Trường Đại học Khoa học Tự nhiên, ĐHQG - HCM
Khoa Điện tử - Viễn thông

Thực hành Kỹ thuật lập trình nâng cao

Lab 3: KẾ THỪA VÀ ĐA HÌNH

Lớp	MSSV	Họ và tên
21DienTu	21200201	Phạm Hoàng Sơn

1 Bài 1

```
internal class UserInformation
{
   private string userName;
   private int userAge;
   private uint userID;
   private double baseSalary;

   public string UserName
   {
      get { return userName; }
      set
      {
         userName = value;
      }
   }
   public int UserAge
   {
      get { return userAge; }
      set
      {
         if (value > 18)
         {
            userAge = value;
         }
         else
         {
            throw new UserAccountException();
         }
      }
   }
   public uint UserID
   {
      get { return userID; }
      set
      {
         if (!(value < 1000 || value > 9999))
```

```
        {
            userID = value;
        }
        else
        {
            throw new UserAccountException();
        }
    }
}
public UserInformation()
{ }
public UserInformation(string name, int age, uint id)
{
    UserName = name;
    UserAge = age;
    UserID = id;
}
public override string ToString()
{
    return String.Format($"[{userName}, {userAge}, {userID}]");
}
//lab3
public double BaseSalary
{
    get { return baseSalary; }
    set { baseSalary = value; }
}
public UserInformation(string name, int age, uint id, double baseSalary) :
    this(name, age, id)
{
    BaseSalary = baseSalary;
}
public virtual double GetSalary()
{
    return BaseSalary;
}
}
```

Listing 1: UserInformation.cs

```
public static void main()
{
    UserInformation[] users = new UserInformation[6]
    {
        new Employee("Employee 0", 23, 1000, 10_000_000),
        new Employee("Employee 1", 21, 1001, 11_000_000),
        new Manager ("Manager 2", 27, 1012, 12_000_000),
        new Manager ("Manager 3", 24, 1013, 18_000_000),
        new Director("Director 4", 32, 1024, 13_000_000),
        new Director("Director 5", 29, 1025, 14_000_000),
    };
    foreach (var user in users)
    {
        Console.WriteLine($"{user.UserName}: {user.GetSalary()}\n");
    }
    Console.WriteLine("_______ HIGH TO LOW _________");
    for (int i = 0; i < 6; i++)
```

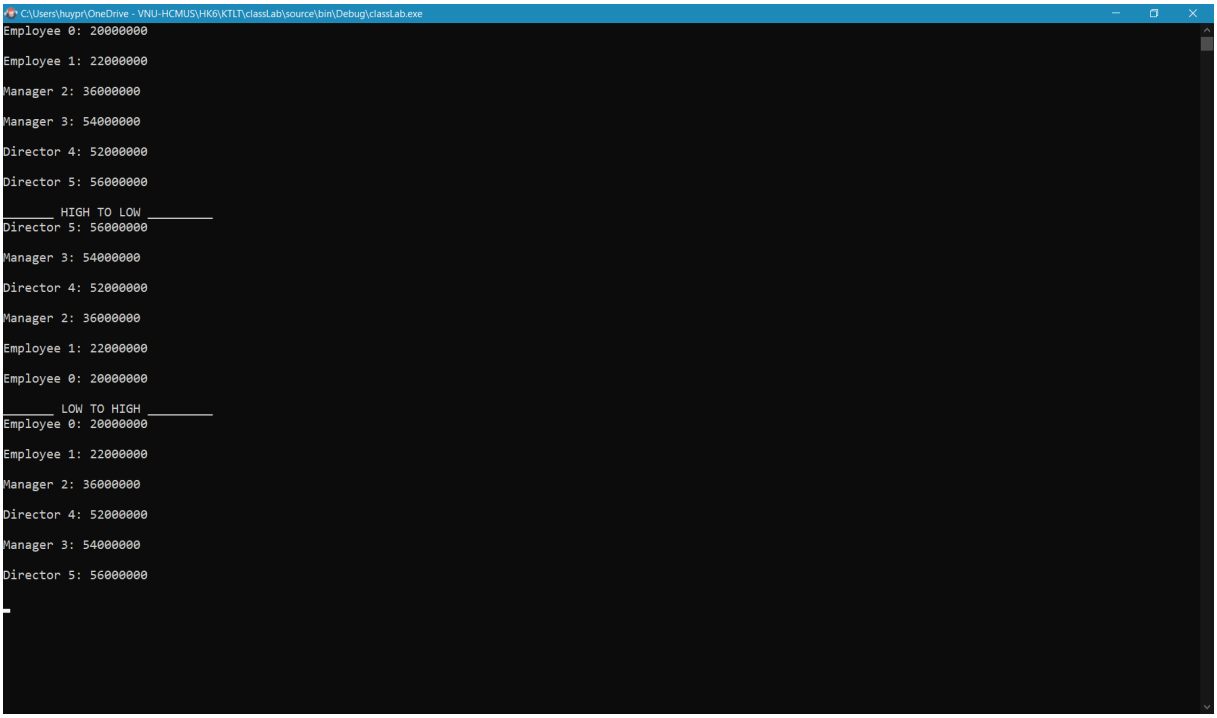
```
    {
        for (int j = 0; j < 6 - i - 1; j++)
        {
            if (users[j].GetSalary() < users[j + 1].GetSalary())
            {
                var temp = users[j];
                users[j] = users[j + 1];
                users[j + 1] = temp;
            }
        }
    }
    foreach (var user in users)
    {
        Console.WriteLine($"{user.UserName}: {user.GetSalary()}\n");
    }

    Console.WriteLine("_______ LOW TO HIGH _________");
    for (int i = 0; i < 6; i++)
    {
        for (int j = 0; j < 6 - i - 1; j++)
        {
            if (users[j].GetSalary() > users[j + 1].GetSalary())
            {
                var temp = users[j];
                users[j] = users[j + 1];
                users[j + 1] = temp;
            }
        }
    }
    foreach (var user in users)
    {
        Console.WriteLine($"{user.UserName}: {user.GetSalary()}\n");
    }
}
```

Listing 2: Program.cs

2 Bài 2

```
public static void main()
{
    try
    {
        BankAccount alexAccount = null;
        GetUserBankAccountInformation(out alexAccount);
        Console.WriteLine("Please type in your password to show current balance");
        alexAccount.VerifyPassword(Console.ReadLine());
        Console.WriteLine($"{alexAccount.CalculateInterestMoney()}");
        Console.WriteLine($"interest rate: {BankAccount.InterestRate}");
    }
    catch (AccessDeniedException)
    {
       Console.WriteLine("Access denied!");
    }
    catch (OverflowException ex)
```



Hình 1: Giá trị trả về của hàm `GetSalary()` trước khi được sắp xếp và sau khi được sắp xếp

```
    {
        Console.WriteLine(ex.StackTrace);
    }
    Console.ReadLine();
}
```

Listing 3: `Program.cs`

```
internal class BankAccount
{
    private string password;
    private bool isPasswordVerified;
    private double balance;

    public double Balance
    {
        get
        {
            CheckAccessPermission();
            return balance;
        }
        set
        {
            CheckAccessPermission();
            balance = value;
        }
    }
    public static double InterestRate { get; set; }
    public UserInformation UserInf { get; set; }
    public BankAccount(string userName, int userAge, uint userID, double Balance,
        string Password)
```

```
    {

        UserInf = new UserInformation(userName, userAge, userID);
        this.password = Password;
        this.balance = Balance;
        isPasswordVerified = false;
    }
    static BankAccount()
    {
        InterestRate = 0.05;
    }
    public double CalculateInterestMoney()
    {
        return Balance * InterestRate;
    }
    public void VerifyPassword(string password)
    {
        if (password == this.password)
            isPasswordVerified = true;
        else
            isPasswordVerified = false;
    }
    public void CheckAccessPermission()
    {
        if (!isPasswordVerified) throw new AccessDeniedException();
    }
}
```

Listing 4: `BankAccount.cs`

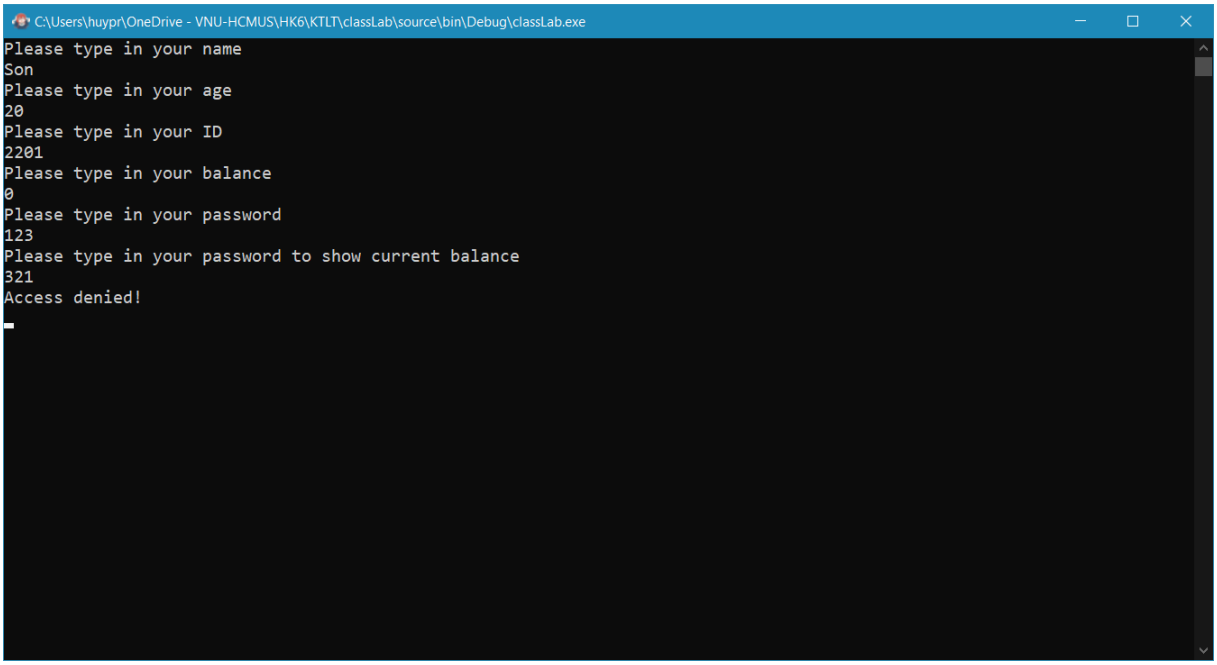
```
public static void GetUserBankAccountInformation(out BankAccount account)
{
    Console.WriteLine("Please type in your name");
    string userName = Console.ReadLine();
    Console.WriteLine("Please type in your age");
    int userAge = int.Parse(Console.ReadLine());
    Console.WriteLine("Please type in your ID");
    uint userID = uint.Parse(Console.ReadLine());
    Console.WriteLine("Please type in your balance");
    double balance = double.Parse(Console.ReadLine());
    Console.WriteLine("Please type in your password");
    string password = Console.ReadLine();
    account = new BankAccount(userName, userAge, userID, balance, password);
    //var account = new BankAccount(string userName, int userAge, uint userID, double
        Balance, string Password);
}
```

Listing 5: Phương thức `GetUserBankAccountInformation`

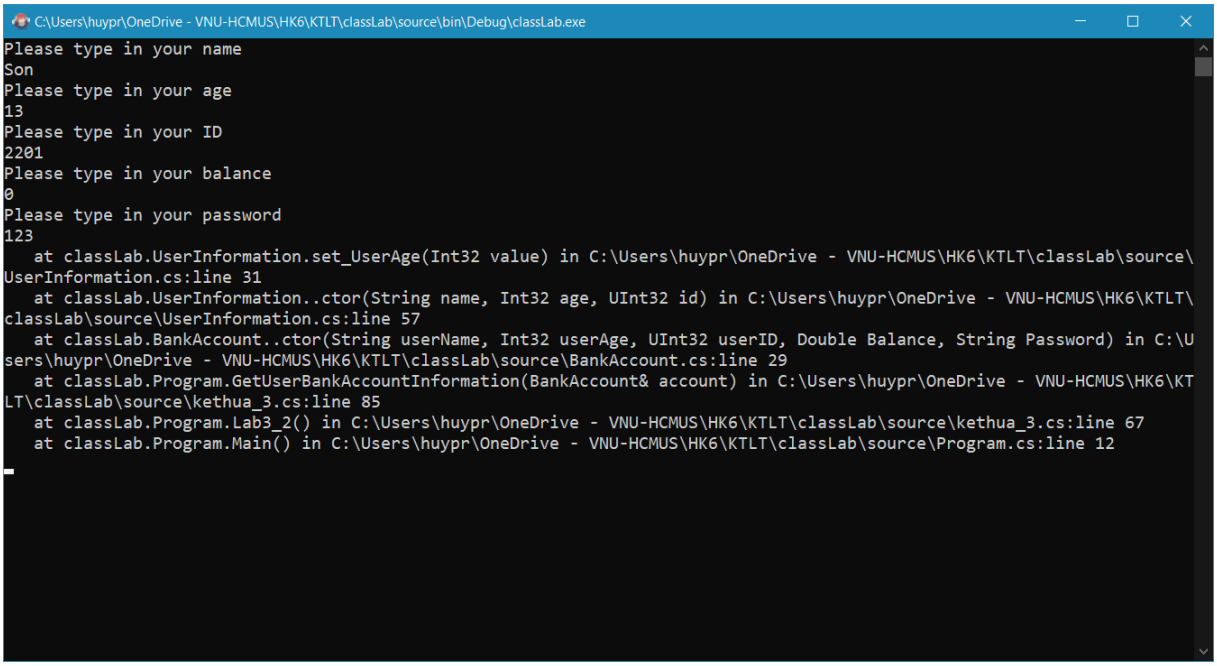
```
internal class AccessDeniedException : ApplicationException
{
    public AccessDeniedException() { }
    public AccessDeniedException(string message) : base(message) { }

}
```

Listing 6: `AccessDeniedException.cs`



Hình 2: Thông báo lỗi khi nhập sai mật khẩu



Hình 3: Thông báo lỗi khi nhập tuổi dưới 18

Source code: Github