

BỘ GIÁO DỤC VÀ ĐÀO TẠO
ĐẠI HỌC CÔNG NGHỆ TP.HCM



HUTECH
Đại học Công nghệ Tp.HCM



THỰC HÀNH LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS

Biên soạn: ThS. Nguyễn Đình Ánh
ThS. Dương Thành Phết
ThS. Trịnh Đồng Thạch Trúc
ThS. Nguyễn Huy Cường

THỰC HÀNH LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS

Ấn bản 2023

MỤC LỤC

MỤC LỤC	I
HƯỚNG DẪN.....	II
BÀI 1: ỨNG DỤNG CONSOLE NGÔN NGỮ C#	1
1.1 MỤC TIÊU	1
1.2 BÀI TẬP.....	2
BÀI 2: LẬP TRÌNH WINDOWS FORM VỚI CONTROLS CƠ BẢN.....	10
2.1 MỤC TIÊU	10
2.2 HƯỚNG DẪN LÀM QUEN VỚI WINDOWS FORM.....	11
2.3 BÀI TẬP.....	17
BÀI 3: LẬP TRÌNH WINDOWS FORM VỚI GIAO DIỆN MDI.....	26
3.1 MỤC TIÊU	26
3.2 BÀI TẬP.....	26
BÀI 4: LẬP TRÌNH VỚI CƠ SỞ DỮ LIỆU SỬ DỤNG ENTITY FRAMEWORK.....	35
4.1 MỤC TIÊU	35
4.2 BÀI TẬP.....	35
BÀI 5: TỔ CHỨC DỰ ÁN VỚI ENTITY FRAMEWORK	51
5.1 MỤC TIÊU	51
5.2 BÀI TẬP.....	51
BÀI 6: ÔN TẬP VÀ KIỂM TRA	62

HƯỚNG DẪN

MÔ TẢ MÔN HỌC

Lập trình C# trên Windows là một trong những môn học nhằm cung cấp kiến thức cơ bản cho những ai muốn xây dựng được ứng dụng trên môi trường trên hệ điều hành Windows. Môn học này cung cấp những kiến thức cơ bản, tổng quan về công nghệ mới của Microsoft: .Net Framework, lập trình hướng đối tượng trên .Net, cũng như giới thiệu bộ thư viện .Net mà cung cấp sẵn (LINQ, Entity framework...), sinh viên có thể tạo giao diện ứng dụng có kết nối các hệ cơ sở dữ liệu (SQL Server). Từ đó sinh viên có thể bước đầu có thể xây dựng những ứng dụng tin học hóa, hỗ trợ các doanh nghiệp trong việc quản lý hàng hóa, nhân sự, mua bán, v.v.

NỘI DUNG THỰC HÀNH

- Bài 1. Tổng quan ngôn ngữ C# trên ứng dụng Console: Bài này cung cấp cho học viên làm quen với ngôn ngữ C#. Với phương pháp lập trình hướng đối tượng, sử dụng thư viện LINQ để giúp truy vấn nhanh chóng.
- Bài 2. Lập trình với ứng dụng giao diện Windows Form với các control cơ bản. Học viên được thiết kế giao diện, làm quen với các controls như: TextBox, Button, ComboBox, RadioButton, DataGridView...Bên cạnh đó việc viết code cho các event trên giao diện người dùng.
- Bài 3. Lập trình với giao diện MDI: Giúp học viên tìm hiểu thêm 1 số controls nâng cao trong Windows Form như MenuStrip, ToolStrip, Timer...
- Bài 4. Sử dụng EntityFrameWork để kết nối với cơ sở dữ liệu SQL Server: Học viên được hướng dẫn cách sử dụng mô hình Code First trên EntityFrameWork với hướng tiếp cận có sẵn cơ sở dữ liệu.
- Bài 5. Tổ chức dự án: Hướng dẫn cách tạo dự án theo mô hình 3 lớp từ đó cung cấp sự phân chia rõ ràng giữa các thành phần trong ứng dụng, tạo điều kiện thuận lợi cho việc bảo trì, mở rộng và quản lý dự án. Nó giúp tăng tính linh hoạt, đồng nhất và tái sử dụng code, cũng như hỗ trợ việc phát triển đa nền tảng.
- Bài 6: Bài kiểm tra kết thúc thực hành giúp đánh giá quá trình học tập của học viên.

KIẾN THỨC TIỀN ĐỀ

Môn học Lập trình C# trên Windows đòi hỏi sinh viên có nền tảng về Lập trình C, và lập trình hướng đối tượng.

YÊU CẦU MÔN HỌC

Người học phải dự học đầy đủ các buổi lên lớp và làm bài tập đầy đủ ở nhà.

CÁCH TIẾP NHẬN NỘI DUNG MÔN HỌC

Để học tốt môn này, người học cần ôn tập các bài đã học, trả lời các câu hỏi và làm đầy đủ bài tập; đọc trước bài mới và tìm thêm các thông tin liên quan đến bài học.

Đối với mỗi bài học, người học đọc trước mục tiêu và tóm tắt bài học, sau đó đọc nội dung bài học. Kết thúc mỗi ý của bài học, người đọc trả lời câu hỏi ôn tập và kết thúc toàn bộ bài học, người đọc làm các bài tập.

PHƯƠNG PHÁP ĐÁNH GIÁ THỰC HÀNH

Môn học được đánh giá:

- Điểm thực hành: Điểm trung bình các bài thực hành hàng tuần - Hình thức tham gia và làm bài tập trên các buổi thực hành trên lớp / ở nhà theo yêu cầu của giáo viên.

BÀI 1: ỨNG DỤNG CONSOLE NGÔN NGỮ C#

1.1 MỤC TIÊU

- Hướng dẫn sinh viên làm quen với ngôn ngữ lập trình C#: qua việc viết các ứng dụng
- Console trong Visual Studio .NET 2022.
- Xây dựng các lớp, tạo đối tượng, truy xuất các phương thức, ...
- Soạn thảo mã nguồn, biên dịch, debug, thực thi chương trình...
- Kế thừa trong lập trình hướng đối tượng trên C#.
- Tìm hiểu về sử dụng thư viện LINQ của .NET
- Khuyến khích sinh viên thực hiện đúng chuẩn coding C# (C# Coding Convention)

- Kiểu dữ liệu: `bool`, `decimal`, `double`, `float`, `int`, `string`, `DateTime`, `bool?`, `decimal?`, `double?`, `float?`, `int?`, `long?`, `DateTime?`, `object`, `var`, `dynamic`...
- Chuyển đổi/ép kiểu dữ liệu: `as`, `is`, Thư viện `Convert`
- Vòng lặp: `for`, `foreach`, `while`, `do...while` và Lệnh điều khiển `break`, `continue`, `return`
- Một số phương thức của thư viện **Console**

```
Console.Write("giá trị cần xuất ra màn hình");
```

```
Console.WriteLine(); // Xuất ra màn hình có xuống dòng
```

```
Console.ReadLine(); // Đọc chuỗi từ bàn phím cho đến khi gặp ký tự xuống dòng
```

```
Console.ReadKey(); // Dừng màn hình để xem kết quả.
```

- Coding Convention C#: Đưa ra các quy ước khi coding với ngôn ngữ lập trình C#, với các quy tắc này giúp tiết kiệm thời gian rất trong quá trình phát triển và bảo trì phần mềm.

Ví dụ: Đặt tên class dùng danh từ, đối với phương thức dùng động từ ...

Tên biến, tên phương thức thể hiện được ý nghĩa

Nên comment những đoạn code khó hiểu hoặc có chức năng đặc biệt

1.2 BÀI TẬP

Bài tập 1: Viết chương trình trò chơi đoán số được mô tả như sau:

- Máy tính sẽ phát sinh ngẫu nhiên một số có 3 chữ số từ 100 đến 999
- Người chơi sẽ đoán số này bằng cách nhập vào một số có 3 chữ số.
- Sau mỗi lần đoán, máy tính sẽ phản hồi dựa trên số đã đoán của người chơi:
 - ✓ Dấu '+' được sử dụng để chỉ ra rằng một chữ số trong số đoán của người chơi là chính xác và nằm ở đúng vị trí tương ứng.
 - ✓ Dấu '?' được sử dụng để chỉ ra rằng một chữ số trong số đoán của người chơi là chính xác, nhưng nằm ở một vị trí khác so với vị trí tương ứng trong số mà máy tính đã phát sinh.
 - ✓ Các vị trí còn lại sẽ không có phản hồi nào. Trường hợp có các số đoán có chữ số trùng nhau thì máy tính vẫn phản hồi ở tất cả các vị trí số đã đoán.
- Thông báo chiến thắng/thất bại sau 7 lần đoán tối đa ?

Ví dụ các phản hồi của máy tính: Máy tính đã phát sinh số 123

- Người chơi đoán: 111. Kết quả phản hồi: +??
- Người chơi đoán: 325. Kết quả phản hồi: ?+
- Người chơi đoán: 249. Kết quả phản hồi: ?
- Người chơi đoán: 490. Kết quả phản hồi:
- Người chơi đoán: 212. Kết quả phản hồi: ???
- Người chơi đoán: 212. Kết quả phản hồi: ???
- Người chơi đoán: 123. Kết quả phản hồi: +++

Minh họa:

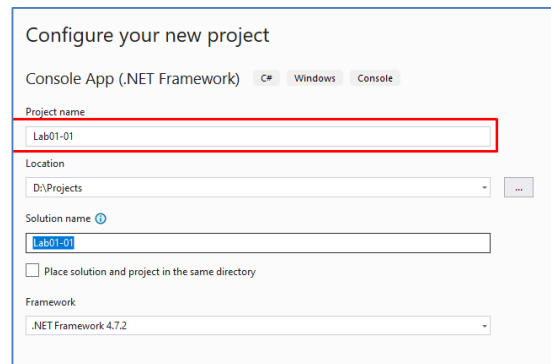
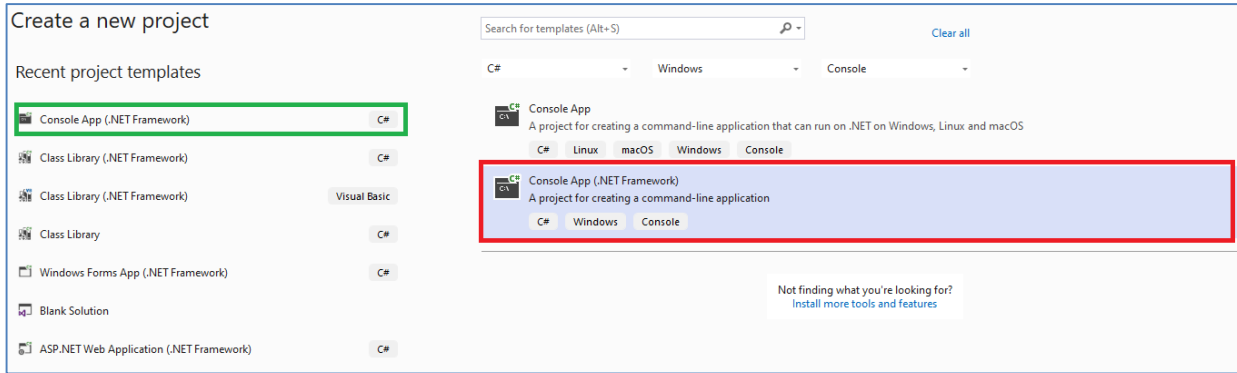
```
=== Chương trình đoán số ===  
Lan đoán thu 1: 123  
Phản hồi từ máy tính: ??  
Lan đoán thu 2: 456  
Phản hồi từ máy tính: ?  
Lan đoán thu 3: 789  
Phản hồi từ máy tính:  
Lan đoán thu 4: 230  
Phản hồi từ máy tính: +  
Lan đoán thu 5: 215  
Phản hồi từ máy tính: +?  
Lan đoán thu 6: 261  
Phản hồi từ máy tính: ++  
Lan đoán thu 7: 266  
Phản hồi từ máy tính: +  
Người chơi đã đoán 7 Lan. Trò chơi kết thúc!  
Người chơi thua cuộc. Số cần đoán là: 241
```

```
=== Chương trình đoán số ===  
Lan đoán thu 1: 123  
Phản hồi từ máy tính: +  
Lan đoán thu 2: 456  
Phản hồi từ máy tính:  
Lan đoán thu 3: 789  
Phản hồi từ máy tính: ?  
Lan đoán thu 4: 107  
Phản hồi từ máy tính: +++  
Người chơi đã đoán 4 Lan. Trò chơi kết thúc!  
Người chơi thắng cuộc!
```

Hướng dẫn:

- Sử dụng một số biến như sau:
targetNumber: Số máy tính đã phát sinh- **targetString**: Biểu diễn chuỗi số từ targetNumber.
attempt: Số lần đoán của người chơi. MAX_GUESS = 7 là số lần đoán tối đa;
guess: Chuỗi số mà người chơi đoán.
feedback: Chuỗi phản hồi từ máy tính dựa trên đoán của người chơi.
- Sử dụng một vòng lặp while khi chưa đoán đúng hoặc hết lượt đoán
Trong vòng lặp, yêu cầu người chơi nhập số đoán và gọi hàm **GetFeedback** để lấy phản hồi từ máy tính dựa trên số đoán và số cần đoán.
- Hàm **GetFeedback** lặp qua từng chữ số **targetString** và so sánh chúng với chữ số tương ứng trong số đoán **guess**. Trả về chuỗi feedback với các ký tự + và ? tương ứng với số đúng vị trí và số sai vị trí.
- Sau khi đoán đúng số hoặc vòng lặp kết thúc và hiển thị thông báo với số lần đoán để kết thúc trò chơi.

Bước 1: Tạo project mới, có template Console App (.NET Framework), ngôn ngữ C# với tên Lab01-01, trong Visual Studio 2022 /2019



Bước 2: Viết chương trình ở trong hàm Main, trong Program.cs

```

static void Main()
{
    Console.WriteLine("=== Chương trình đoán số ===");
    Random random = new Random();
    int targetNumber = random.Next(100, 999);
    string targetString = targetNumber.ToString();

    int attempt = 1, MAX_GUESS = 7;
    string guess, feedback = "";
    while (feedback != "+++" && attempt <= MAX_GUESS)
    {
        Console.Write("Lan đoán thu {0}: ", attempt);
        guess = Console.ReadLine();
        feedback = GetFeedback(targetString, guess);
        Console.WriteLine("Phản hồi từ máy tính: {0}", feedback);
        attempt++;
    }
    Console.WriteLine("Người chơi đã đoán {0} lần. Trò chơi kết thúc!", attempt-1);
    if (feedback != "+++")
        Console.WriteLine("Người chơi thua cuộc. Số cần đoán là: {0}", targetNumber);
    else
        Console.WriteLine("Người chơi thắng cuộc!", attempt);
    Console.ReadLine();
}

1 reference
static string GetFeedback(string target, string guess)
{
    string feedback = "";

    for (int i = 0; i < target.Length; i++)
    {
        if (target[i] == guess[i])
            feedback += "+";
        else if (target.Contains(guess[i].ToString()) )
            feedback += "?";
    }
    return feedback;
}

```

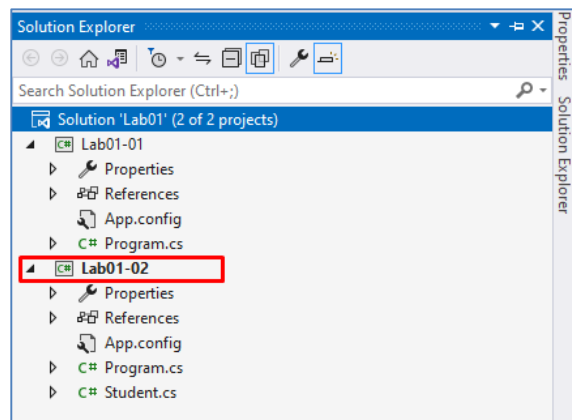
Bài tập 2A: Viết chương trình menu cho phép người dùng quản lý sinh viên. Biết rằng mỗi sinh viên có (mã số, họ tên, khoa và điểm trung bình).

Các chức năng menu:

- 1 – Thêm sinh viên
- 2 – Xuất danh sách sinh viên
- 0 - Thoát

Hướng Dẫn

Bước 1: Thêm project mới Lab01-02 trong cùng solution ở bài tập 1. (Visual Studio cho phép có nhiều projects trong cùng 1 solution. Để Run project thì click phải chọn Set as Startup Project)



Bước 2: Tạo lớp **Student**: Khai báo thuộc tính, phương thức (Right Click vào Lab01-02 Project, chọn Add/ Class đặt tên là Student.cs). Các Property và Constructor nên sử dụng tools để sinh tự động (Quick actions and Refactorings...)

```
class Student
{
    //1. Field
    private string studentID;
    private string fullName;
    private float averageScore;
    private string faculty;

    //2. Property
    public string StudentID { get => studentID; set => studentID = value; }
    public string FullName { get => fullName; set => fullName = value; }
    public float AverageScore { get => averageScore; set => averageScore =
value; }
    public string Faculty { get => faculty; set => faculty = value; }

    //3. Constructor
    public Student()
```

```

    {
    }
    public Student(string studentID, string fullName, float averageScore, string
faculty)
    {
        this.studentID = studentID;
        this.fullName = fullName;
        this.averageScore = averageScore;
        this.faculty = faculty;
    }
    //4. Methods
    public void Input()
    {
        Console.Write("Nhập MSSV:");
        StudentID = Console.ReadLine();
        Console.Write("Nhập Họ tên Sinh viên:");
        FullName = Console.ReadLine();
        Console.Write("Nhập Điểm TB:");
        AverageScore = float.Parse(Console.ReadLine()); //ép sang kiểu float

        Console.Write("Nhập Khoa:");
        Faculty = Console.ReadLine();
    }
    public void Show()
    {
        Console.WriteLine("MSSV:{0} Họ Tên:{1} Khoa:{2} ĐiểmTB:{3}",
this.StudentID, this.fullName, this.Faculty, this.AverageScore);
    }
}

```

Bước 3: Viết code trong hàm **Main()** để tạo menu

```

List<Student> studentList = new List<Student>();

bool exit = false;
while (!exit)
{
    Console.WriteLine("=== MENU ===");
    Console.WriteLine("1. Thêm sinh viên");
    Console.WriteLine("2. Hiển thị danh sách sinh viên");
    Console.WriteLine("0. Thoát");
    Console.Write("Chọn chức năng (0-2): ");

    string choice = Console.ReadLine();

    switch (choice)
    {
        case "1":
            AddStudent(studentList);
            break;
        case "2":
            DisplayStudentList(studentList);
            break;
        case "0":
            exit = true;
            Console.WriteLine("Kết thúc chương trình.");
            break;
        default:
            Console.WriteLine("Tùy chọn không hợp lệ. Vui lòng chọn
lại.");
    }
}

```

```
        break;
    }

    Console.WriteLine();
}
```

Bước 4: Thực hiện các hàm cho case 1 và case 2

```
static void AddStudent(List<Student> studentList)
{
    Console.WriteLine("=== Nhập thông tin sinh viên ===");
    Student student = new Student();
    student.Input();
    studentList.Add(student);
    Console.WriteLine("Thêm sinh viên thành công!");
}
static void DisplayStudentList(List<Student> studentList)
{
    Console.WriteLine("=== Danh sách chi tiết thông tin sinh viên ===");
    foreach (Student student in studentList)
    {
        student.Show();
    }
}
```

Bài tập 2B: Tiếp tục bài tập 2 thực hiện các menu chức năng sau:

- 3 - Xuất ra thông tin của các SV đều thuộc khoa "CNTT"
- 4 - Xuất ra thông tin sinh viên có điểm TB lớn hơn bằng 5.
- 5 - Xuất ra danh sách sinh viên được sắp xếp theo điểm trung bình tăng dần
- 6 - Xuất ra danh sách sinh viên có điểm TB lớn hơn bằng 5 và thuộc khoa "CNTT"
- 7- Xuất ra danh sách sinh viên có điểm trung bình cao nhất và thuộc khoa "CNTT"
- 8- Hãy cho biết số lượng của từng xếp loại trong danh sách? Biết rằng theo thang điểm 10.
Từ 9,0 đến 10,0: Xuất sắc; Từ 8,0 đến cận 9,0: Giỏi; Từ 7,0 đến cận 8,0: Khá;
Từ 5,0 đến cận 7,0: Trung bình; Từ 4,0 đến cận 5,0: Yếu; Dưới 4,0: Kém.

Sử dụng Cú pháp truy vấn (query syntax) hoặc cú pháp phương thức (method syntax) trong LINQ. Danh sách truy vấn trong LINQ

Loại	Phương thức sử dụng
Lọc dữ liệu	Where
Chọn dữ liệu	Select, SelectMany
Phân vùng dữ liệu	Take, Skip, TakeWhile, SkipWhile
Kết hợp	Join, GroupJoin
Sắp Xếp	OrderBy / ThenBy, Reverse
Gom nhóm	GroupBy
Toán tử tập hợp	Distinct, Union, Intersect, Except
Chuyển đổi kiểu dữ liệu	ToSequence, ToArray, ToList, ToDictionary, ToLookup, OfType, Cast
Phần tử	First, FirstOrDefault, Last, LastOrDefault, Single, SingleOrDefault, ElementAt, ElementAtOrDefault, DefaultIfEmpty
Các hàm tính toán kết hợp	Count, LongCount, Sum, Min, Max, Average, Aggregate
Toán tử phát sinh	Range, Repeat, Empty
Toán tử lượng hóa	Any, All, Contains
Trộn	Concat

Bước 1: Thêm các case tương ứng cho bài tập

```

case "3":
    DisplayStudentsByFaculty(studentList, "CNTT");
    break;
case "4":
    DisplayStudentsWithHighAverageScore(studentList, 5);
    break;
case "5":
    SortStudentsByAverageScore(studentList);
    break;
case "6":
    DisplayStudentsByFacultyAndScore(studentList, "CNTT", 5);
    break;
case "7":
    DisplayStudentsWithHighestAverageScoreByFaculty(studentList, "CNTT");
    break;

```

Bước 2: Thực hiện từng case tương ứng

//case 3- DS Sinh viên khoa CNTT

```

static void DisplayStudentsByFaculty(List<Student> studentList, string faculty)
{
    Console.WriteLine("=== Danh sách sinh viên thuộc khoa {0}", faculty);
    var students = studentList.Where(s => s.Faculty.Equals(faculty,
        StringComparison.OrdinalIgnoreCase));
    DisplayStudentList(studentList);
}

//case 4: Xuất ra thông tin sinh viên có điểm TB lớn hơn bằng 5.
static void DisplayStudentsWithHighAverageScore(List<Student> studentList,
float minDTB)
{
    Console.WriteLine("=== Danh sách sinh viên có điểm TB >= {0}", minDTB);
    var students = studentList.Where(s => s.AverageScore >= minDTB);
    DisplayStudentList(studentList);
}

//case 5: Xuất ra danh sách sinh viên được sắp xếp theo điểm trung bình tăng dần

```

```
static void SortStudentsByAverageScore(List<Student> studentList)
{
    Console.WriteLine("=== Danh sách sinh viên được sắp xếp theo điểm trung bình tăng dần ===");
    var sortedStudents = studentList.OrderBy(s => s.AverageScore).ToList();
    DisplayStudentList(sortedStudents);
}

//case 6: DS sinh vien co DTB >=5 va thuoc khoa CNTT
static void DisplayStudentsByFacultyAndScore(List<Student> studentList,
string faculty, float minDTB)
{
    Console.WriteLine("=== Danh sách sinh viên có điểm TB >= {0} và thuộc khoa {1}", minDTB, faculty);
    var students = studentList.Where(s => s.AverageScore >= minDTB
&& s.Faculty.Equals(faculty,
StringComparison.OrdinalIgnoreCase)).ToList();
    DisplayStudentList(students);
}

//case 7,8: SV tự thực hiện
```

Bài tập 3: Tạo thêm 1 project ở trong cùng solution Lab01 có tên là "Lab01-03".

- ✓ Viết lại chương trình từ bài tập trên theo cách tạo thêm một lớp là **Person** làm lớp cơ sở cho lớp **Student**. Chọn Mã số, Họ tên làm field để đưa lên lớp **Person**.
- ✓ Thêm thông tin lớp giảng viên **Teacher** kế thừa từ lớp **Person**. Mỗi giảng viên đều có Mã số, Họ Tên và Địa chỉ.
- ✓ Viết chương trình menu cho phép thực hiện các yêu cầu

1- Thêm sinh viên

2- Thêm giáo viên

3- Xuất danh sách sinh viên

4- Xuất danh sách giáo viên

5- Số lượng từng danh sách (tổng số sinh viên, tổng số giáo viên)

6- Xuất danh sách các Sinh Viên thuộc khoa "**CNTT**".

7- Xuất ra danh sách giáo viên có địa chỉ chứa thông tin "Quận 9"

8- Xuất ra danh sách sinh viên có điểm trung bình cao nhất và thuộc khoa "CNTT"

9- Hãy cho biết số lượng của từng xếp loại trong danh sách? Biết rằng theo thang điểm 10.