|  |  |
| --- | --- |
| **Programming 2**  Diploma in IT / DS / CSF  Year 1 (2022/23) Semester 2 | Week 6 |
| **2 hours** |
| **Practical 6: Inheritance** | |

**Objectives**

At the end of this practical, the students should be able to:

* write their own classes and subclasses
* use their own classes

|  |
| --- |
| **IMPORTANT**   * Create a folder, W**eek06.** * Create a new Console App (.NET Core) project, **Snnnnnnnn\_MyBankApp**, in the **Week06** folder created above *(note:* ***Snnnnnnnn*** *is your Student Number)*. * At the end of the session, copy the folder **Week06** folder (which contains all your work) to PRG2 network folder: **\\ictspace.ict.np.edu.sg\PRG2**. |

The ***savings account*** is a type of bank account that earns interest based on some prevailing interest rate. Assuming the **BankAccount** class and the **SavingsAccount** class are defined as shown below.

|  |
| --- |
| **BankAccount** |
| -accNo:string  -accName:string  -balance:double |
| +BankAccount()  +BankAccount(string,string,double)  +Deposit(double)  +Withdraw(double):bool  +ToString():string |

|  |
| --- |
| **SavingsAccount** |
| -rate:double |
| +SavingsAccount()  +SavingsAccount(string,string,double,double)  +CalculateInterest():double  +ToString():string |

1. Implement the **BankAccount** class.

Note:

* **Deposit(double)** deposits a given amount to the bank account. No value is returned.
* **Withdraw(double)** withdraws a given amount from the bank account if there is sufficient balance. The method returns true if withdrawal is successful, false otherwise.
* **ToString()** returns the attribute values with description as a string.

1. Implement the **SavingsAccount** class as subclass of **BankAccount**.

Note:

* **rate** records the interest rate in percentage.
* **CalculateInterest()** calculates and returns the interest based on the account balance and interest rate using the formula

Balance \* Interest rate / 100.

* **ToString()** returns the attribute values with description as a string. You should reuse the method defined in the superclass.

1. In the **Program.cs**, do the following:
2. Read the **savings\_account.csv** file to create **SavingsAccount** objects and store the objects in the List called **savingsAccList**.

Note: you should do this task in a method. Main() method will call this method to initialize data in the list.

1. Write a DisplayMenu() method to display the following menu:

|  |
| --- |
| Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: |

1. Call the DisplayMenu() method to display the menu and perform task according to the selected option until user chooses to exit.

Option 1: Display all records

* Write a method to display all **SavingsAccount** objects stored in **savingsAccList**. The method header should be

void DisplayAll(List<SavingsAccount> sList)

* Call the method when option 1 is selected.

Sample output (value underlined is the user input):

|  |
| --- |
| Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **1**  Acc No:111-01 Acc Name:Bob Balance:2000 Rate:2  Acc No:111-02 Acc Name:Mary Balance: 000 Rate:2  Acc No:111-03 Acc Name:Tom Balance:1000 Rate:1.5 |

Option 2: Deposit

* Write a method to search for the savings account with the following method header:

SavingsAccount Search(List<SavingsAccount> sList, string accNo)

The method returns the **SavingsAccount** object if found, or null if not found.

* Prompt the user to input the savings account number thatthe amount is to be deposited to.
* Call the Search() method to search for the account.
* Prompt the user for the amount to be deposited, and call Deposit() method on the **SavingsAccount** object.
* Display the details of the object.

Sample output (values underlined are user input):

|  |
| --- |
| Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **2**  Enter the Account Number: 111-01  Amount to deposit: 300  $300 deposited successfully  Acc No:111-01 Acc Name:Bob Balance:2300 Rate:2  Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **2**  Enter the Account Number: 111-05  Unable to find account number. Please try again. |

Option 3: Withdraw

* Prompt the user to input the savings account number that the amount is to withdrawn from.
* Call the Search() method to search for the account.
* Prompt user for the amount to be withdrawn.
* Call Withdraw() method on the **SavingsAccount** object.
* If there are insufficient funds, the program should display an alert.
* Display the details of the object if withdrawal is successful.

Sample output (values underlined are user input):

|  |
| --- |
| Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **3**  Enter the Account Number: 111-03  Amount to withdraw: 2000  Insufficient funds.  Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **3**  Enter the Account Number: 111-03  Amount to withdraw: 500  $500 withdrawn successfully  Acc No:111-03 Acc Name:Tom Balance:500 Rate:1.5  Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **3**  Enter the Account Number: 111-05  Unable to find account number. Please try again.  Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [0] Exit  Enter option: **0**  ---------  Goodbye!  --------- |

**Additional Features**

1. Add an option 4 to display details of all accounts together with the interest.

Sample output (values underlined are user input)

|  |
| --- |
| Menu  [1] Display all accounts  [2] Deposit  [3] Withdraw  [4] Display details  [0] Exit  Enter option: **4**  Acc No Acc Name Balance Rate Interest  111-01 Bob 2000.00 2.00 40.00  111-02 Mary 5000.00 2.00 100.00  111-03 Tom 1000.00 1.50 15.00 |

**Advanced**

The class diagram for an application is given below.

|  |
| --- |
| **Circle** |
| -radius:double |
| +Circle()  +Circle(double)  +CalculateArea():double  +ToString():string |

|  |
| --- |
| **Cylinder** |
| -length:double |
| +Cylinder()  +Cylinder(double,double)  +CalculateArea():double  +CalculateVolume():double  +ToString():string |

1. Create a Console project **MyShapeApp** in Visual Studio 2022.
2. Implement the **Circle** class and add to the project.
3. Implement the **Cylinder** class and add to the project.
4. Write an Application program to do the following:

(a) create a circle (object), circle1, with radius = 5.0

(b) create a cylinder (object), cylinder1, with the following attributes:  
 radius:5.0, length:10.0

(c) display a simple menu as shown below:

|  |
| --- |
| ---------------- M E N U ----------------- [1] Change the radius of circle [2] Change the radius of cylinder [3] Change the length of cylinder [4] Display the area of circle [5] Display the surface area of cylinder  [6] Display the volume of cylinder [0] Exit ------------------------------------------ Enter your option : |

Process the option selected by the user as follows:

Option 1: Change the radius of circle

* *display current radius of the circle*
* *prompt user to enter the new radius*
* *set the radius of the circle to the new radius*

Option 2: Change the radius of cylinder

* *display current radius of the cylinder*
* *prompt user to enter the new radius*
* *set the radius of the cylinder to the new radius*

Option 3: Change the length of cylinder

* *display current length of the cylinder*
* *prompt user to enter the new length*
* *set the length of the cylinder to the new length*

Option 4: Display the area of circle

* *display the area of the circle*

Option 5: Display the surface area of cylinder

* *display the surface area of the cylinder*

Option 6: Display the volume of cylinder

* *display the volume of the cylinder*

Option 0: Exit

**Note**

* *You should implement methods where appropriate.*
* *You should implement and test ONE option at a time.*

**Plagiarism Warning:**

**If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical. Disciplinary action may also be taken.**

**Similar action will be taken for student who allows other student(s) to copy his/her work.**