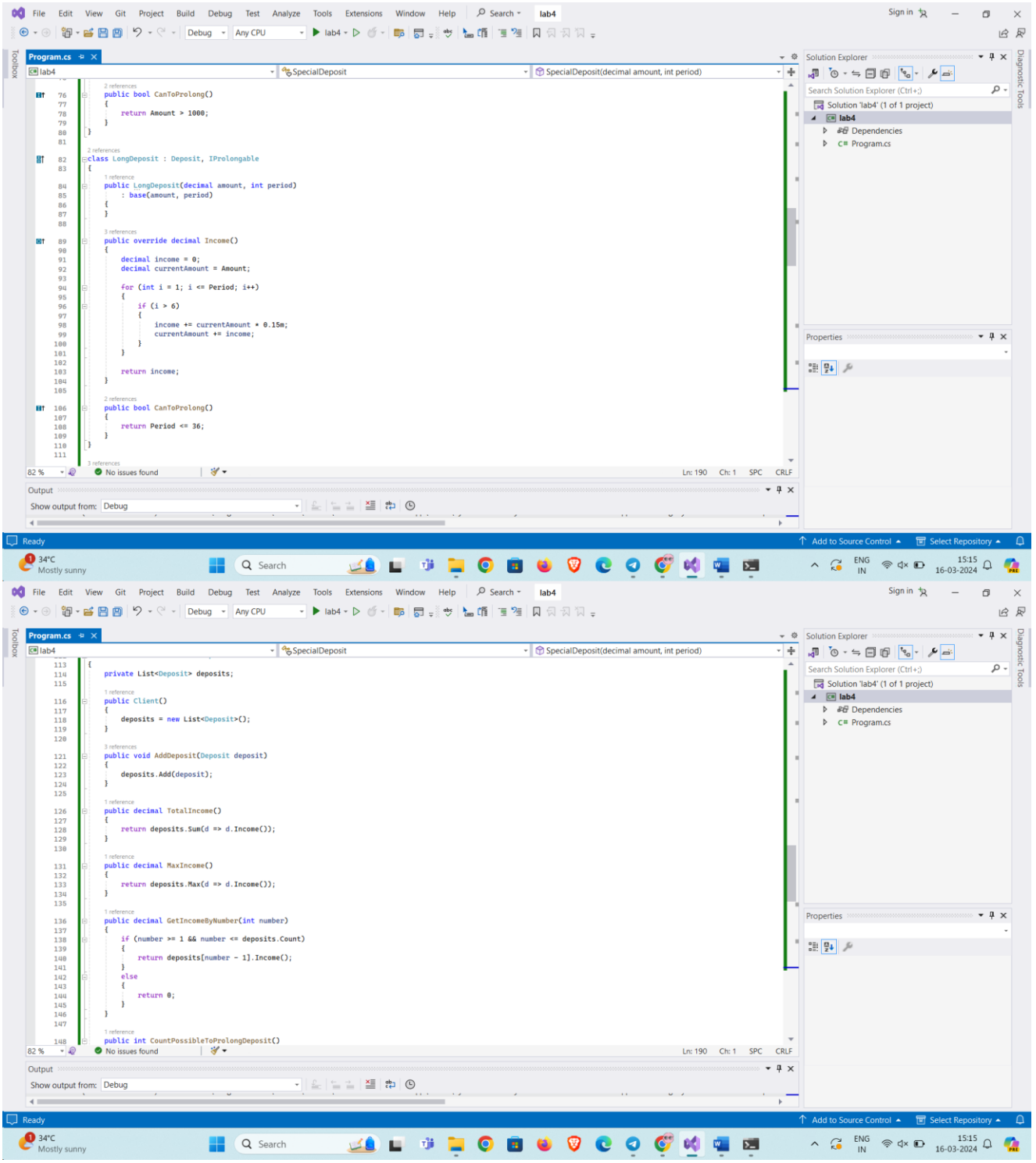
1. To create interface **Iprolongable** (prolonging deposit) and declare within it method **CanToProlong** without parameters that returns logic value true or false, depending on the fact whether this specific deposit can be prolonged or not.
2. To implement interface **IProlongable** in classes **SpecialDeposit** and **LongDeposit**.
3. In addition, special deposit (**SpecialDeposit**) can be prolonged only when more than 1000 UAH were deposited, and long-term deposit (**LongDeposit**) can be prolonged if the period of deposit is no longer than 3 years.
4. To implement standard generic interface **IComparable<Deposit>** in abstract class **Deposit**. Total sum amount (sum deposited plus interest during entire period) should be considered as comparison criteria of **Deposit** instances.
5. To implement additionally in class **Client**:

- interface **IEnumerable<Deposit>**.
- Method **SortDeposits**, which performs deposits sorting in array **deposits** in descending order of total sum amount on deposit upon deposit expiration.
- Method **CountPossibleToProlongDeposit**, which returns integer – amount of current client's deposits that can be prolonged.

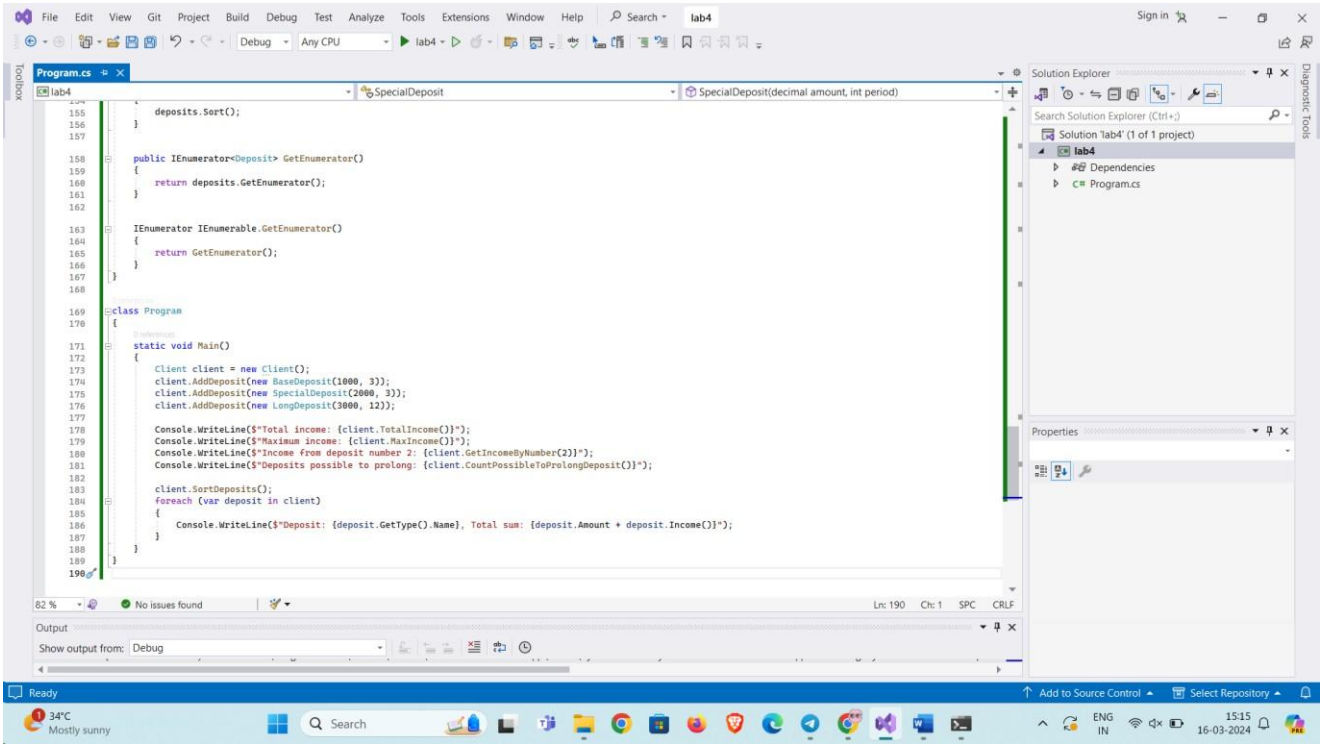
Solution:


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search lab4 Sign in
Debug Any CPU lab4
Program.cs
lab4 SpecialDeposit SpecialDeposit(decimal amount, int period)
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
4 references
interface IProlongable
{
3 references
bool CanToProlong();
}
14 references
abstract class Deposit : IComparable<Deposit>
{
7 references
public decimal Amount { get; }
5 references
public int Period { get; }
3 references
public Deposit(decimal depositAmount, int depositPeriod)
{
Amount = depositAmount;
Period = depositPeriod;
}
5 references
public abstract decimal Income();
0 references
public int CompareTo(Deposit other)
{
decimal totalSum = Amount + Income();
decimal otherTotalSum = other.Amount + other.Income();
return totalSum.CompareTo(otherTotalSum);
}
}
2 references
class BaseDeposit : Deposit
82 % No issues found Ln: 190 Ch: 1 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
Ready
Add to Source Control Select Repository
34°C Mostly sunny
Search
ENG IN 15:15 16-03-2024

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search lab4 Sign in
Debug Any CPU lab4
Program.cs
lab4 SpecialDeposit SpecialDeposit(decimal amount, int period)
2 references
class BaseDeposit : Deposit
{
1 reference
public BaseDeposit(decimal amount, int period)
: base(amount, period)
{
}
3 references
public override decimal Income()
{
decimal income = 0;
decimal currentAmount = Amount;
for (int i = 0; i < Period; i++)
{
income += currentAmount * 0.05m;
currentAmount += income;
}
return income;
}
}
2 references
class SpecialDeposit : Deposit, IProlongable
{
1 reference
public SpecialDeposit(decimal amount, int period)
: base(amount, period)
{
}
3 references
public override decimal Income()
{
decimal income = 0;
decimal currentAmount = Amount;
for (int i = 1; i <= Period; i++)
{
82 % No issues found Ln: 190 Ch: 1 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
Ready
Add to Source Control Select Repository
34°C Mostly sunny
Search
ENG IN 15:15 16-03-2024

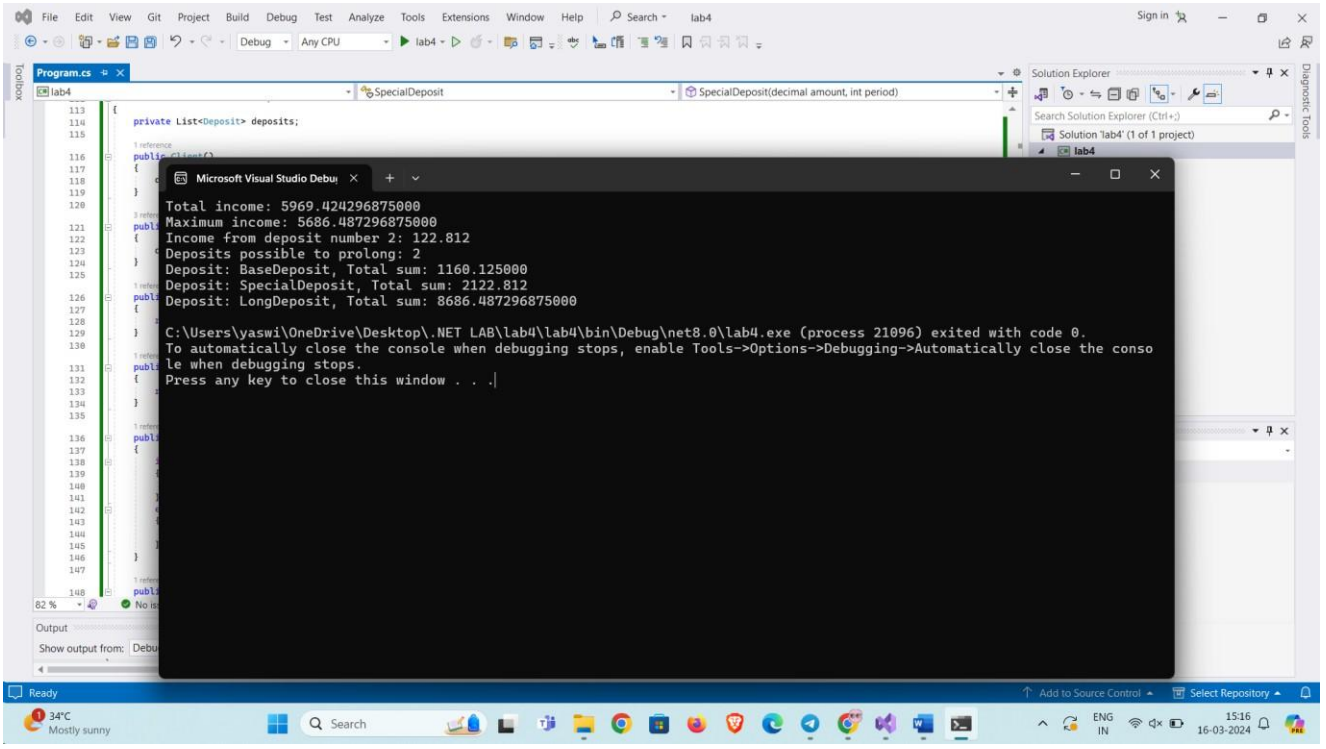

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search lab4 Sign in
Debug Any CPU lab4
Program.cs
lab4 SpecialDeposit SpecialDeposit(decimal amount, int period)
2 references
public bool CanToProlong()
{
return Amount > 1000;
}
}
2 references
class LongDeposit : Deposit, IProlongable
{
1 reference
public LongDeposit(decimal amount, int period)
: base(amount, period)
{
}
3 references
public override decimal Income()
{
decimal income = 0;
decimal currentAmount = Amount;
for (int i = 1; i <= Period; i++)
{
if (i > 6)
{
income += currentAmount * 0.15m;
currentAmount += income;
}
}
return income;
}
2 references
public bool CanToProlong()
{
return Period <= 36;
}
}
3 references
82 % No issues found Ln: 190 Ch: 1 SPC CRLF
Output
Show output from: Debug
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Toolbox
Diagnostic Tools
Ready
Add to Source Control
Select Repository
34°C Mostly sunny
Search
ENG IN
15:15 16-03-2024

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search lab4 Sign in
Debug Any CPU lab4
Program.cs
lab4 SpecialDeposit SpecialDeposit(decimal amount, int period)
{
private List<Deposit> deposits;
1 reference
public Client()
{
deposits = new List<Deposit>();
}
3 references
public void AddDeposit(Deposit deposit)
{
deposits.Add(deposit);
}
1 reference
public decimal TotalIncome()
{
return deposits.Sum(d => d.Income());
}
1 reference
public decimal MaxIncome()
{
return deposits.Max(d => d.Income());
}
1 reference
public decimal GetIncomeByNumber(int number)
{
if (number >= 1 && number <= deposits.Count)
{
return deposits[number - 1].Income();
}
else
{
return 0;
}
}
1 reference
public int CountPossibleToProlongDeposit()
82 % No issues found Ln: 190 Ch: 1 SPC CRLF
Output
Show output from: Debug
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Toolbox
Diagnostic Tools
Ready
Add to Source Control
Select Repository
34°C Mostly sunny
Search
ENG IN
15:15 16-03-2024



```
        deposits.Sort();
    }

    public IEnumerator<Deposit> GetEnumerator()
    {
        return deposits.GetEnumerator();
    }

    IEnumerator IEnumerable.GetEnumerator()
    {
        return GetEnumerator();
    }
}

class Program
{
    static void Main()
    {
        Client client = new Client();
        client.AddDeposit(new BaseDeposit(1000, 3));
        client.AddDeposit(new SpecialDeposit(2000, 3));
        client.AddDeposit(new LongDeposit(3000, 12));

        Console.WriteLine($"Total income: {client.TotalIncome()}");
        Console.WriteLine($"Maximum income: {client.MaxIncome()}");
        Console.WriteLine($"Income from deposit number 2: {client.GetIncomeByNumber(2)}");
        Console.WriteLine($"Deposits possible to prolong: {client.CountPossibleToProlongDeposit()}");

        client.SortDeposits();
        foreach (var deposit in client)
        {
            Console.WriteLine($"Deposit: {deposit.GetType().Name}, Total sum: {deposit.Amount + deposit.Income()}");
        }
    }
}
```

OUTPUT:

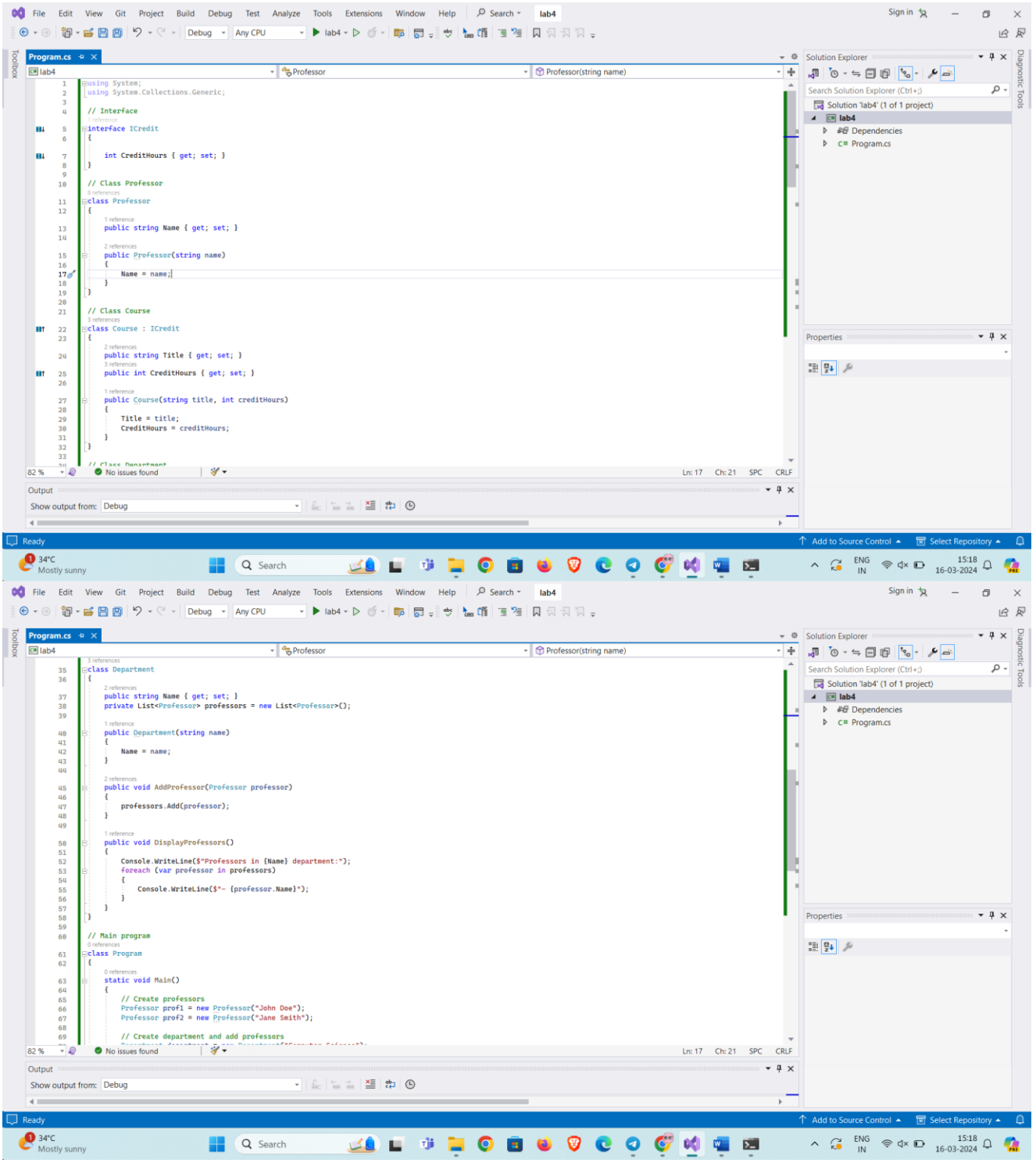


```
Total income: 5969.424296875000
Maximum income: 5686.487296875000
Income from deposit number 2: 122.812
Deposits possible to prolong: 2
Deposit: BaseDeposit, Total sum: 1160.125000
Deposit: SpecialDeposit, Total sum: 2122.812
Deposit: LongDeposit, Total sum: 8686.487296875000

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab4\lab4\bin\Debug\net8.0\lab4.exe (process 21096) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

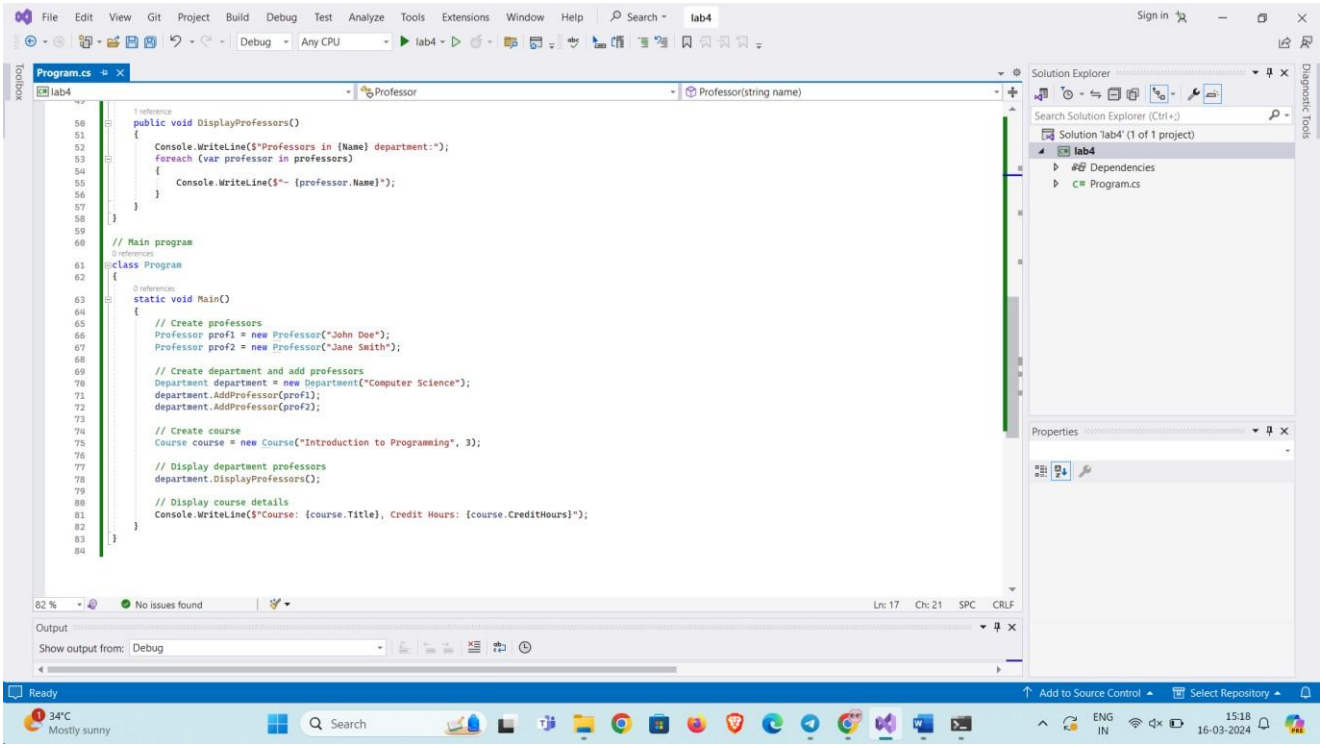
POST-LAB

1. Implement & Analyze a small Application that uses the Aggregation and Interface concepts?

Solution:

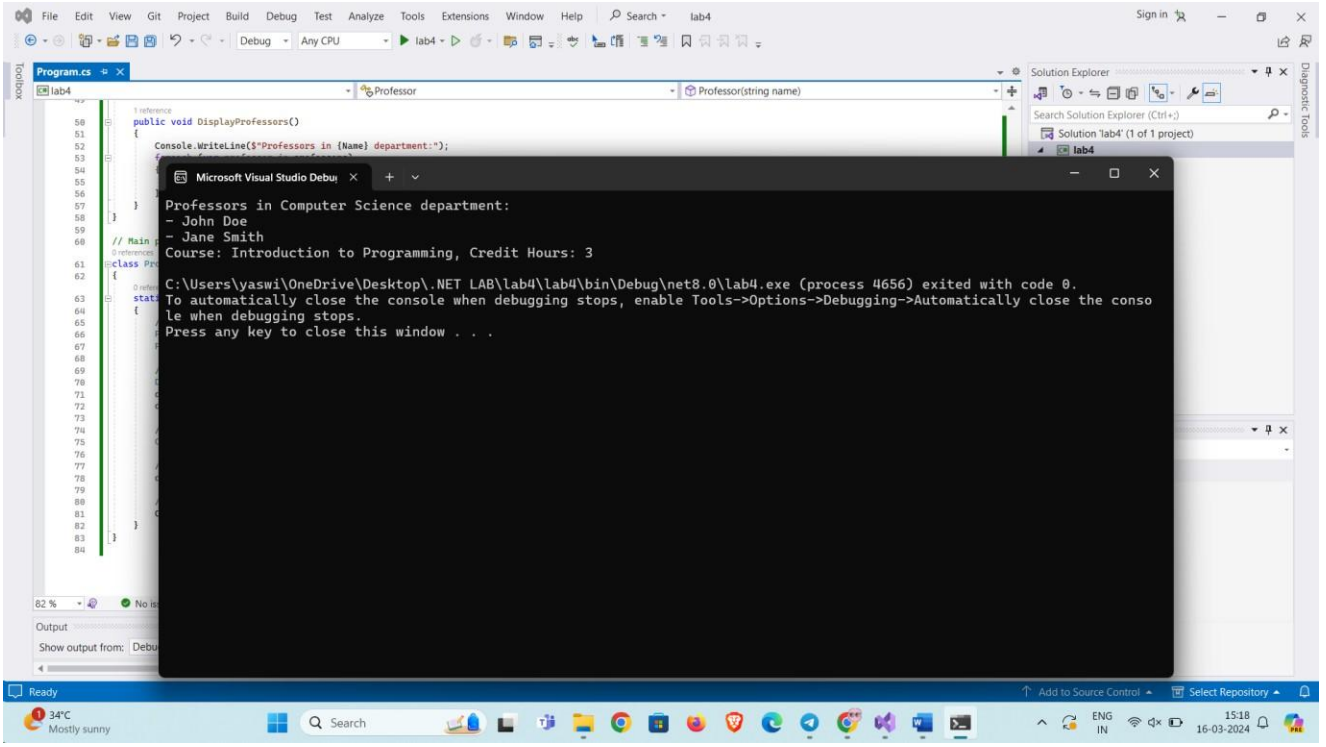

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
Professor
Professor(string name)
using System;
using System.Collections.Generic;
// Interface
interface ICredit
{
int CreditHours { get; set; }
}
// Class Professor
class Professor
{
public string Name { get; set; }
public Professor(string name)
{
Name = name;
}
}
// Class Course
class Course : ICredit
{
public string Title { get; set; }
public int CreditHours { get; set; }
public Course(string title, int creditHours)
{
Title = title;
CreditHours = creditHours;
}
}
82 %
No issues found
Ln: 17 Ch: 21 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
Ready
Add to Source Control
Select Repository
34°C
Mostly sunny
Search
ENG
IN
15:18
16-03-2024

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
lab4
Sign in
Debug
Any CPU
lab4
Program.cs
lab4
Professor
Professor(string name)
class Department
{
public string Name { get; set; }
private List<Professor> professors = new List<Professor>();
public Department(string name)
{
Name = name;
}
public void AddProfessor(Professor professor)
{
professors.Add(professor);
}
public void DisplayProfessors()
{
Console.WriteLine($"Professors in {Name} department:");
foreach (var professor in professors)
{
Console.WriteLine($"- {professor.Name}");
}
}
}
// Main program
class Program
{
static void Main()
{
// Create professors
Professor prof1 = new Professor("John Doe");
Professor prof2 = new Professor("Jane Smith");
// Create department and add professors
82 %
No issues found
Ln: 17 Ch: 21 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'lab4' (1 of 1 project)
lab4
Dependencies
Program.cs
Properties
Output
Show output from: Debug
Ready
Add to Source Control
Select Repository
34°C
Mostly sunny
Search
ENG
IN
15:18
16-03-2024



```
public void DisplayProfessors()
{
    Console.WriteLine($"Professors in {Name} department:");
    foreach (var professor in professors)
    {
        Console.WriteLine($"- {professor.Name}");
    }
}

// Main program
class Program
{
    static void Main()
    {
        // Create professors
        Professor prof1 = new Professor("John Doe");
        Professor prof2 = new Professor("Jane Smith");

        // Create department and add professors
        Department department = new Department("Computer Science");
        department.AddProfessor(prof1);
        department.AddProfessor(prof2);

        // Create course
        Course course = new Course("Introduction to Programming", 3);

        // Display department professors
        department.DisplayProfessors();

        // Display course details
        Console.WriteLine($"Course: {course.Title}, Credit Hours: {course.CreditHours}");
    }
}
```

OUTPUT:



```
Professors in Computer Science department:
- John Doe
- Jane Smith
Course: Introduction to Programming, Credit Hours: 3

C:\Users\yaswi\OneDrive\Desktop\.NET LAB\lab4\lab4\bin\Debug\net8.0\lab4.exe (process 4656) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```