TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

**Vấn đề nghiên cứu**

Đề tài "Xây dựng ứng dụng quản lý cửa hàng bán vật tư nông nghiệp" xoay quanh việc số hóa quy trình quản lý, lưu trữ và xử lý thông tin tại các cửa hàng bán vật tư nông nghiệp. Hiện nay, nhiều cửa hàng vẫn sử dụng các phương pháp quản lý thủ công, không chỉ làm tốn thời gian mà còn dễ xảy ra sai sót trong quá trình quản lý, đặc biệt khi lượng thông tin tại cửa hàng ngày càng nhiều. Do đó, việc nghiên cứu và xây dựng một ứng dụng sẽ giúp tự động hóa quy trình quản lý, hỗ trợ chủ cửa hàng trong việc quản lý thông tin sản phẩm, các giao dịch, nhân viên và nhà cung cấp. Ứng dụng giúp tiết kiệm thời gian, công sức và tối ưu hoạt động của cửa hàng bán vật tư nông nghiệp, góp phần nâng cao hiệu quả quản lý và thuận lợi trong quá trình kinh doanh.

**Các hướng tiếp cận và giải quyết vấn đề**

* Phân tích yêu cầu: xác định các yêu cầu của ứng dụng như quản lý nhóm sản phẩm, thông tin từng sản phẩm, thành phần, công dụng, quản lý thông tin nhân viên để xác định các chức năng cần thiết và hướng giải quyết phù hợp.
* Xây dựng mô hình: dựa trên các yêu cầu chức năng xây dựng các lớp, đối tượng, thuộc tính và phướng thức xử lý, đồng thời xác định các mối quan hệ giữa các đối tượng đảm bảo toàn vẹn dữ liệu.
* Lập trình ứng dụng: sử dụng ngôn ngữ C# và .NET Framework với phương pháp lập trình hướng đối tượng để xây dựng ứng dụng gồm các chức năng cơ bản như thêm, sửa, xóa và tìm kiếm trên thông tin sản phẩm. Tạo các menu lựa chọn trực quan để dễ dàng tương tác với các chức năng cửa ứng dụng.
* Kiểm tra và chỉnh sửa ứng dụng: đảm bảo hoạt động tốt, các chức năng hoạt động đúng với yêu cầu.

**Kết quả đạt được**

* Xây dựng thành công ứng dụng quản lý cửa hàng bán vật tư nông nghiệp, đáp ứng đúng với các yêu cầu chức năng đã đặt ra.
* Nắm vững kiến thức về lập trình C#, rèn luyện kỹ năng lập trình hướng đối tượng và cải thiện khả năng phân tích yêu cầu, xây dựng ứng dụng.
* Ứng dụng không chỉ giải quyết các vấn đề nghiên cứu mà còn là nền tảng cho việc phát triển các hệ thống quản lý thực tế trong tương lai.

**MỞ ĐẦU**

**Lý do chọn đề tài**

Ngày nay, các cửa hàng bán vật tư nông nghiệp đóng vai trò quan trọng trong việc cung cấp các sản phẩm thiết yếu cho công việc nông nghiệp của người dân. Tuy nhiên, việc quản lý thông tin tại các cửa hàng này thường gặp nhiều khó khăn do quy trình thủ công, dẫn đến những sai sót trong theo dõi số lượng hàng hóa, lưu trữ, xử lý thông tin. Việc số hóa quy trình này thông qua một ứng dụng quản lý sẽ giúp tối ưu hóa hoạt động kinh doanh, nâng cao hiệu quả và giảm thiểu sai sót. Vì vậy, việc thực hiện đề tài “Xây dựng ứng dụng quản lý cửa hàng bán vật tư nông nghiệp” là cần thiết.

**Mục đích nghiên cứu**

Mục đích chính của nghiên cứu là xây dựng một ứng dụng quản lý cửa hàng bán vật tư nông nghiệp, nhằm giúp các cửa hàng dễ dàng hơn trong việc theo dõi và quản lý thông tin về sản phẩm. Ứng dụng này không chỉ giúp chủ cửa hàng tiết kiệm thời gian mà còn nâng cao độ chính xác trong công tác quản lý. Bên cạnh đó, việc thực hiện đồ án cũng nhằm giúp áp dụng kiến thức lý thuyết vào thực tiễn, rèn luyện kỹ năng lập trình ứng dụng.

**Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là các cửa hàng bán vật tư nông nghiệp. Đề tài sẽ nghiên cứu và xây dựng ứng dụng với các chức năng cần thiết để quản lý thông tin về danh mục sản phẩm, thành phần, công dụng, thương hiệu nhà sản xuất và thông tin nhân viên chịu trách nhiệm từng nhóm sản phẩm.

**Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài giới hạn trong việc xây dựng một ứng dụng quản lý dành cho các cửa hàng bán vật tư nông nghiệp. Ứng dụng sẽ được xây dựng bằng ngôn ngữ lập trình C# và .NET Framework bằng phương pháp lập trình hướng đối tượng với các chức năng cơ bản như thêm, sửa, xóa và tìm kiếm thông tin sản phẩm.

**CHƯƠNG 1: TỔNG QUAN**

**1.1. Giới thiệu về cửa hàng bán vật tư nông nghiệp**

Trong nền kinh tế hiện đại, cửa hàng bán vật tư nông nghiệp đóng một vai trò quan trọng trong việc cung cấp các sản phẩm thiết yếu như phân bón, thuốc bảo vệ thực vật, hạt giống và công cụ nông nghiệp, giúp nông dân cải thiện chất lượng sản phẩm nông nghiệp. Với việc tư vấn kỹ thuật, cung cấp kiến thức và hướng dẫn sử dụng sản phẩm, các cửa hàng vật tư nông nghiệp không chỉ là điểm mua sắm mà còn là nơi người dân học hỏi và trao đổi kinh nghiệm. Điều này giúp người dân áp dụng những phương pháp canh tác hiệu quả hơn, bảo vệ môi trường, giảm thiểu rủi ro và nâng cao hiệu quả kinh tế. Do đó, cửa hàng bán vật tư nông nghiệp trở thành một nhân tố rất quan trọng trong quá trình vận hành và phát triển của ngành nông nghiệp.

**1.2. Sự cần thiết của việc số hóa quản lý cửa hàng**

Mặc dù các cửa hàng bán vật tư nông nghiệp là nhân tố quan trọng trong ngành nông nghiệp nhưng do một số cửa hàng vẫn còn sử dụng hình thức quản lý truyền thống dẫn đến khó khăn trong việc kiểm soát thông tin sản phẩm, giao dịch và nhân viên. Ngày nay, với sự phát triển của công nghệ thông tin đã mở ra nhiều cơ hội cho việc số hóa quy trình quản lý trong ngành nông nghiệp. Việc ứng dụng công nghệ vào quản lý cửa hàng không chỉ giúp tự động hóa các công việc thủ công mà còn tăng cường độ chính xác trong lưu trữ và xử lý thông tin. Lợi ích của việc số hóa quy trình quản lý:

* Tiết kiệm thời gian và công sức: Việc sử dụng ứng dụng quản lý giúp giảm thiểu thời gian cho các công việc như theo dõi tồn kho, ghi chép, quản lý thông tin sản phẩm và nhân viên.
* Cải thiện chất lượng dịch vụ: Ứng dụng cung cấp khả năng lưu trữ và xử lý thông tin chính xác và nhanh chóng từ đó nâng cao trải nghiệm và sự hài lòng về quy trình phụ vụ khách hàng.

**1.3. Các chức năng chính của ứng dụng quản lý**

**1.3.1. Quản lý danh mục các nhóm sản phẩm**

Chức năng này cho phép người dùng tạo, sửa, xóa và cập nhật thông tin về các nhóm sản phẩm trong cửa hàng. Các nhóm sản phẩm giúp phân loại mặt hàng theo đặc điểm chung, dễ dàng quản lý và tìm kiếm.

**1.3.2. Quản lý danh mục các sản phẩm có trong cửa hàng**

Chức năng này giúp người dùng thêm mới, sửa đổi và cập nhật thông tin của từng sản phẩm cụ thể trong cửa hàng, bao gồm tên sản phẩm, giá bán, số lượng tồn kho và nhóm sản phẩm. Điều này hỗ trợ việc kiểm soát hàng hóa và theo dõi tình hình tồn kho của từng sản phẩm.

**1.3.3. Quản lý danh mục thành phần có trong sản phẩm**

Chức năng này cho phép người dùng quản lý các thành phần trong mỗi sản phẩm. Đây là thông tin quan trọng đối với các sản phẩm như phân bón hoặc thuốc bảo vệ thực vật, giúp hiểu rõ thành phần hóa học hoặc sinh học trong sản phẩm để đáp ứng yêu cầu của khách hàng.

**1.3.4. Quản lý danh mục các công dụng của từng sản phẩm**

Chức năng này cho phép lưu trữ và quản lý thông tin về công dụng chính của mỗi sản phẩm. Thông tin công dụng giúp khách hàng chọn lựa sản phẩm phù hợp với nhu cầu.

**1.3.5. Quản lý danh mục các nhân viên của cửa hàng chịu trách nhiệm về một nhóm sản phẩm**

Chức năng này quản lý danh sách nhân viên và phân công trách nhiệm cho từng nhóm sản phẩm như quản lý, tư vấn hoặc hỗ trợ khách hàng đối với các sản phẩm trong nhóm được giao, giúp cửa hàng quản lý công việc hiệu quả hơn.

**1.3.6. Quản lý danh mục thương hiệu (tên) nhà sản xuất sản phẩm**

Chức năng này quản lý thông tin về các thương hiệu nhà sản xuất của sản phẩm trong cửa hàng. Người dùng có thể thêm mới, sửa đổi và cập nhật thông tin nhà sản xuất, giúp phân biệt các sản phẩm theo thương hiệu và cung cấp nguồn gốc rõ ràng cho khách hàng.

**1.4. Kết luận**

Chương này đã trình bày một cách tổng quan về vai trò quan trọng của các cửa hàng bán vật tư nông nghiệp đối với ngành nông nghiệp, đồng thời chương cũng đã chỉ rõ thực trạng quản lý hiện tại, những hạn chế và khó khăn đang gặp phải trong việc quản lý cửa hàng. Từ đó, việc số hóa quy trình quản lý trở nên cần thiết trong thời đại công nghệ phát triển.

Chương cũng đã xác định các chức năng chính mà ứng dụng quản lý cần phải đáp ứng, như quản lý danh mục nhóm sản phẩm, sản phẩm, thành phần, công dụng, thông tin nhân viên và nhà sản xuất để tạo ra một công cụ hỗ trợ quản lý cho cửa hàng.

Những thông tin trong chương này đóng vai trò là cơ sở quan trọng cho việc xây dựng và triển khai ứng dụng quản lý trong các chương tiếp theo.

**Tổng quan về .NET Framework**

1. **Giới thiệu .NET Framework**

.NET Framework là một nền tảng công nghệ do Microsoft phát hành vào năm 2002, cung cấp môi trường để phát triển và chạy các ứng dụng trên hệ điều hành Windows và các dịch vụ web. Với những thư viện phong phú và mạnh mẽ, .NET Framework hỗ trợ hiệu quả cho việc xây dựng các ứng dụng từ đơn giản đến phức tạp, giúp tối ưu hóa quá trình phát triển và nâng cao hiệu suất hoạt động của ứng dụng.

1. **Kiến trúc của .NET Framework**

.NET Framework được tạo ra để giải quyết các mục tiêu sau:

* Cung cấp môi trường phát triển ứng dụng nhất quán
* Giảm thiểu xung đột khi triển khai và quản lý phiên bản phần mềm
* Tăng cường an toàn và bảo mật khi thực thi mã
* Cải thiện hiệu suất so với lập trình theo kịch bản
* Hỗ trợ đa nền tảng và tính nhất quán trong phát triển ứng dụng
* Tương thích và dễ tích hợp với bất kỳ loại mã nào.

.NET Framework gồm hai phần chính là: Common Language Runtime (CLR) và Framework Class Library (FCL).

1. **Common Language Runtime (CLR)**

CLR chịu trách nhiệm về quản lý thực thi mã cung cấp các chức năng sau:

* Quản lý bộ nhớ: cấp phát và giải phóng bộ nhớ cho các đối tượng giúp giảm thiểu lỗi bộ nhớ.
* Thực thi mã: biên dịch mã nguồn thành mã máy để hệ thống hiểu và thực hiện.
* An toàn mã: đảm bảo thực thi mã an toàn toàn thông qua xác minh quyền truy cập.
* Quản lý luồng: đảm bảo chạy hiệu quả các tác vụ đồng thời.
* Quản lý lỗi và ngoại lệ: đảm bảo phát hiện và xử lý các lỗi trong quá trình thực thi.

1. **Framework Class Library (FCL)**

FCL là tập hợp các kiểu dữ liệu có thể tái sử dụng, cung cấp các thành phần để xây dựng, phát triển ứng dụng và dịch vụ web như:

* Ứng dụng Console và GUI Windows: Cung cấp các lớp để xây dựng các ứng dụng Console, Windows Forms, và Windows Presentation Foundation (WPF).
* Các dịch vụ web và ứng dụng server: Hỗ trợ phát triển ứng dụng ASP.NET, các dịch vụ sử dụng Windows Communication Foundation (WCF) và các ứng dụng sử dụng Windows Workflow Foundation (WF).

1. **Các thư viện lớp cơ sở phổ biến trong .NET Framework**
   1. **System**

System bao gồm các lớp cơ bản và lớp nền tảng xác định các kiểu dữ liệu phổ biến, các sự kiện, bộ xử lý sự kiện, giao diện, thuộc tính và xử lý các ngoại lệ.

Một số lớp trong System: Console, Exception, Array, Object, Enum, Int32, Boolean, String, Math, Random, Type,…

* 1. **System.IO**

System.IO chứa các kiểu dữ liệu cho phép đọc và ghi vào các tệp tin và luồng dữ liệu, có các kiểu dữ liệu hỗ trợ cho tệp tin và thư mục.

Một số lớp trong System.IO: File, Stream, Directory, Path,…

* 1. **System.Collection**

System.Collection chứa các giao diện và lớp định nghĩa các tập hợp đối tượng khác nhau.

Một số lớp trong System.Collection: ArrayList, BittArray, Stack, Queue, HashTable, StoredList,…

* 1. **System.Collection.Generic**

System.Collection.Genericchứa các giao diện và lớp định nghĩa các tập hợp tổng quát, giúp tạo ra các tập hợp với kiểu dữ liệu cụ thể.

Một số lớp trong System.Collection.Generic: Comparer<T>, HashSet<T>, List<T>, LinkList<T>, Stack<T>, Queue<T>, SortedList<TKey,TValue>,…

* 1. **System.Text**

System.Text chứa các lớp để đại diện cho mã hóa ký tự ASCII và Unicode, các lớp cơ sở để chuyển đổi giữa các khối ký tự và khối byte, các lớp hỗ trợ thao tác định dạng các chuỗi.

Một số lớp trong System.Text: StringBuider, Encoder, Decoder,…

* 1. **System.Threading**

System.Threading chứa các lớp và giao diện giúp lập trình đa luồng, đồng bộ hóa hoạt động giữa các luồng và truy cập dữ liệu.

Một số lớp trong System.Threading: Monitor, Mutex, ThreadPool, Timer,…

* 1. **System.Security.Cryptography**

System.Security cung cấp các dịch vụ mã hóa và giải mã dữ liệu một cách an toàn, tạo giá trị băm, sinh số ngẫu nhiên và xác thực tin nhắn.

Một số lớp trong System.Security.Cryptography: MD5, SHA256, SHA512,…

**Tổng quan về ngôn ngữ lập trình C#**

1. **Giới thiệu ngôn ngữ lập trình C#**

C# là một ngôn ngữ lập trình thiên về hướng đối tượng do Microsoft phát hành năm 2000. Đây là ngôn ngữ chính và phổ biến nhất của nền tảng .NET. C# không chỉ hỗ trợ lập trình ứng dụng trên máy tính, di động, mà còn cho phép phát triển trang web, trò chơi, thực tế ảo và nhiều ứng dụng khác. Dựa trên phương pháp lập trình hướng đối tượng, C# cung cấp khả năng viết mã an toàn và hiệu quả với hiệu suất cao, đặc biệt là lập trình chức năng. Chính vì những khả năng mà C# cung cấp, ngôn ngữ này đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất thế giới.

1. **Một số đặc điểm cơ bản của C#**

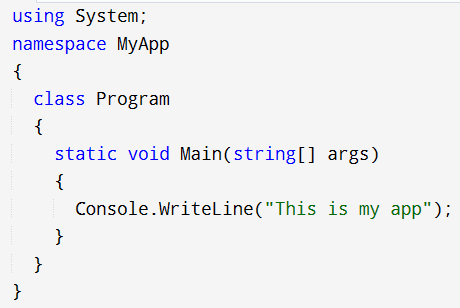
C# dễ học và cung cấp tính năng mở rộng cho lập trình viên. Ứng dụng C# được hỗ trợ quản lý bộ nhớ tự động và có thể sử dụng các thư viện từ .NET SDK, bao gồm cả thư viện nền tảng chung và thư viện chuyên biệt.

C# có cú pháp tương tự C, C++, JavaScript và Java: dùng dấu chấm phẩy để kết thúc câu lệnh, phân biệt chữ hoa chữ thường và hỗ trợ các lệnh điều kiện, vòng lặp cơ bản.

C# là ngôn ngữ có kiểu dữ liệu mạnh, đòi hỏi xác định kiểu cho các biến khi khai báo. Ngoài các kiểu dữ liệu cơ bản như số nguyên, số thực, và chuỗi, người dùng có thể tự định nghĩa kiểu mới bằng struct, class, hoặc interface. C# hỗ trợ tính kế thừa và đa hình, cho phép mở rộng các loại đối tượng dễ dàng.

Để xử lý lỗi, C# sử dụng ngoại lệ, cho phép chương trình xử lý lỗi thông qua các khối try-catch khi gặp sự cố.

1. **Một số khái niệm trong C#**
   1. **Cấu trúc cơ bản của một chương trình C#**

****

Giải thích, trong ảnh trên:

* using.System cho biết chương trình khai báo sử dụng thư viện lớp System của .NET Framework.
* namespace có thể hiểu là không gian làm việc, chứa toàn bộ mã của chương trình. Tên namespace do người lập trình đặt.
* class (lớp) chứa các dữ liệu và phương thức của chương trình, các câu lệnh phải nằm trong class. Tên class do người lập trình đặt.
* Main là phương thức khi chương trình chạy sẽ được gọi đầu tiên, chứa các câu lệnh.
* Console là lớp thuộc thư viện System đã được khai báo. WriteLine() là phương thức của lớp Console thực hiện chức năng xuất văn bản. Chương trình trên sẽ xuất ra chuỗi “This is my app”.
  1. **Biến và hằng dữ liệu**

Biến dùng để lưu trữ giá trị dữ liệu. Cú pháp khai báo biến:

<Kiểu dữ liệu> <Tên biến> = <Giá trị>;

Ví dụ: string name = “Vinh”;

Cú pháp khai báo nhiều biến cùng kiểu dữ liệu:

<Kiểu dữ liệu> <Tên biến 1> = <Giá trị 1>, <Tên biến 2> = <Giá trị 2>;

Ví dụ: int namsinh = 2004, tuoi = 20;

Hằng dùng để lưu trữ giá trị dữ liệu nhưng không thể bị ghi đè. Hằng được khai báo bằng từ khóa const. Cú pháp:

const <Kiểu dữ liệu> <Tên hằng> = <Giá trị>;

Ví dụ: const int maxValue = 100;

* 1. **Kiểu dữ liệu**
  2. **Một số câu lệnh cơ bản**
     1. **Câu lệnh phân nhánh**

Câu lệnh phân nhánh giúp chương trình thực thi các đoạn mã theo logic. C# hỗ trợ một số câu lệnh sau:

* if (Điều kiện) { Câu lệnh } Nếu điều kiện đúng câu lệnh trong cặp ngoặc nhọn sẽ được thực thi.
* if (Điều kiện) { Câu lệnh 1 } else { Câu lệnh 2}. Nếu điều kiện đúng câu lệnh 1 sẽ được thực thi. Ngược lại, câu lệnh 2 sẽ được thực thi.
* if (Điều kiện 1) { Câu lệnh 1 } else if (Điều kiện 2) { Câu lệnh 2} else { Câu lệnh 3 }. Nếu điều kiện 1 đúng câu lệnh 1 sẽ được thực thi. Ngược lại, nếu điều kiện 2 đúng câu lệnh 2 sẽ được thực thi. Ngược lại tất cả các điều kiện trước đó câu lệnh 3 sẽ được thực thi.
* switch (Biểu thức) { case <Giá trị 1>: Câu lệnh 1; break; case <Giá trị 2>: Câu lệnh 2; break; default: Câu lệnh n; break; }. Nếu giá trị ở case nào khớp với biểu thức, câu lệnh thuộc case đó sẽ được thực thi và sẽ dừng lại nếu đến lệnh break (dừng kiểm tra vì đã tìm thấy một giá trị khớp với biểu thức). Nếu không có giá trị nào khớp với biểu thức, câu lệnh thuộc defautl sẽ được thực thi.
  + 1. **Câu lệnh lặp**
    2. **Câu lệnh**
  1. **Xử lý ngoại lệ**

**Tổng quan về lập trình hướng đối tượng trong C#**

1. **Khái niệm lập trình hướng đối tượng**

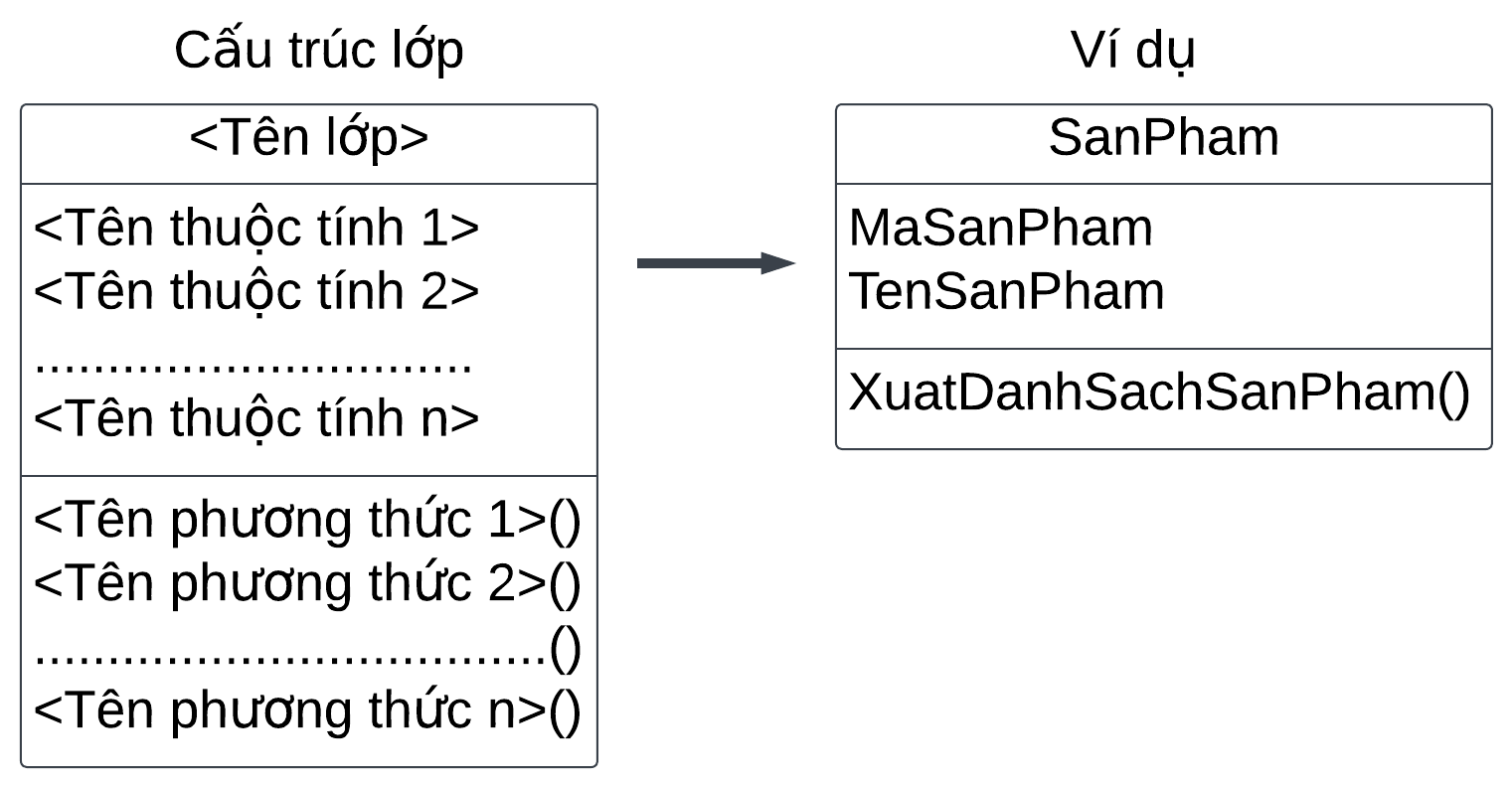
Lập trình hướng đối tượng (OOP – Object Oriented Programing) là một phương pháp lập trình mà cấu trúc của nó được xây dựng dựa trên “đối tượng”. Trong đối tượng có các thuộc tính (properties) dùng để chứa dữ liệu, các đối tượng tương tác với nhau thông qua phương thức (methods) để xử lý một nhiệm vụ cụ thể. Để ngăn chặn truy cập dữ liệu tùy tiện, đảm bảo an toàn dữ liệu bằng việc giới hạn quyền truy cập vào các thuộc tính và phương thức cửa đối tượng ở những mức độ cụ thể.

1. **Một số khái niệm trong lập trình hướng đối tượng**
   1. **Đối tượng**

Đối tượng có thể được xem như là một thực thể trong thế giới thực chứa các thành phần dữ liệu đặc của riêng nó và các hành động để xử lý những dữ liệu đó.

* 1. **Lớp**

Lớp được hiểu như là bản thiết kế hay cấu trúc mẫu cho đối tượng, nó định nghĩa các thuộc tính và phương thức xử lý cho đối tượng. Đối tượng là thể hiện của lớp đã được định nghĩa. Lớp có các thành phần cơ bản là: tên lớp, dữ liệu thành viên (thuộc tính), hàm thành viên (phương thức).



* + 1. Phạm vi truy cập trong lớp

Phạm vi truy cập lớp là cách giới hạn quyền truy cập vào các thuộc tính và phương thức, giúp bảo vệ dữ liệu tránh truy cập và thay đổi tùy ý dữ liệu của lớp. Các phạm vi truy cập phổ biến:

* Public (+): các thuộc tính và phương thức sẽ có thể được truy cập ở bất kỳ đâu trong chương trình.
* Private (-): các thuộc tính và phương thức chỉ có thể truy cập trong phạm vi của lớp.
* Protected (#): các thuộc tính và phương thức được truy cập trong phạm vi của lớp và trong lớp con của lớp đó khi có kế thừa.
  + 1. Thuộc tính trong lớp

Thuộc tính dùng để lưu trữ dữ liệu về đặc trưng hay trạng thái của đối tượng. Cú pháp khai báo thuộc tính như sau:

<Phạm vi truy cập> <Kiểu dữ liệu> <Tên thuộc tính>;

Trong đó:

* + Phạm vi truy cập là public, private hoặc protected.
  + Kiểu dữ liệu có thể là kiểu dữ liệu có sẵn hoặc kiểu dữ liệu tự định nghĩa.
  + Tên thuộc tính là thành phần chứa dữ liệu của lớp. Tên thuộc tính phải tuân theo quy tắc đặt tên của ngôn ngữ lập trình.

Ví dụ: public string name;

private int age;

protected float cost;

* + 1. Phương thức trong lớp

Phương thức dùng để thực hiện các hành động xử lý trên dữ liệu của đối tượng. Cú pháp khai báo phương thức như sau:

<Phạm vi truy cập> <Kiểu trả về> <Tên phương thức>

(<Kiểu dữ liệu> <Tham số>) {

Các câu lệnh xử lý

…

return <dữ liệu trả về>;

};

Trong đó:

* + Phạm vi truy cập là public, private hoặc protected.
  + Kiểu trả về là các kiểu dữ liệu của kết quả mà phương thức trả về, có thể là kiểu dữ liệu có sẵn hoặc kiểu dữ liệu tự định nghĩa. Nếu phương thức không có kiểu trả về, nó sẽ được đặt bằng từ khóa void và cuối phương thức không có từ câu lệnh return.
  + Tên phương thức là thành phần xử lý dữ liệu của lớp. Tên phương thức phải tuân theo quy tắc đặt tên của ngôn ngữ lập trình.
  + Tham số có thể có nhiều hoặc không có tham số nhưng bắt buộc phải được đặt trong cặp dấu ngoặc tròn. Tham số cũng có kiểu dữ liệu như một thuộc tính, cho phép truyền dữ liệu vào một phương thức để xử lý dữ liệu đó.
  + Câu lệnh xử lý phải được đặt trong cặp dấu ngoặc nhọn, các câu lệnh này sẽ xử lý các dữ liệu và thao tác trong phương thức theo nguyên lý do người lập trình viết.

Ví dụ: public void InChuoi() { Console.WriteLine(“Đây là 1 chuỗi”); }

protected int CongSoNguyen(int a, int b) {

int sum = a + b;

return sum;

}

1. **Đặc trưng của lập trình hướng đối tượng**
2. **Lợi ích của lập trình hướng đối tượng**