**TRƯỜNG ĐẠI HỌC ĐỒNG THÁP**

**KHOA KỸ THUẬT – CÔNG NGHỆ**



**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**LẬP TRÌNH DOTNET C#**

Lớp: ĐHCNTT17 – Chuyên ngành: KHMT

Đề tài:

**QUẢN LÝ QUÁN CÀ PHÊ**

**GVHD: ThS.Trần Kim Hương**

**Thành viên nhóm:**

**0017410663 - Nguyễn Tuấn Kiệt**

**0017410729 - Nguyễn Thị Trúc Lan**

**0017410905 - Lạc Ngọc Khánh**

**Đồng Tháp – ngày 20 tháng 12 năm 2019**

**LỜI NÓI ĐẦU**

Công nghệ thông tin đã ngày càng trở lên phổ biến và hữu dụng trong cuộc sống hiện nay. Đặc biệt đó là ứng dụng phần mềm với nhiều chuơng trình ứng dụng có hiệu quả. Trong đó có các chuơng trình quản lý đã giúp con nguời tổ chức xử lý các công việc trở lên đơn giản, nhanh chóng và chính xác hơn.

Chính vì vậy chúng em đã xây dựng một chương trình quản lý. Đó là: "Quản lý quán cà phê". Chương trình sẽ giúp thực hiện mọi công việc trong một quán cà phê, quản lý các thông tin của quán cũng như việc bán hàng.

Thông qua lần làm bài tập lớn này chúng em muốn nâng cao sự hiểu biết của mình về lĩnh vực Công nghệ thông tin nói chung, cũng như khả năng xây dựng chương trình và kỹ thuật lập trình của mình.

Chúng em rất mong đuợc sự quan tâm, giúp đỡ của các thầy cô. Xin chân thành cảm ơn!

**BẢNG PHÂN CÔNG CÔNG VIỆC (Bài tập nhóm)**

|  |  |
| --- | --- |
| **Họ tên học viên** | **Nội dung công việc thực hiện** |
| 0017410663 - Nguyễn Tuấn Kiệt | Phần mã nguồn, cơ sở dữ liệu |
| 0017410729 - Nguyễn Thị Trúc Lan | Viết báo cáo |
| 0017410905 - Lạc Ngọc Khánh | Viết báo cáo |

**PHẦN MỘT: MỞ ĐẦU**

1. **Đặt vấn đề**

Ngày nay, xã hội ngày càng phát triển nhiều quán thức uống ngày càng mở ra, với cách bày trí không gian đẹp, thoải mái. Khách hàng có thể thưởng thức cà phê trong phòng lạnh hoặc trên sân thượng ngắm nhìn phong cảnh xung quanh và toàn thành phố. Với lượng khách ngày càng gia tăng, để phục khách hàng tốt hơn, chính xác hơn, nhanh hơn, thì chủ quán cần một phần mềm quản lý hàng hóa, đồ uống và công tác kế toán. Phần mềm quản lý quán cà phê phải hỗ trợ việc quản lý các thông tin của quán… quản lý nhân viên trong quán, ca làm việc của nhân viên, doanh thu của quán, quản lý các nguyên liệu và nhà cung cấp nguyên liệu, báo cáo về những món được yêu thích, hóa đơn khách hàng khi thanh toán. Giúp cho việc giải quyết các dữ liệu dễ dàng và hiệu quả.

1. **Ý nghĩa lý thuyết, thực tiễn của đề tài**

* **Ý nghĩa lý thuyết**

Đề tài góp phần xây dựng cơ sở đánh giá hiện trạng, lưu trữ các thông tinn cần thiết trong quá trình quản lý và cho thấy được những ưu và nhược điểm trong phần mềm quản lý quán cà phê, quá trình quản lý, xử lý cần những gì từ đó đưa ra những tính toán số liệu thích hợp.

* **Ý nghĩa thực tiễn**

Với thời đại 4.0 phát triển như hiện nay thì việc quản lý một quán cà phê bằng chương trình phần mềm hỗ trợ sẽ tiện lợi hơn cách thủ công cũ rất nhiều. Vì vậy, kết quả nghiên cứu của đề tài này sẽ tài liệu giúp cho người sử dụng quản lý dễ dàng hơn, nhanh chóng và hiệu quả.

1. **Mục tiêu của đề tài**

Xây dựng chương trình quản lý quán cà phê, quản lý nhân viên trong quán, ca làm việc của nhân viên, doanh thu của quán, quản lí các nguyên liệu và nhà cung cấp nguyên liệu, báo cáo về những món được yêu thích, hóa đơn khách hàng khi thanh toán. Giúp cho việc giải quyết các dữ liệu dễ dàng và hiệu quả.

1. **Phạm vi của đề tài**

Phạm vi nghiên cứu của đề tài là khu vực thành phố Cao Lãnh, song tập trung chủ yếu trung tâm Cao Lãnh, gần trường học, gần khu vui chơi, nơi đông dân cư.

1. **Các phương pháp nghiên cứu**

Các phương pháp nghiên cứu chính được sử dụng trong quá trình thực hiện đề tài này gồm:

* Phương pháp phân tích hệ thống: vận dụng phương pháp này để làm sáng tỏ việc xây dựng chương trình quản lí quán cà phê.
* Phương pháp lập trình C-Sharp.
* Phương pháp cơ sở dữ liệu SQL .
* Phương pháp công nghệ phần mềm.

**PHẦN HAI: NỘI DUNG**

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. **Vài nét về Ngôn ngữ lập trình C#**

C # là một ngôn ngữ lập trình hiện đại được phát triển bởi Microsoft và được phê duyệt bởi European Computer Manufacturers Association (ECMA) và International Standards Organization (ISO).

C # được thiết kế cho các ngôn ngữ chung cơ sở hạ tầng (Common Language Infrastructure – CLI), trong đó bao gồm các mã (Executable Code) và môi trường thực thi (Runtime Environment) cho phép sử dụng các ngôn ngữ cấp cao khác nhau trên đa nền tảng máy tính và kiến trúc khác nhau.

* 1. **Ngôn ngữ ra đời cùng với .NET**
* Kết hợp C++ và Java.
* Hướng đối tượng.
* Hướng thành phần.
* Mạnh mẽ (robust) và bền vững (durable).
* Mọi thứ trong C# đều Object oriented. Kể cả dữ liệu cơ bản.
* Chỉ cho phép đơn kế thừa. Dùng interface để khắc phục.
* Lớp Object là cha của tất cả các lớp. Mọi lớp đều dẫn xuất từ Object.
* Cho phép chia chương trình thành các thành phần nhỏ độc lập nhau.
* Mỗi lớp gói gọn trong một file, không cần file header như C/C++.
* Bổ sung khái niệm namespace để gom nhóm các lớp.
* Bổ sung khái niệm “*property*” cho các lớp.
* Khái niệm delegate & event.
  1. **C# – mạnh mẽ & bền vững**
* Garbage Collector: Tự động thu hồi vùng nhớ không dùng.
* Kiểm soát và xử lý ngoại lệ: Đoạn mã bị lỗi sẽ không được thực thi.
* Type – safe: Không cho gán các kiểu dữ liệu khác nhau.
* Versioning: Đảm bảo sự tương thích giữa lớp con và lớp cha.
  1. **Ưu điểm của khác của C#**
* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

1. **Các kỹ thuật lập trình**
2. **Entity Framework**
3. **Sơ lược về Entity Framework**

Trước .NET 3.5, chúng tôi (nhà phát triển) thường sử dụng để viết mã ADO.NET hoặc Khối truy cập dữ liệu doanh nghiệp để lưu hoặc truy xuất dữ liệu ứng dụng từ cơ sở dữ liệu cơ bản. Chúng tôi đã từng mở một kết nối tới cơ sở dữ liệu, tạo một Bộ dữ liệu để tìm nạp hoặc gửi dữ liệu đến cơ sở dữ liệu, chuyển đổi dữ liệu từ các đối tượng Dataset sang .NET hoặc ngược lại để áp dụng các quy tắc kinh doanh. Đây là một quá trình dễ bị rườm rà và dễ bị lỗi. Microsoft đã cung cấp một khung gọi là "Entity Framework" để tự động hóa tất cả các hoạt động liên quan đến cơ sở dữ liệu này cho ứng dụng của bạn.

Entity Framework là một khung ORM mã nguồn mở cho các ứng dụng .NET được Microsoft hỗ trợ. Nó cho phép các nhà phát triển làm việc với dữ liệu bằng cách sử dụng các đối tượng của các lớp cụ thể của miền mà không cần tập trung vào các bảng và cột cơ sở dữ liệu cơ bản nơi dữ liệu này được lưu trữ. Với Entity Framework, các nhà phát triển có thể làm việc ở mức độ trừu tượng cao hơn khi họ xử lý dữ liệu và có thể tạo và duy trì các ứng dụng hướng dữ liệu với ít mã hơn so với các ứng dụng truyền thống.

Định nghĩa chính thức: Khung Entity Ent là một trình ánh xạ quan hệ đối tượng (O / RM) cho phép các nhà phát triển .NET làm việc với cơ sở dữ liệu bằng các đối tượng .NET. Nó loại bỏ sự cần thiết của hầu hết các mã truy cập dữ liệu mà các nhà phát triển thường cần viết.

1. **Các tính năng của Entity Framework**

* Đa nền tảng: EF Core là một khung đa nền tảng có thể chạy trên Windows, Linux và Mac.
* Mô hình hóa: EF (Entity Framework) tạo EDM (Mô hình dữ liệu thực thể) dựa trên các thực thể POCO (Plain Old CLR Object) với các thuộc tính get / set của các loại dữ liệu khác nhau. Nó sử dụng mô hình này khi truy vấn hoặc lưu dữ liệu thực thể vào cơ sở dữ liệu cơ bản.

- Truy vấn: EF cho phép chúng tôi sử dụng các truy vấn LINQ (C # / VB.NET) để truy xuất dữ liệu từ cơ sở dữ liệu cơ bản. Nhà cung cấp cơ sở dữ liệu sẽ dịch các truy vấn LINQ này sang ngôn ngữ truy vấn dành riêng cho cơ sở dữ liệu (ví dụ: SQL cho cơ sở dữ liệu quan hệ). EF cũng cho phép chúng tôi thực hiện các truy vấn SQL thô trực tiếp đến cơ sở dữ liệu.

- Theo dõi thay đổi: EF theo dõi các thay đổi xảy ra đối với các phiên bản của các thực thể của bạn (Giá trị thuộc tính) cần được gửi đến cơ sở dữ liệu.

- Lưu: EF thực thi các lệnh INSERT, UPDATE và DELETE vào cơ sở dữ liệu dựa trên những thay đổi xảy ra với các thực thể khi bạn gọi phương thức SaveChanges (). EF cung cấp phương thức SaveChangesAsync () không đồng bộ.

- Đồng thời: EF sử dụng Đồng thời lạc quan theo mặc định để bảo vệ các thay đổi ghi đè do người dùng khác thực hiện do dữ liệu được lấy từ cơ sở dữ liệu.

- Giao dịch: EF thực hiện quản lý giao dịch tự động trong khi truy vấn hoặc lưu dữ liệu. Nó cũng cung cấp các tùy chọn để tùy chỉnh quản lý giao dịch.

- Bộ nhớ đệm: EF bao gồm cấp bộ nhớ đệm đầu tiên ra khỏi hộp. Vì vậy, truy vấn lặp đi lặp lại sẽ trả về dữ liệu từ bộ đệm thay vì nhấn cơ sở dữ liệu.

- Các quy ước tích hợp: EF tuân theo các quy ước về mẫu lập trình cấu hình và bao gồm một bộ quy tắc mặc định tự động định cấu hình mô hình EF.

- Cấu hình: EF cho phép chúng tôi định cấu hình mô hình EF bằng cách sử dụng các thuộc tính chú thích dữ liệu hoặc API Fluent để ghi đè các quy ước mặc định.

- Di chuyển: EF cung cấp một tập hợp các lệnh di chuyển có thể được thực thi trên Bảng điều khiển Trình quản lý gói NuGet hoặc Giao diện dòng lệnh để tạo hoặc quản lý Lược đồ cơ sở dữ liệu cơ bản.

1. **DevExpress**

**Giới thiệu về Devexpress**

Đối với những lập trình viên .NET thì DevEpress là một công cụ hết sức hữu dụng, cung cấp rất nhiều control trong Visual Studio. DevExpress không chỉ giúp thiết kế winform hay website đẹp hơn mà còn giúp cho việc lập trình được dễ dàng hơn, ta có thể thấy rõ nhất là trong việc tương tác với cơ sở dữ liệu.

DevExpress được ra mắt lần đầu tiên vào năm 2011 và được đông đảo lập trình viên .NET sử dụng. Từ đó đến này đã trải qua rất nhiều phiên bản với nhiều nâng cấp đáng kể. Phiên bản mới nhấtt hiện nay là v18.2. Với DevExpress ta có thể tự tạo cho mình một bộ Office riêng chỉ trong vòng một vài tiếng

**Thành phần của DevExpress**

* WinForms Controls: Cung cấp các control cho WinForms.
* [ASP.NET](http://asp.net/) Controls: Cung cấp các control cho WebForms.
* WPF Controls: Cung cấp các control cho WPF.
* Silverlight Controls: Cung cấp các control cho Silverlight.
* XtraCharts: Control cung cấp các loại biểu đồ.
* XtraReports: Cung cấp các control tạo báo cáo.
* XPO: Cung cấp môi trường làm việc với database.
* XAF: Một công nghệ mới giúp việc phát triển một cách nhanh chóng.

**Yêu cầu hệ thống cài đặt**

* Microsoft .NET Framework từ 2.0 trở lên
* Microsoft Visual Studio
* SQL Server

**CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH**

1. **Mô tả bài toán**

Khi có khách hàng bước vào quán, nhân viên phục vụ sẽ đến hỏi khách số lượng người đi cùng để có thể sắp xếp chỗ ngồi hợp lí, sau đó đưa khách tới chỗ ngồi đó và mang một số thứ miễn phí như nước lọc, trà đá và thực đơn để khách lựa chọn. Mỗi nhân viên sẽ trực ở một khu vực theo chỉ định của người quản lí. Trong khu vực trực của nhân viên, mỗi bàn sẽ được đánh dấu số thứ tự để nhân viên phục vụ dễ phân biệt và tiện việc quan sát khi khách hàng nào đó có yêu cầu.

Trong quán được chia làm nhiều khu vực khác nhau, mỗi khu vực khu vực sẽ có số lượng bàn nhất định. Nhân viên sau khi ghi hết các yêu cầu của khách thì sẽ chuyển những yêu cầu đó đến chỗ nhân viên pha chế. Đồ uống khi được pha xong sẽ được nhân viên phục vụ mang ra cho khách.

Nhân viên lập hóa đơn để chờ khách hàng thanh toán, khách hàng có thể gọi nhân viên trong khu vực đó để tính tiền, sau đó nhân viên đến quầy thanh toán nhận hóa đơn và đưa khách hàng hoặc khách hàng có thể trực tiếp lại quầy thu ngân thanh toán và nhận hóa đơn.Thông tin của khách hàng sẽ được lưu lại.

Khi thanh toán, nhân viên có thể xuất hóa đơn cho 2 loại khách hàng khách giản lai hoặc khách thành viên. Nếu khách cung cấp thông tin thì sẽ được ghi nhận lại để khuyến mãi về sau thì sẽ xuất hóa đơn khách thành viên.

Sau quá trình tuyển chọn nhân viên làm việc tại quán, người quản lí sẽ mô tả và sắp lịch làm theo ca cho nhân viên. Khi mà nhân viên nhận được thông báo trúng tuyển sẽ được người quản lí phân chia công việc và phân các chức vụ. Nhân viên sẽ đăng kí ca làm việc cho quản lí. Hàng tháng nhân viên sẽ được trả lương và tiền lương đó ứng với số ca mà nhân viên đó làm được trong một tháng.

Nhân viên quản lí lập báo cáo doanh thu để biết tình hình kinh doanh của quán và đưa ra các chiến lược kinh doanh có hiệu quả nhằm nâng cao doanh thu như tăng cường khuyến mãi vào các ngày lễ, giảm giá các sản phẩm trong quán.

Sẽ có một nhân viên đến kho kiểm tra nguyên liệu hằng ngày và lập báo cáo để biết nguyên liệu nào hết mà chuẩn bị đặt hàng để đủ nguyên liệu tạo ra sản phẩm để phục vụ khách hàng vào ngày hôm sau. Mỗi nguyên liệu mà quán cần thì phải báo cáo cho nhân viên quản lí kho lập danh sách các nguyên liệu cần rồi thông báo cho cung cấp.

Khi nguyên liệu được nhà cung cấp chuyển đến sẽ được nhân viên kiểm tra và chuyển hàng đầy đủ vào kho. Và xuất một hóa đơn nhập nguyên liệu.

Khi có khách gọi món thì một sản phẩm sẽ được tạo ra bởi nhân viên, sản phẩm sẽ thuộc các loại sản phẩm khách nhau, được tính bằng đơn vị khác nhau.

Sau mỗi ngày kinh doanh, nhân viên sẽ thống kê báo cáo sản phẩm được ưu chuộng nhất để đưa ra hình thức kinh doanh mới cho quán, báo cáo sản phẩm yêu thích, kèm theo đó là chi tiết báo cáo món ăn yêu thích.

1. **Cách tổ chức dữ liệu**
   1. **Thiết kế dữ liệu vật lý – PDM**

**Bảng Chức vụ: lưu thông tin chức vụ của nhân viên**

* Bảng lưu trữ dữ liệu

|  |  |
| --- | --- |
| **MACHUCVU** | **TENCHUCVU** |
| Mã chức vụ | Tên chức vụ |
| Nvarchar(10) | Nvarchar(20) |
| Khóa chính |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MACHUCVU | Nvarchar(10) |  | Mã chức vụ |
| 2 | TENCHUCVU | Nvarchar(20) |  | Tên chức vụ |

**Bảng Nhân viên: lưu thông tin nhân viên**

* Bảng lưu trữ dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MANV** | **HOTEN** | **GIOITINH** | **NAMSINH** | **DIACHI** |
| Mã nhân viên | Họ tên | Giới tính | Năm sinh | Địa chỉ |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(4) | Int | Nvarchar(20) |
| Khóa chính |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SDT** | **EMAIL** | **NGAYVAOLAM** | **MATKHAU** | **TINHTRANG** |
| Số điện thoại | Email | Ngày vào làm | Mật khẩu | Tình trạng |
| Nvarchar(11) | Nvarchar | DateTime | Nvarchar(10) | Nvarchar(20) |
| **MACHUCV** |
| Mã chức vụ |
| Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MANV | Nvarchar(10) |  | Mã nhân viên |
| 2 | HOTEN | Nvarchar(20) |  | Họ tên |
| 3 | GIOITINH | Nvarchar(4) |  | Giới tính |
| 4 | NAMSINH | Int |  | Năm sinh |
| 5 | DIACHI | Nvarchar(20) |  | Địa chỉ |
| 6 | SDT | Nchar(11) |  | Số điện thoại |
| 7 | EMAIL | Nvarchar(20) |  | Email |
| 8 | NGAYVAOLAM | DateTime |  | Ngày vào làm |
| 9 | MATKHAU | Nvarchar(10) |  | Mật khẩu |
| 10 | TINHTRANG | Nvarchar(20) |  | Tình trạng |
| 11 | MACV | Nvarchar(10) |  | Mã chức vụ |

**Bảng Chi tiết bàn: lưu thông tin chi tiết các bàn trong quán**

* Bảng lưu trữ dữ liệu

|  |  |
| --- | --- |
| **MAHD** | **MABAN** |
| Mã hóa đơn | Mã bàn |
| Nvarchar(10) | Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MAHD | Nvarchar(10) |  | Mã hóa đơn |
| 2 | MABAN | Nvarchar(20) |  | Mã bàn |

**Bảng Bàn: lưu thông tin các bàn ở trong quán**

* Bảng lưu trữ dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MABAN** | **TENBAN** | **TINHTRANG** | **GHICHU** | **MAKHUVUC** |
| Mã bàn | Tên bàn | Tình trạng | Ghi chú | Ghi chú |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(30) | Nvarchar(30) | Nvarchar(10) |
| Khóa chính |  |  |  |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MABAN | Nvarchar(10) |  | Mã bàn |
| 2 | TENBAN | Nvarchar(20) |  | Tên bàn |
| 4 | GHICHU | Nvarchar(30) |  | Ghi chú |
| 5 | MAKHUVUC | Nvarchar(10) |  | Mã khu vực |

**Bảng Khu vực: lưu thông tin khu vực**

* Bảng lưu trữ dữ liệu

|  |  |  |
| --- | --- | --- |
| **MAKHUVUC** | **TENKHUVUC** | **GHICHU** |
| Mã khu vực | Tên khu vực | Ghi chú |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(30) |
| Khóa chính |  |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MAKHUVUC | Nvarchar(10) |  | Mã khu vực |
| 2 | TENKHUVUC | Nvarchar(20) |  | Tên khu vực |
| 3 | GHICHU | Nvarchar(30) |  | Ghi chú |

**Bảng Hóa đơn: lưu thông tin hóa đơn của quán**

* Bảng lưu trữ dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **MAHOADON** | **NGAYLAP** | **TONGTIEN** | **TINHTRANG THANHTOAN** |
| Mã hóa đơn | Ngày lập | Tổng tiền | Tình trạng thanh toán |
| Nvarchar(10) | DateTime | Money | Nvarchar(20) |
| Khóa chính |  |  |  |

|  |
| --- |
| **MANV** |
| Mã nhân viên |
| Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MAHD | Nvarchar(10) |  | Mã hóa đơn |
| 2 | NGAYLAP | DateTime |  | Ngày lập |
| 3 | TONGTIEN | Money |  | Tổng tiền |
| 4 | TINHTRANG | Nvarchar(20) |  | Tình trạng thanh toán |
| 5 | MANV | Nvarchar(10) |  | Mã nhân viên |
| 6 | MAKH | Nvarchar(10) |  | Mã khách hàng |

**Bảng Chi tiết hóa đơn: lưu thông tin chi tiết hóa đơn**

* Bảng lưu trữ dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MAHOADON** | **MASP** | **SOLUONG** | **GHICHU** | **TINHTRANG** |
| Mã hóa đơn | Mã sản phẩm | Số lượng | Ghi chú | Tình trạng |
| Nvarchar(10) | Nvarchar(10) | Int | Nvarchar(30) | Nvarchar(30) |
| Khóa chính |  |  |  |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MAHD | Nvarchar(10) |  | Mã hóa đơn |
| 2 | MASP | Nvarchar(10) |  | Mã sản phẩm |
| 3 | SOLUONG | Int |  | Số lượng |
| 4 | GHICHU | Nvarchar(30) |  | Ghi chú |
| 5 | TINHTRANG | Nvarchar(30) |  | Tình trạng |

**Bảng Khách hàng: lưu thông tin khách hàng**

* Bảng lưu trữ dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **MAKH** | **TENKH** | **SDT** | **TINHTRANG** |
| Mã khách hàng | Tên khách hàng | Số điện thoại | Tình trạng |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(11) | Nvarchar(30) |
| Khóa chính |  |  |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MAKH | Nvarchar(10) |  | Mã khách hàng |
| 2 | TENKH | Nvarchar(20) |  | Tên khách hàng |
| 3 | SDT | Nvarchar(11) |  | Số điện thoại |
| 4 | TINHTRANG | Nvarchar(30) |  | Tình trạng |

**Bảng Sản phẩm: lưu thông tin sản phẩm**

* Bảng lưu trữ dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| **MASP** | **TENSP** | **HINHANH** | **DONGIA** |
| Mã sản phẩm | Tên sản phẩm | Hình ảnh | Đơn giá |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(20) | Money |
| Khóa chính |  |  |  |

|  |  |
| --- | --- |
| **MADONVI** | **MALOAISP** |
| Mã đơn vị | Mã loại sản phẩm |
| Nvarchar(10) | Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MASP | Nvarchar(10) |  | Mã sản phẩm |
| 2 | TENSP | Nvarchar(20) |  | Tên sản phẩm |
| 3 | HINHANH | Nvarchar(20) |  | Hình ảnh |
| 4 | DONGIA | Money |  | Đơn giá |
| 5 | MADONVI | Nvarchar(10) |  | Mã đơn vị |
| 6 | MALOAISP | Nvarchar(10) |  | Mã loại sản phẩm |

**Bảng loại sản phẩm : lưu thông tin các laoi sản phẩm**

* Bảng lưu trữ dữ liệu

|  |  |
| --- | --- |
| **MALOAISP** | **TENLOAISP** |
| Mã loại sản phẩm | Tên loại sản phẩm |
| Nvarchar(10) | Nvarchar(20) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MALOAISP | Nvarchar(10) |  | Mã loại sản phẩm |
| 2 | TENLOAISP | Nvarchar(20) |  | Tên loại sản phẩm |

**Bảng nguyên liệu: lưu thông tin nguyên liệu**

* Bảng lưu trữ dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MANL** | **TENNL** | **GIANL** | **SOLUONGTON** | **MADV** |
| Mã nguyên liệu | Tên nguyên liệu | Giá nguyên liệu | Số lượng tồn | Mã dịch vụ |
| Nvarchar(10) | Nvarchar(10) | Float | Int | Nvarchar(10) |
| Khóa chính |  |  |  |  |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MANL | Nvarchar(10) |  | Mã nguyên liệu |
| 2 | TENNL | Nvarchar(10) |  | Tên nguyên liệu |
| 3 | GIA | Money |  | Giá nguyên liệu |
| 4 | SOLUONG | Int |  | Số lượng tồn |
| 5 | MADV | Nvarchar(10) |  | Mã đơn vị |

**Bảng Chi tiết nguyên liệu: lưu thông tin chi tiết nguyên liệu**

* Bảng lưu trữ dữ liệu

|  |  |
| --- | --- |
| **MANCC** | **MANL** |
| Mã nhà cung cấp | Mã nguyên liệu |
| Nvarchar(10) | Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MANCC | Nvarchar(10) |  | Mã nhà cung cấp |
| 2 | MANL | Nvarchar(20) |  | Mã nguyên liệu |

**Bảng Đơn vị tính: lưu thông tin đơn vị sản phẩm, nguyên liệu**

* Bảng lưu trữ dữ liệu

|  |  |
| --- | --- |
| **MADV** | **TENDONVI** |
| Mã đơn vị | Mã nguyên liệu |
| Nvarchar(10) | Nvarchar(10) |

* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MADV | Nvarchar(10) |  | Mã đơn vị |
| 2 | TENDONVI | Nvarchar(10) |  | Tên đơn vị |

**Bảng nhà cung cấp: lưu thông tin nhà cung cấp**

* Bảng lưu trữ dữ liệu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MANCC** | **TENNCC** | **DIACHI** | **SDT** | **EMAIL** |
| Mã nhà cung cấp | Tên nhà cung cấp | Địa chỉ | Số điện thoại | Email |
| Nvarchar(10) | Nvarchar(20) | Nvarchar(20) | Nvarchar(11) | Nvarchar(20) |
| Khóa chính |  |  |  |  |

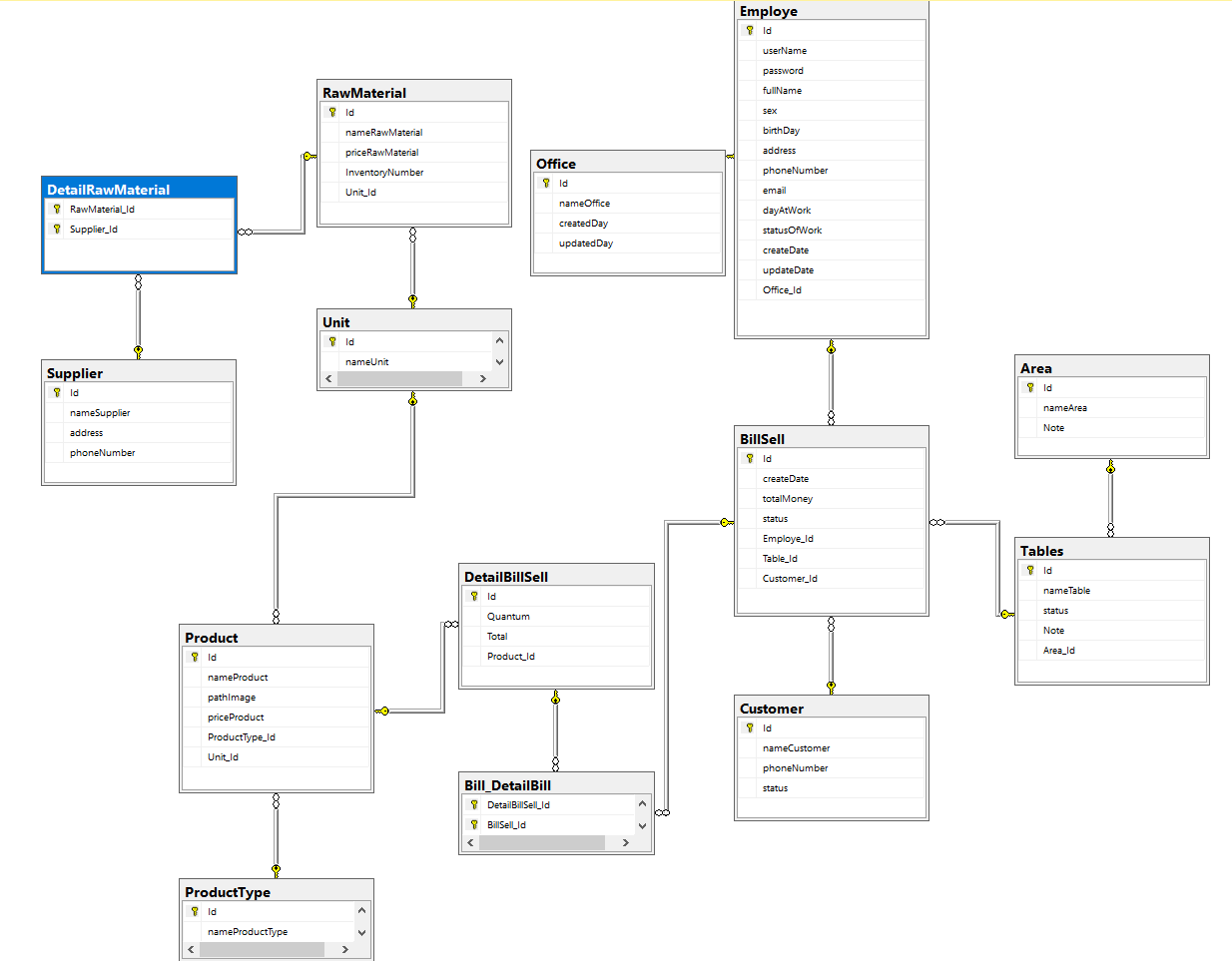
* Bảng cấu trúc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Kiểu dữ liệu** | **Ràng buộc** | **Mô tả** |
| 1 | MANCC | Nvarchar(10) |  | Mã nhà cung cấp |
| 2 | TENNCC | Nvarchar(20) |  | Tên nhà cung cấp |
| 3 | DIACHI | Nvarchar(20) |  | Địa chỉ |
| 4 | SDT | Nvarchar(11) |  | Số điện thoại |
| 5 | EMAIL | Nvarchar(20) |  | Email |

* 1. **Các ràng buộc toàn vẹn trên dữ liệu**

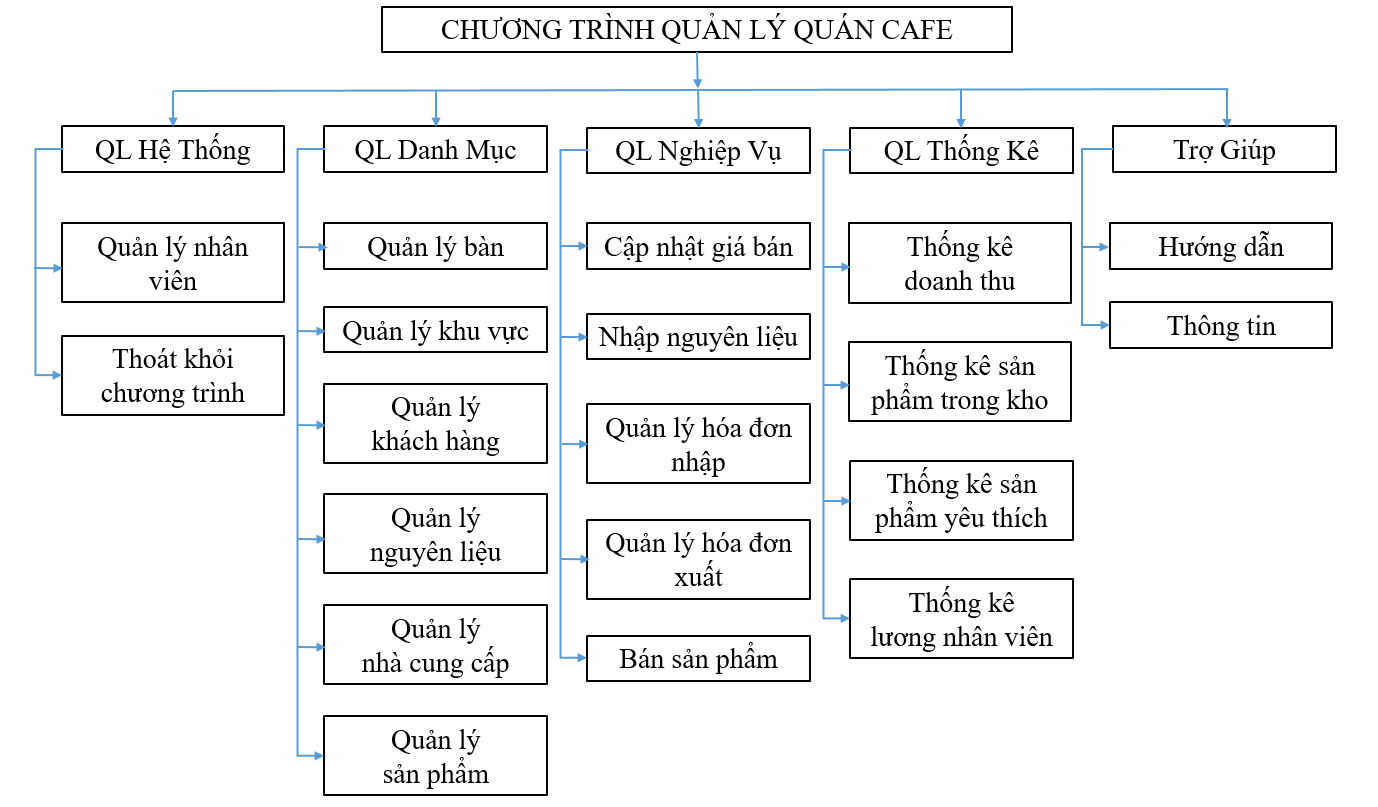
Ràng buộc miền giá trị

* Kiểu dữ liệu Nvarchar (): là chuỗi dài không quá số kí tự trong ngoặc, không chứa các kí tự đặc biệt, không được rỗng.
* Kiểu dữ liệu Int là kiểu số nguyên, không có kí tự đặt biệt, không được rỗng.
* Kiểu dữ liệu datatime định dạng theo kiểu dd/mm/yyyy, năm không được lớn hơn năm hiện tại, không được rỗng.
* Kiểu dữ liệu Float là kiểu số thực, không chứa kí tự đặc biệt, không được rỗng.
* Kiểu dữ liệu Money định dạng theo kiểu tiền tệ.
  1. **Diagram**

****

Hình 2.1 Sơ đồ Diagram

1. **Các giải thuật, module xử lý tiêu biểu**
   1. **Sơ đồ phân rã chức năng hệ thống**

****

Hình 2.2 Sơ đồ phân ra chương trình quản lý quán Café

**Quản lý Hệ Thống:** Bao gồm các chức năng: *Quản lý danh sách nhân viên.* Quản lý các thông tin nhân viên trong quán, với chức năng cập nhật, sửa, xóa, thêm dữ liệu nhân viên như: Tên đăng nhập, mật khẩu, quyền hạn, tên, giới tính, năm sinh, địa chỉ, số điện thoại, email, ngày vào làm, tình trạng. chức vụ; *Thoát khỏi hệ thống.*

**Quản lý Danh Mục:** Bao gồm các chức năng: *Quản lý bàn, khu vực, khách hàng, nguyên liệu, nhà cung cấp, sản phẩm.* Quản lý các thông tin liên quan đến bàn: tên bàn, tình trạng, ghi chú, đến khu vực bàn như: tên khu vực, ghi chú. Loại khách hàng gồm: tên khách hàng, số điện thoại, tình trạng, nguyên liệu bao gồm: tên, giá, số lượng tồn, nhà cung cấp gồm: tên, địa chỉ, số điện thoại, sản phẩm gồm: tên, hình ảnh minh họa, đơn giá. Có thể cập nhật, thêm xóa các thông tin trên.

**Quản lý Nghiệp Vụ:** Bao gồm các chức năng: *Cập nhật giá bán*: Cập nhật, thêm, sửa, xóa giá bán của các loại sản phẩm ứng với loại khách hàng khác nhau. *Nhập nguyên liệu* với các thông tin: mã hóa đơn nhập nguyên liệu, ngày lập, tổng tiền, trạng thái thanh toán, người nhập, mã nguyên liệu, tên nguyên liệu, số lượng, giá nhập; có thể cập nhật, sửa, xóa, thêm các thông tin này. *Quản lý hóa đơn nhập:* Có thể tìm kiếm các thông tin về nhập nguyên liệu theo ngày nhập và đưa ra danh sách, ngoài ra còn có chức năng sửa, xóa hóa đơn nhập đó. *Quản lý bán hàng:* Có thể tìm kiếm, thống kê các thông tin liên quan đến bán sản phẩm theo ngày và có thể sửa, xóa hóa đơn bán sản phẩm. *Bán sản phẩm:* Theo bàn và loại khách. Khi đã chọn xong thông tin bàn và loại khách chương trình đưa ra hóa đơn bán hàng (hóa đơn xuất) để điền tiếp các thông tin và xử lý: mã hóa đơn bán hàng, ngày bán, nhân viên bán, bàn, loại khách, ghi chú, mã sản phẩm, số lượng, giá bán, thành tiền, tiền trả, tiền thừa, tổng tiền; cũng có thể sửa, xóa thông tin khi có sự nhầm lẫn.

**Quản lý Thống Kê:** Bao gồm các chức năng: *Thống kê bán sản phẩm (doanh thu):* Có thể thống kê các sản phẩm được, doanh thu của số sản phẩm đó theo: mã sản phẩm, tên sản phẩm, ngày, tên nhân viên bán, bàn, khu vực. *Thống kê nguyên liệu trong kho:* theo mã nguyên liệu, tên nguyên liệu, số lượng cụ thể; từ các thông tin đó chương trình sẽ đưa ra danh sách các loại nguyên liệu còn trong kho và có thể in được danh sách đó. *Thống kê sản phẩm được yêu thích nhất:* theo tên sản phẩm, ngày bán. *Thống kê lương nhân viên:* theo mã nhân viên, tên nhân viên

**Trợ Giúp:** Bao gồm các chức năng: *Hướng dẫn* cho người sử dụng nắm rõ cách sử dụng chương trình tránh những sai sót không cần thiết. *Trợ giúp* chứ thông tin người hay nhóm người viết chương trình để tiện liên lạc khi có sự cố xảy ra.

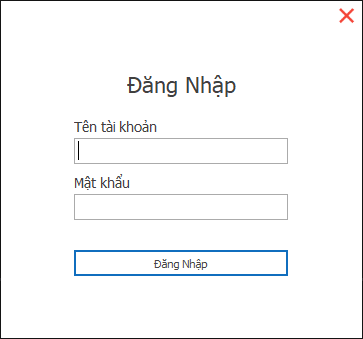
* 1. **Mô tả chi tiết module, chức năng tiêu biểu nhất**

Chương trình gồm có 2 module, module thứ nhất là quản lý thông tin hàng hóa, module thứ hai là hỗ trợ quá trình bán hàng.

* + - Module 1: Giúp cho người quản trị có thể dễ dàng kiểm tra hàng hóa, còn thiếu hay thừa ở trong kho. Có thể quản lý nhân viên, những nhà cung cấp sản phẩm cho cửa hàng.
    - Module 2: Giúp cho người bán hàng dễ dàng trong công tác nghiệp vụ. Tiếp nhận sản phẩm của khách hàng yêu cầu, khi khách yêu cầu tính tiền chỉ cần xuất hóa đơn cho khách. Làm cho công tác tính toán nhanh hơn và chính xác hơn.

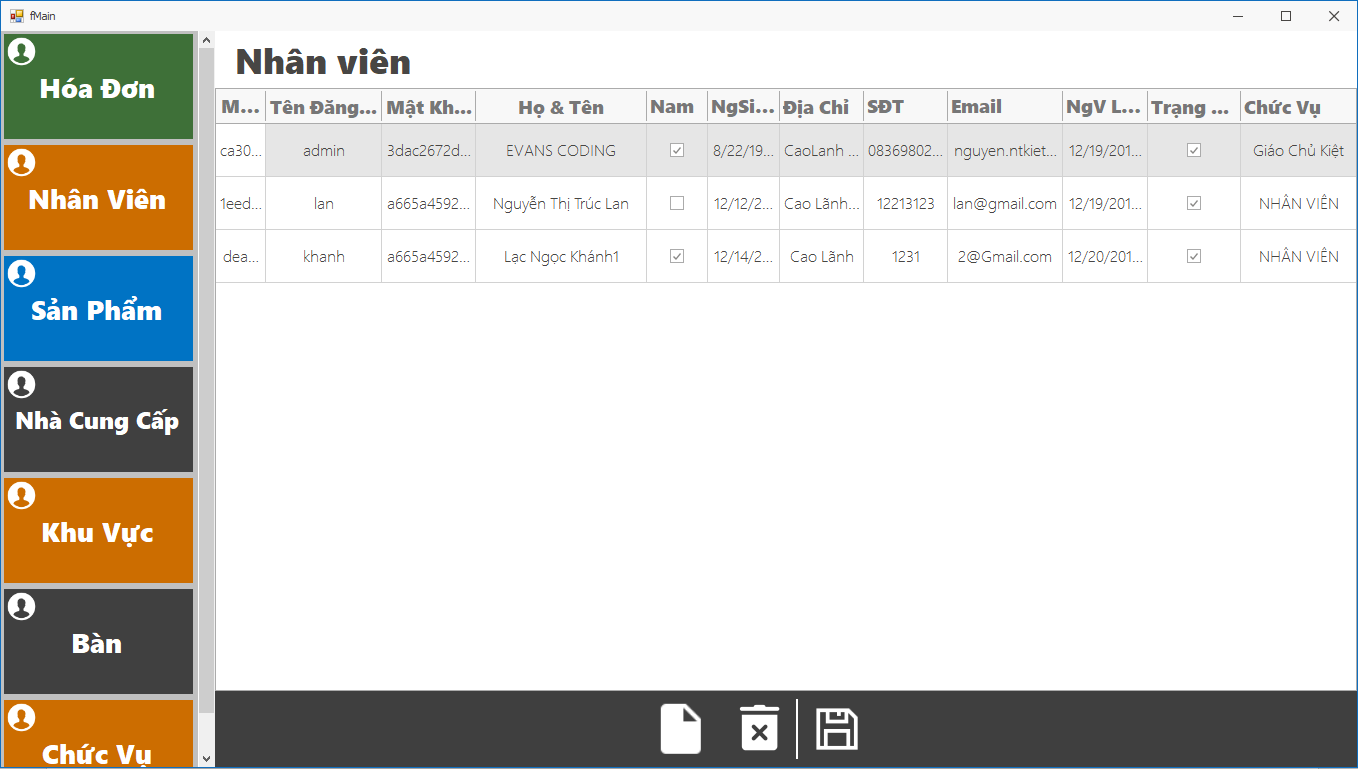
1. **Công cụ sử dụng, xây dựng demo**
   * + Visual studio 2017
     + SQL Sever 2018 Management Studio.
     + Entity Framework
     + DevExpress 19.1.3

1. **Một số giao diện chính của chương trình**
   1. **Giao diện đăng nhập**

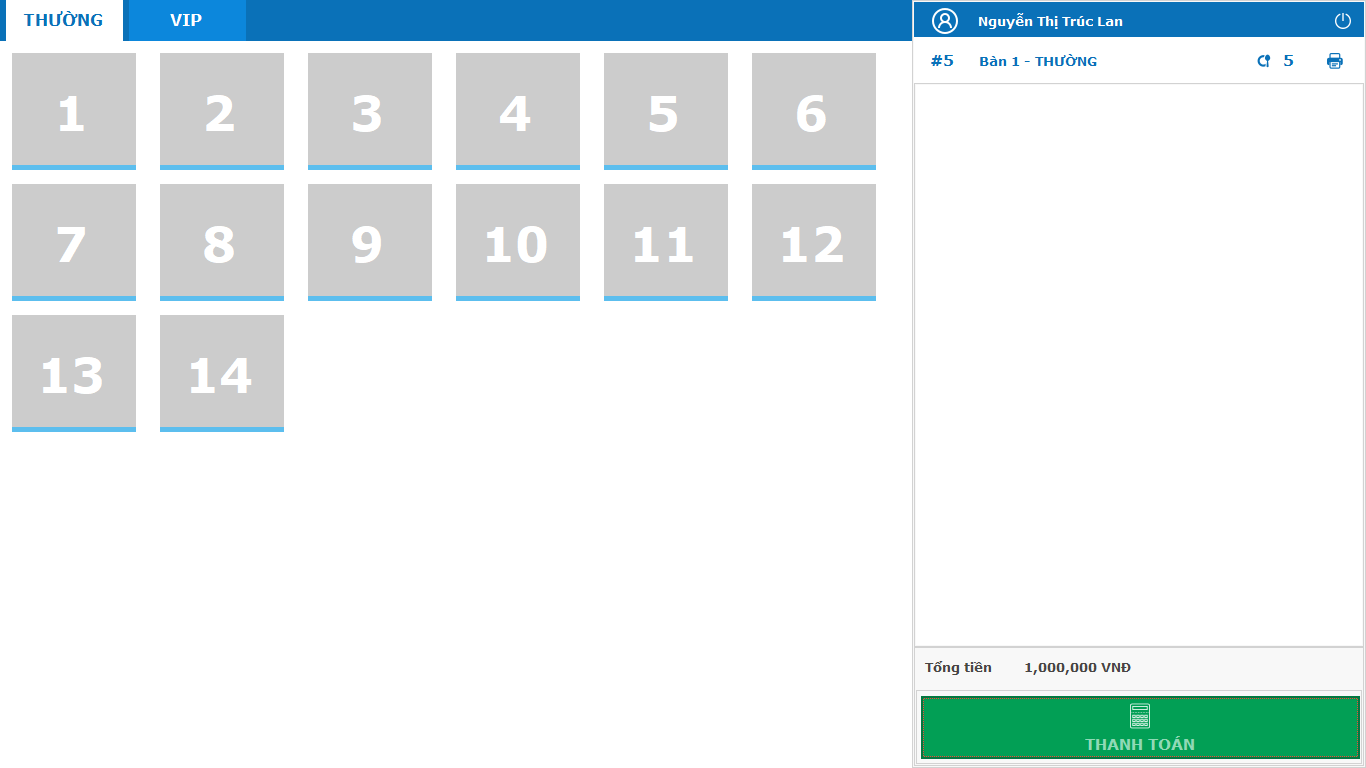


Hình 2.3 Giao diện đăng nhập

* 1. **Giao diện chính**

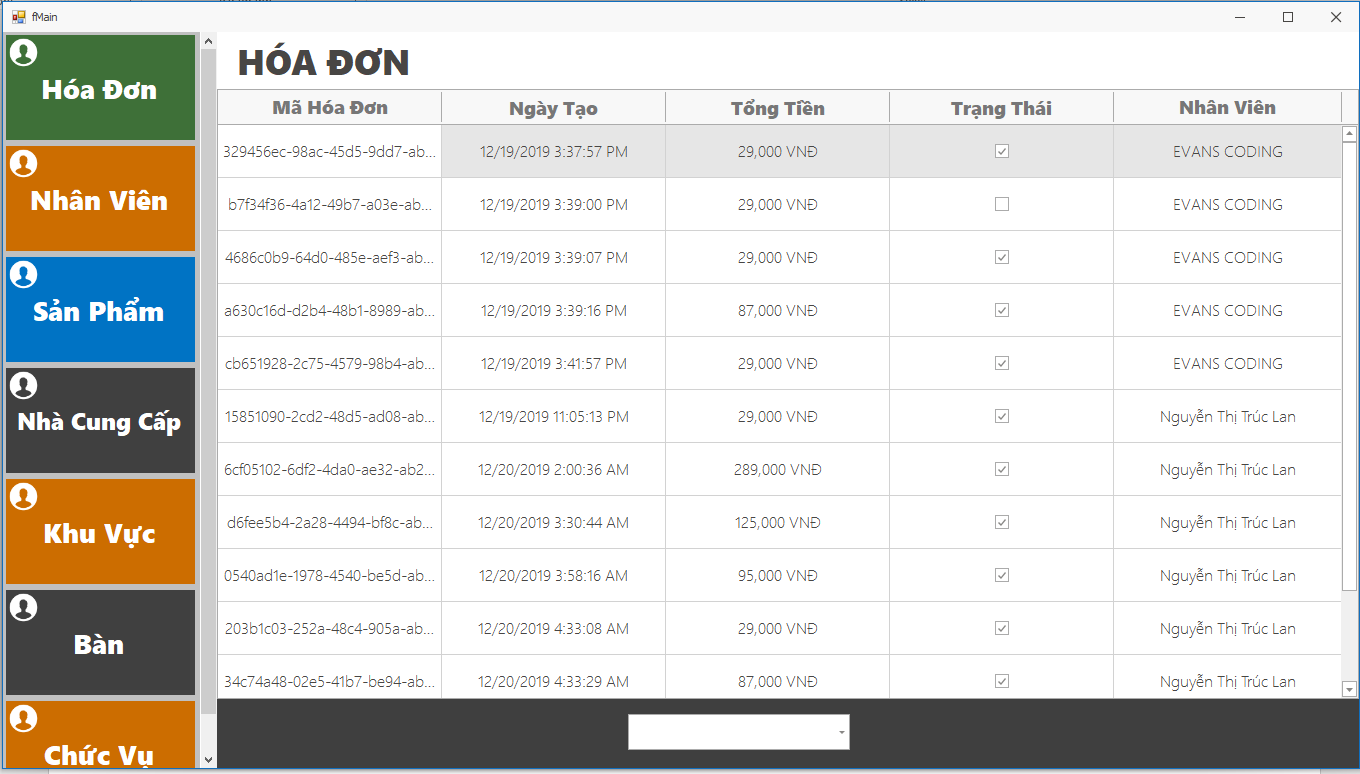


Hình 2.4 Giao diện quản lý



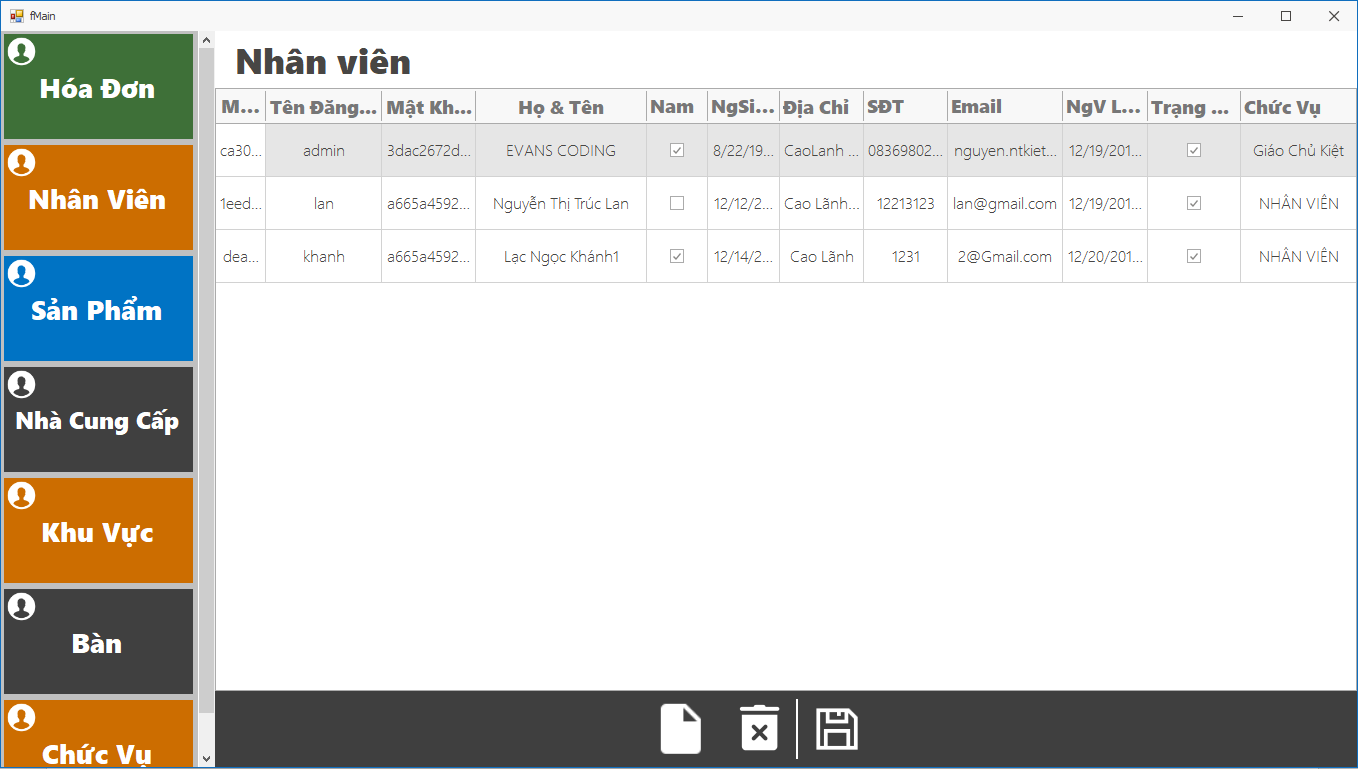
Hình 2.5 Giao diện bàn

* 1. **Giao diện nhập liệu**

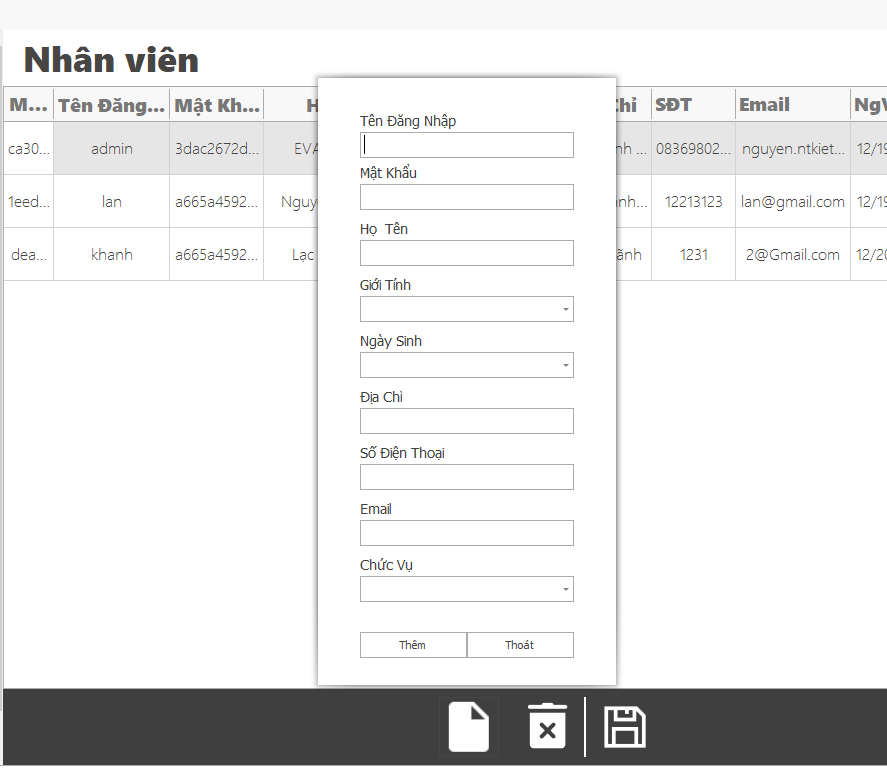
****

Hình 2.6 Giao diện quản lý hóa đơn

* 1. **Các giao diện tiêu biểu khác**



Hình 2.7 Giao diện quản lý nhân viên



Hình 2.8 Giao diện nhập nhân viên mới



Hình 2.9 Hóa đơn khi thanh toán

# BÀI TẬP CÁ NHÂN – SV1: Nguyễn Tuấn Kiệt

**Bài tập chương 1: Lập trình trên Console Application (chọn ra 2 bài tiêu biểu nhất để trình bày)**

**1. Bài tập bảng cửu chương**

- Trình bày giải thuật

class Program

{

static void Main(string[] args)

{

Console.Write("Nhap vao n: ");

Random rb = new Random();

int n = rb.Next(9) + 1;

Console.WriteLine("Nhap vao so cuu chuong: {0}",n);

Console.WriteLine("Ket qua bang cuu chuong {0} la",n);

for (int i = 1; i <= 10; i++)

{

Console.WriteLine("{0} x {1} = {2}", n, i, n \* i);

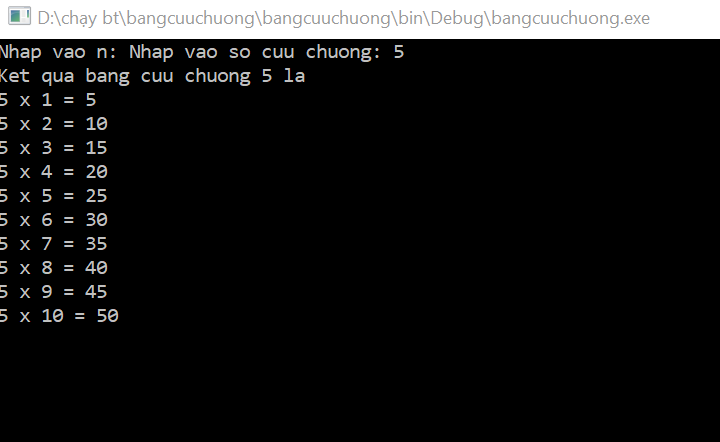
}

Console.ReadKey();

}

}

- Giao diện demo kết quả



**2. Bài tập do - while**

- Trình bày giải thuật

static void Main(string[] args)

{

string str;

do

{

Console.Write("Nhap vao chuoi: ");

str = Console.ReadLine();

if (str.Equals("abc")) break;

} while (true);

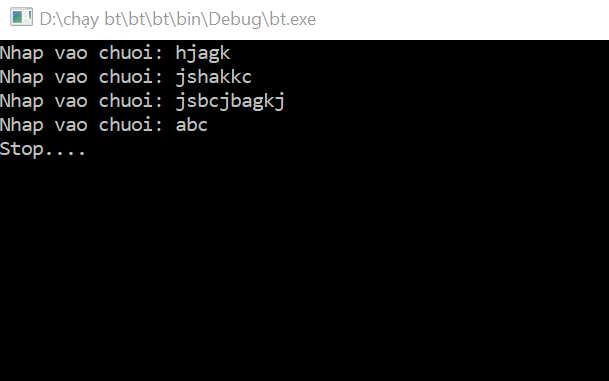
Console.WriteLine("Stop....");

Console.ReadKey();

}

}

- Giao diện demo kết quả



**Bài tập chương 2: Lập trình Hướng đối tượng**

**3. Bài tập kế thừa căng phòng**

- Trình bày giải thuật

public class cCanPhong

{

private float chieuDai;

private float chieuRong;

protected cCanPhong()

{

chieuDai = 0;

chieuRong = 0;

}

protected cCanPhong(float \_chieuDai, float \_chieuRong)

{

chieuDai = \_chieuDai;

chieuRong = \_chieuRong;

}

protected double S()

{

return chieuDai \* chieuRong;

}

protected string toString()

{

return "Chieu dai: " + chieuDai + "| Chieu rong: " + chieuRong + "| Dien tich: " + S();

}

}

public class cChiPhi : cCanPhong

{

private float donGia;

public cChiPhi(float chieuDai, float chieuRong, float \_donGia) :base(chieuDai, chieuRong)

{

donGia = \_donGia;

}

public double tinhChiPhi()

{

return S() \* donGia;

}

public string toString()

{

return base.toString() + "| Chi Phi: " + tinhChiPhi();

}

}

class Program

{

static void Main(string[] args)

{

List<cChiPhi> listChiPhi = new List<cChiPhi>();

string again;

do

{

float chieuDai;

float chieuRong;

float donGia;

Console.WriteLine("CAN PHONG: ");

Console.Write("Nhap chieu dai: ");

chieuDai = float.Parse(Console.ReadLine());

Console.Write("Nhap chieu rong: ");

chieuRong = float.Parse(Console.ReadLine());

Console.Write("Nhap chieu don gia: ");

donGia = float.Parse(Console.ReadLine());

cChiPhi chiPhi = new cChiPhi(chieuDai, chieuRong, donGia);

listChiPhi.Add(chiPhi);

Console.Write("Tiep tuc: ");

again = Console.ReadLine();

} while (again.ToUpper().Equals("YES"));

foreach (cChiPhi item in listChiPhi)

{

Console.WriteLine(item.toString());

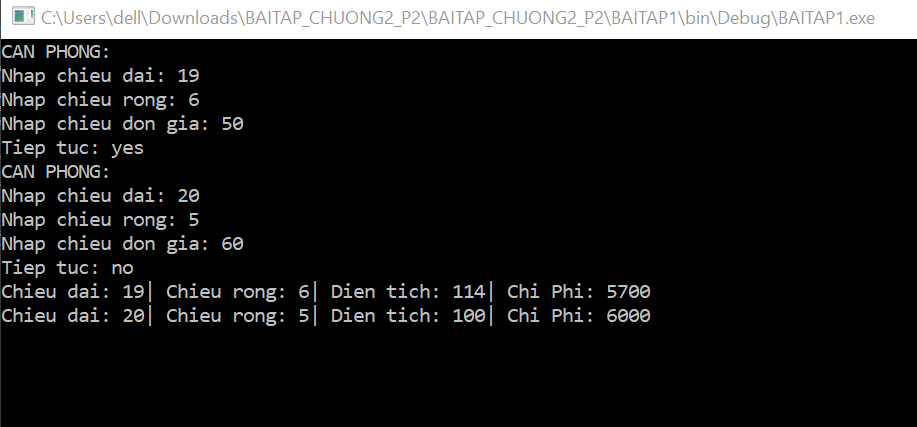
}

Console.ReadKey();

}

}

- Giao diện demo kết quả



**4. Bài tập kế thừa quản lý sinh viên**

- Trình bày giải thuật

public class cSinhVien

{

private string hoTen;

private bool gioiTinh;

private DateTime ngaySinh;

private string chuyenNganh;

private double diemTB;

public string HoTen { get => hoTen; set => hoTen = value; }

protected bool GioiTinh { get => gioiTinh; set => gioiTinh = value; }

protected DateTime NgaySinh { get => ngaySinh; set => ngaySinh = value; }

public string ChuyenNganh { get => chuyenNganh; set => chuyenNganh = value; }

public cSinhVien()

{

HoTen = "";

GioiTinh = true;

NgaySinh = DateTime.Now;

ChuyenNganh = "";

}

protected virtual double dtb()

{

return diemTB;

}

protected string toString()

{

return "Ho ten: " + HoTen + "\nGioi Tinh: " + GioiTinh + "\nNgay sinh: " + NgaySinh + "\nDiem TB: " + dtb();

}

protected void nhapNV()

{

Console.WriteLine("\n sinh vien: ");

Console.Write("Ho ten SV: ");

hoTen = Console.ReadLine();

Console.Write("Gioi tinh: ");

gioiTinh = bool.Parse(Console.ReadLine());

}

}

public class cQuanLySinhVien

{

public void nhapSV()

{

List<cSinhVienCNTT> sinhVienCNTTs = new List<cSinhVienCNTT>();

List<cSinhVienVan> sinhVienVans = new List<cSinhVienVan>();

List<cSinhVienVL> sinhVienVLs = new List<cSinhVienVL>();

int i = 0;

do

{

Console.Write("KHOA: ");

string khoa = Console.ReadLine();

if (khoa.Equals("CNTT"))

{

cSinhVienCNTT cSinhVien = new cSinhVienCNTT();

do

{

cSinhVien.nhapNV();

} while (!String.IsNullOrEmpty(cSinhVien.HoTen));

}else if (khoa.Equals("VAN"))

{

}else if (khoa.Equals("VL"))

{

}

} while (true);

}

}

public class cSinhVienCNTT : cSinhVien

{

private double csharp;

private double pascal;

private double sql;

public double Csharp { get => csharp; set { if (value > 0) csharp = value; else csharp = 0; } }

public double Pascal { get => pascal; set { if (value > 0) pascal = value; else pascal = 0; } }

public double Sql { get => sql; set { if (value > 0) sql = value; else sql = 0; } }

public cSinhVienCNTT()

{

Csharp = 0;

pascal = 0;

sql = 0;

}

protected override double dtb()

{

return (Csharp + pascal + sql) / 3;

}

public void nhapNV()

{

base.nhapNV();

Console.Write("Nhap diem csharp: ");

csharp = double.Parse(Console.ReadLine());

Console.Write("Nhap diem pascal: ");

pascal = double.Parse(Console.ReadLine());

Console.Write("Nhap diem sql: ");

sql = double.Parse(Console.ReadLine());

}

}

public class cSinhVienVan : cSinhVien

{

private double coDien;

private double hienDai;

public double CoDien { get => coDien; set { if (value > 0) coDien = value; else coDien = 0; } }

public double HenDai { get => hienDai; set { if (value > 0) hienDai = value; else hienDai = 0; } }

public cSinhVienVan()

{

coDien = 0;

hienDai = 0;

}

protected override double dtb()

{

return (coDien + hienDai) / 2;

}

}

public class cSinhVienVL : cSinhVien

{

#region Properties

private double co;

private double hatNhan;

private double dien;

private double quang;

public double Co { get => co; set { if (value > 0) co = value; else co = 0; } }

public double HatNhan { get => hatNhan; set { if (value > 0) hatNhan = value; else hatNhan = 0; } }

public double Dien { get => dien; set { if (value > 0) dien = value; else dien = 0; } }

public double Quang { get => quang; set { if (value > 0) quang = value; else quang = 0; } }

#endregion

#region Initialize

public cSinhVienVL()

{

co = 0;

hatNhan = 0;

dien = 0;

quang = 0;

}

public cSinhVienVL(double \_co, double \_hatNhan, double \_dien, double \_quang)

{

this.co = \_co;

this.hatNhan = \_hatNhan;

this.dien = \_dien;

this.quang = \_quang;

}

#endregion

protected override double dtb()

{

return (co + hatNhan + dien + quang)/ 4;

}

}

class Program

{

static void Main(string[] args)

{

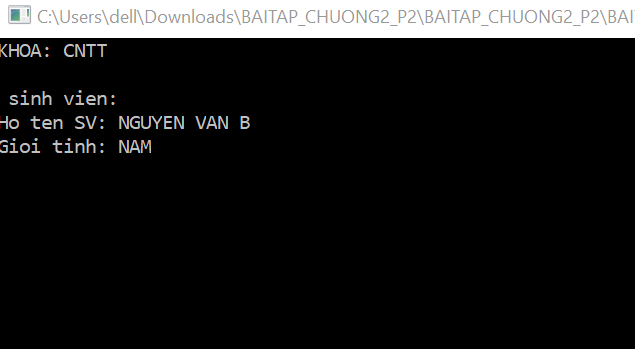
cQuanLySinhVien quanLySinhVien = new cQuanLySinhVien();

quanLySinhVien.nhapSV();

}

}

- Giao diện demo



**BÀI TẬP CÁ NHÂN – SV2: Lạc Ngọc Khánh**

**Bài tập chương 1: Lập trình trên Console Application (chọn ra 2 bài tiêu biểu nhất để trình bày)**

**1. Bài tập while - for**

- Trình bày giải thuật

static void Main(string[] args)

{

int i = 100;

Console.Write("Nhap vao n: ");

Random rb = new Random();

int n = rb.Next(100);

Console.WriteLine(n);

int max = 0;

while (1 <= i)

{

if (i % n == 0)

{

max = i;

break;

}

i--;

}

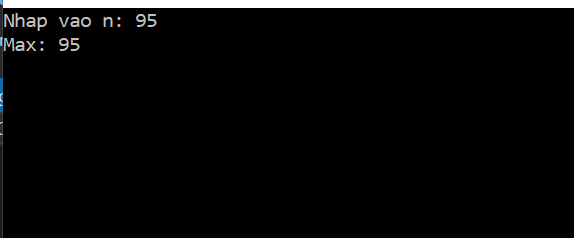
Console.WriteLine("Max: {0}", max);

Console.ReadKey();

}

}

- Giao diện demo kết quả



**2. Bài tập mảng max – min**

- Trình bày giải thuật

class Program

{

static void MaxMin()

{

int n, i, Max, Min;

int[] A = new int[20];

Console.Write("Nhap so phan tu trong mang: ");

n = int.Parse(Console.ReadLine());

// Nhập mảng

for (i = 0; i < n; i++)

{

Console.Write("A[{0}] = ", i);

A[i] = int.Parse(Console.ReadLine());

}

// Tìm Max

Max = A[i];

for (i = 0; i < n; i++)

{

if (Max < A[i])

Max = A[i];

}

Console.WriteLine("Max la:{0} ", Max);

//Tìm Min

Min = A[0];

for (i = 0; i < n; i++)

{

if (Min > A[i])

Min = A[i];

}

Console.WriteLine("Min la:{0} ", Min);

}

static void Main(string[] args)

{

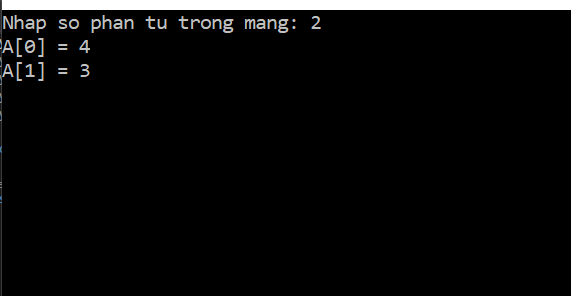
MaxMin();

}

}

}

- Giao diện demo kết quả



**BÀI TẬP CÁ NHÂN – SV3: Nguyễn Thị Trúc Lan**

**Bài tập chương 1: Lập trình trên Console Application (chọn ra 2 bài tiêu biểu nhất để trình bày)**

**1. Bài tập lệnh For – tam giác sao**

- Trình bày giải thuật

static void Main(string[] args)

{

int i, j;

for (i = 1; i <= 5; i++)

{

for (j = 1; j <= i; j++)

Console.Write("\*");

Console.Write(i);

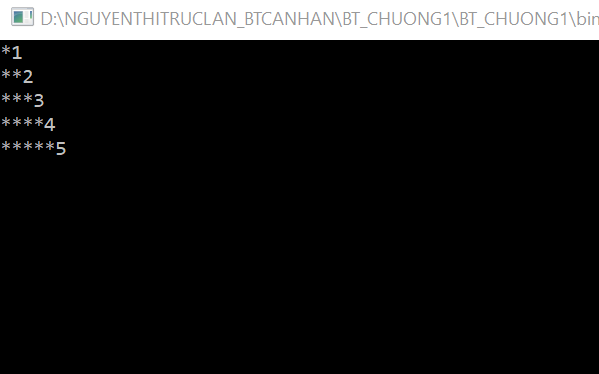
Console.Write("\n");

}

Console.ReadKey();

}

- Giao diện demo kết quả



**2. Bài tập tập nhập- xuất mảng**

- Trình bày giải thuật

static void Mangnhaptubanphim()

{

// Nhập mảng từ bàn phím

int n, i;

int[] A = new int[20];

Console.Write("nhap so phan tu trong mang: ");

n = int.Parse(Console.ReadLine());

//Nhập mảng

Console.WriteLine("Nhap Mang.......");

for (i = 0; i < n; i++)

{

Console.Write("A[{0}] = ", i);

A[i] = int.Parse(Console.ReadLine());

}

//Xuất Mảng

Console.WriteLine("Mang da nhap......");

for (i = 0; i < n; i++)

{

Console.WriteLine("A[{0}] = {1}", i, A[i]);

}

}

static void Main(string[] args)

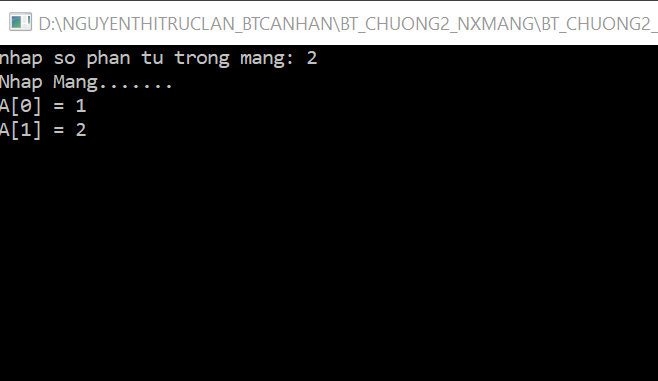
{

Mangnhaptubanphim();

}

}

- Giao diện demo kết quả



**Bài tập chương 2: Lập trình Hướng đối tượng (chọn ra 2 bài tiêu biểu nhất để trình bày)**

**3. Bài tập hinh tròn**

- Trình bày giải thuật

class Program

{

class HinhTron

{

private float R;

private const float PI = 3.14f;

public HinhTron()

{

Console.Write("Nhap R: ");

R = float.Parse(Console.ReadLine());

}

public HinhTron(float bk)

{

R = bk;

}

public float S()//Diện tích

{

float s = PI \* R \* R;

return s;

}

public float P()//Chu vi

{

float p = 2 \* PI \* R;

return p;

}

}

static void Main(string[] args)

{

HinhTron hinhtron = new HinhTron();

Console.WriteLine("Dien tich la: " + hinhtron.S());

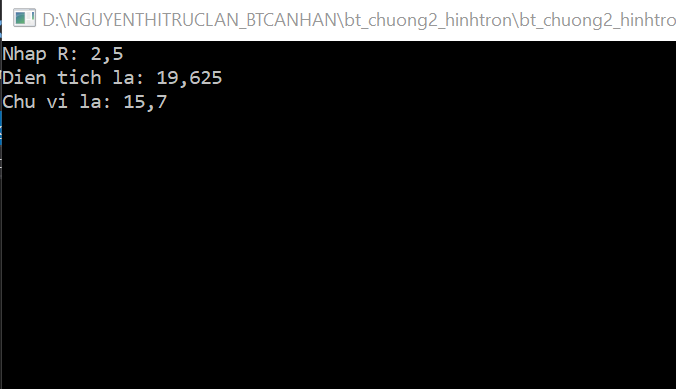
Console.WriteLine("Chu vi la: " + hinhtron.P());

Console.ReadKey();

}

}

- Giao diện demo kết quả



**4. Bài tập về các phép toán cộng trừ nhân chia**

- Trình bày giải thuật

class pheptoan

{

public double so\_a;

public double so\_b;

public void Nhap()

{

Console.Write("Nhap so a: ");

so\_a = double.Parse(Console.ReadLine());

Console.Write("Nhap so b: ");

so\_b = double.Parse(Console.ReadLine());

}

public double PhepCong()

{

return so\_a + so\_b;

}

public double PhepTru()

{

return so\_a - so\_b;

}

public double PhepNhan()

{

return so\_a \* so\_b;

}

public double PhepChiaLayDu()

{

return so\_a % so\_b;

}

public double PhepChiaLayNguyen()

{

return so\_a / so\_b;

}

}

static void Main(string[] args)

{

pheptoan pt = new pheptoan();

pt.Nhap();

Console.WriteLine("Phep cong la: {0}",pt.PhepCong());

Console.WriteLine("Phep tru la: {0}",pt.PhepTru());

Console.WriteLine("Phep nhan la: {0}", pt.PhepNhan());

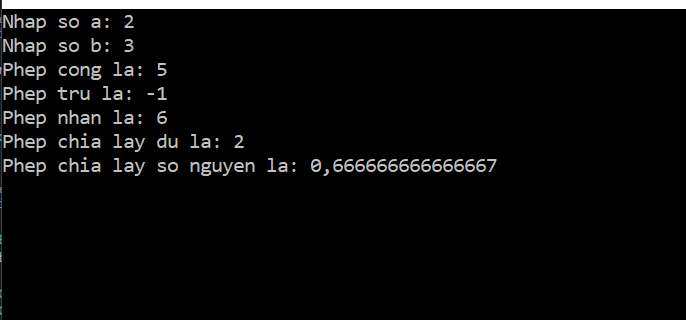
Console.WriteLine("Phep chia lay du la: {0}", pt.PhepChiaLayDu());

Console.WriteLine("Phep chia lay so nguyen la: {0}", pt.PhepChiaLayNguyen());

Console.ReadKey();

}

- Giao diện demo kết quả



**PHẦN BA: KẾT LUẬN**

1. **Kết quả đạt được**
   * + Hoàn thành những giao diện chính của phần mềm quản lý quán Cà Phê.
     + Phần mềm của khả năng quản lý nghiệp vụ ở mức khá. Mỗi module chuyên biệt hóa với nhau.
     + Giao diện thân thiện với người dùng.
     + Tìm hiểu được thêm nhiều kiến thức mới trong quá trình làm bài tập.
     + Tăng khả năng trao đổi thông tin giữa những thành viên trong nhóm.
2. **Hạn chế**
   * + Vì trong quá trình học tập và nghiên cứu nên kiến thức chưa đủ để phát triển một chương trình có độ hoàn chỉnh cao. Trong khi hoạt động có thể xảy ra những lỗi về kỹ thuật lập trình.
     + Cách tổ chức mã nguồn không ổn định, gây khó khăn cho thành viên nhóm trong quá trình tìm hiểu.
3. **Hướng phát triển**
   * + Phát triển hệ thống hoàn chỉnh hơn.
     + Tìm và sửa chữa những lỗi trong hệ thống.
     + Tổ chức mã nguồn dễ hiểu hơn.