**Lập trình Ứng dụng Desktop**

**Lê Gia Công**

**Mục lục**

[1 Mở đầu 3](#_Toc524588868)

[1.1 Mô hình Bloom 3](#_Toc524588869)

[1.2 Theo lập trình, nên học gì? 3](#_Toc524588870)

[1.2.1 Về Stack Overflow 3](#_Toc524588871)

[1.2.2 Nên chọn công việc nào để theo học 4](#_Toc524588872)

[1.2.3 Mức lương của mỗi công việc 4](#_Toc524588873)

[1.2.4 Chọn công nghệ nào để theo 5](#_Toc524588874)

[1.3 Theo nghề web, nên học gì? 9](#_Toc524588875)

[2 Lập trình cơ sở dữ liệu 11](#_Toc524588876)

[2.1 Giới thiệu môn học 11](#_Toc524588877)

[2.2 Cài đặt SQL Server, Visual Studio 11](#_Toc524588878)

[2.2.1 Cơ sở dữ liệu là gì? 11](#_Toc524588879)

[2.2.2 Hệ quản trị cơ sở dữ liệu là gì? 13](#_Toc524588880)

[2.2.3 Cài đặt SQL Server 13](#_Toc524588881)

[2.2.4 Cài đặt Visual Studio 15](#_Toc524588882)

[2.3 Làm quen với WinForms và SQL server 16](#_Toc524588883)

[2.3.1 WinForms và WPF 16](#_Toc524588884)

[2.3.2 Lập trình WinForms căn bản 18](#_Toc524588885)

[3 Phần mềm quản lý quán café 21](#_Toc524588886)

[3.1.1 Hướng dẫn kết nối, tạo database, tạo table và tạo liên kết giữa các table trong Sql Server 21](#_Toc524588887)

[3.1.2 Kết nối Visual Studio với SQL server 22](#_Toc524588888)

[4 WPF 22](#_Toc524588889)

# Mở đầu

## Mô hình Bloom

Áp dụng mô hình Bloom[[1]](#footnote-1) vào việc học và làm việc, có thể hiểu nôm na là để tìm hiểu và làm việc trong lĩnh vực lập trình cần trải qua các bước sau, Biết > Hiểu > Vận dụng > Phân tích > Tổng hợp > Đánh giá, xem thêm phần giải thích dưới đây:

|  |  |
| --- | --- |
| Đánh giá (evaluation) | Đánh giá được giải pháp đó là hợp lý hay không |
| Tổng hợp (synthesis) | Gắn kết các thành phần lại để tạo ra một hệ thống mới |
| Phân tích (analysis) | Nhận biết được nguyên lý tổ chức, hiểu được sự gắn kết của các thành phần |
| Vận dụng (application) | Áp dụng vào làm thực tế, áp dụng trong các tình huống khác nhau |
| Hiểu (comprehension) | Hiểu được tại sao lại phải làm như thế |
| Biết (knowledge) | Nhớ và làm theo |

Phương pháp học là,

– Xem các bài hướng dẫn, rồi làm theo. Ghi lại quá trình học (đã coi clip nào, các trang web hay, các tài liệu hay)

– Ghi lại các khái niệm chưa biết, tìm đọc tài liệu, để hiểu các khái niệm này. Ghi chép lại các khái niệm này

– Qua việc làm theo, ghi chép, tìm hiểu, người học sẽ biết được để làm lập trình cần các công cụ gì, sử dụng các công cụ đó như thế nào, bước đầu có những khái niệm ở mức cơ bản (hiểu)

– Tự làm các dự án nhỏ (lưu trên Github, để giới thiệu với nhà tuyển dụng)

– Xin làm thêm để học nghề

## Theo lập trình, nên học gì?

### Về Stack Overflow

– Là một mạng xã hội lớn, uy tín của các lập trình viên trên thế giới

– Giúp các lập trình viên tìm kiếm giải pháp cho các vấn đề gặp phải trong công việc hàng ngày; là nơi mỗi cá nhân thể hiện trình độ của mình thông qua việc giúp đỡ các đồng nghiệp khác; tuyển người làm & tìm việc làm

– Jeff Atwood và Joel Spolsky (người Mỹ) sáng lập Stack Overflow năm 2008

– Hơn 50M lượt người truy cập mỗi tháng, trên 14M câu hỏi, trên 19M trả lời

­– Thực hiện các khảo sát hàng năm, năm nay là lần thứ 8, với hơn 100.000 lập trình viên đang làm việc tại 183 quốc gia và vùng lãnh thổ tham gia khảo sát

­– Nội dung khảo sát liên quan đến ngành lập trình[[2]](#footnote-2),[[3]](#footnote-3),[[4]](#footnote-4),[[5]](#footnote-5),[[6]](#footnote-6)

### Nên chọn công việc nào để theo học

Bảng sau là một số công việc phổ biến của nghề lập trình, đồng thời cho biết tỉ lệ của mỗi công việc dựa trên số người được khảo sát. Tham khảo để chọn sẽ theo học cái gì? Thông thường, nên chọn công việc có nhiều người đang làm, tức thị trường đang cần nhiều. Hoặc có người đang làm nghĩa là có nhu cầu tuyển dụng, vì vậy học cái gì cũng được, nếu thực sự đam mê. Tuy nhiên, nếu chưa thử hết các công việc làm sao biết mình thích cái nào hơn cái nào? Cần người tư vấn và tự trải nghiệm?

|  |  |  |  |
| --- | --- | --- | --- |
| ***Loại công việc*** | ***Tỉ lệ*** | ***Loại công việc*** | ***Tỉ lệ*** |
| Back-end developer | 57.9% | Data or business analyst | 8.2% |
| Full-stack developer | 48.2% | Data scientist or machine learning specialist | 7.7% |
| Front-end developer | 37.8% | QA or test developer | 6.7% |
| Mobile developer | 20.4% | Engineering manager | 5.7% |
| Desktop or enterprise applications developer | 17.2% | Embedded applications or devices developer | 5.2% |
| Student | 17.1% | Game or graphics developer | 5.0% |
| Database administrator | 14.3% | Product manager | 4.7% |
| Designer | 13.1% | Educator or academic researcher | 4.0% |
| System administrator | 11.3% | C-suite executive (CEO, CTO, etc.) | 3.8% |
| DevOps specialist | 10.4% | Marketing or sales professional | 1.2% |

### Mức lương của mỗi công việc

Ngoài yếu tố dễ xin việc, hợp với đam mê của bản thân, cũng nên quan tâm tới mức lương của mỗi công việc. Tất nhiên, lương cao thì đòi hỏi trình độ, kinh nghiệm, áp lực công việc cũng cao.

|  |  |  |  |
| --- | --- | --- | --- |
| ***Loại công việc*** | ***Mức lương*** | ***Loại công việc*** | ***Mức lương*** |
| Engineering manager | $89,000 | System administrator | $56,000 |
| DevOps specialist | $72,000 | QA or test developer | $55,000 |
| Data scientist or machine learning specialist | $60,000 | Database administrator | $51,000 |
| Data or business analyst | $59,000 | Front-end developer | $51,000 |
| Embedded applications or devices developer | $59,000 | Designer | $46,000 |
| Full-stack developer | $59,000 | Educator or academic researcher | $44,000 |
| Desktop or enterprise applications developer | $57,000 | Mobile developer | $43,000 |
| Back-end developer | $56,000 | Game or graphics developer | $40,000 |

### Chọn công nghệ nào để theo

Với mỗi công việc có thể lựa chọn nhiều công nghệ khác nhau, bao gồm: ngôn ngữ lập trình, cơ sở dữ liệu, mã nguồn mở hay đóng, trình viết mã, framework, thư viện.

**Ngôn ngữ**

Bảng dưới đây là danh sách các ngôn ngữ phổ biến đang được mọi người sử dụng,

|  |  |  |  |
| --- | --- | --- | --- |
| ***Ngôn ngữ*** | ***Tỉ lệ*** | ***Ngôn ngữ*** | ***Tỉ lệ*** |
| JavaScript | 71.5% | Swift | 8.3% |
| HTML | 69.4% | Objective-C | 7.3% |
| CSS | 66.2% | Go | 7.2% |
| SQL | 58.5% | Assembly | 6.9% |
| Java | 45.4% | VB.NET | 6.9% |
| Bash/Shell | 40.4% | R | 6.0% |
| Python | 37.9% | Matlab | 5.5% |
| C# | 35.3% | VBA | 4.8% |
| PHP | 31.4% | Kotlin | 4.7% |
| C++ | 24.6% | Groovy | 4.5% |
| C | 22.1% | Scala | 4.5% |
| TypeScript | 18.3% | Perl | 4.2% |
| Ruby | 10.3% |  |  |

**Cơ sở dữ liệu**

Bảng dưới đây là danh sách các cơ sở dữ liệu phổ biến đang được mọi người sử dụng,

|  |  |  |  |
| --- | --- | --- | --- |
| ***Cơ sở dữ liệu*** | ***Tỉ lệ*** | ***Cơ sở dữ liệu*** | ***Tỉ lệ*** |
| MySQL | 58.7% | Memcached | 5.5% |
| SQL Server | 41.2% | Amazon DynamoDB | 5.2% |
| PostgreSQL | 32.9% | Amazon RDS/Aurora | 5.1% |
| MongoDB | 25.9% | Cassandra | 3.7% |
| SQLite | 19.7% | IBM Db2 | 2.5% |
| Redis | 18.0% | Neo4j | 2.4% |
| Elasticsearch | 14.1% | Amazon Redshift | 2.2% |
| MariaDB | 13.4% | Apache Hive | 2.2% |
| Oracle | 11.1% | Google BigQuery | 2.1% |
| Microsoft Azure (Tables, CosmosDB, SQL, etc) | 7.9% | Apache Hbase | 1.7% |
| Google Cloud Storage | 5.5% |  |  |

**Framework, thư viện, công cụ**

Bảng dưới đây là danh sách các framework, thư viện, công cụ phổ biến đang được mọi người sử dụng,

|  |  |  |  |
| --- | --- | --- | --- |
| ***Framework, thư viện, công cụ*** | ***Tỉ lệ*** | ***Framework, thư viện, công cụ*** | ***Tỉ lệ*** |
| Node.js (JS) | 49.6% | Cordova (JS mobile) | 8.5% |
| Angular (JS) | 36.9% | TensorFlow (Machine Learning) | 7.8% |
| React (JS) | 27.8% | Xamarin (mobile app) | 7.4% |
| .NET Core | 27.2% | Spark (Java) | 4.8% |
| Spring (Java) | 17.6% | Hadoop (big data, distributed data) | 4.7% |
| Django (Python) | 13.0% | Torch/PyTorch (deep learning – python) | 1.7% |

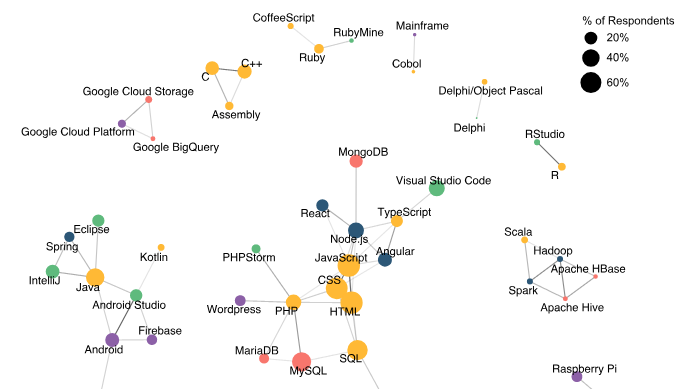
**Trình viết mã**

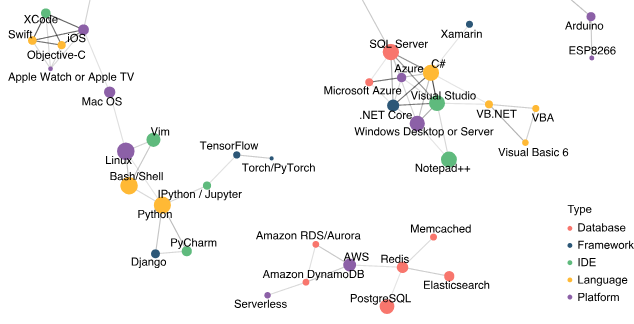
Bảng dưới đây là danh sách trình soạn thảo mã nguồn (IDE, code editor) đang được mọi người sử dụng,

|  |  |  |  |
| --- | --- | --- | --- |
| ***IDE/code editor*** | ***Tỉ lệ*** | ***IDE/code editor*** | ***Tỉ lệ*** |
| Visual Studio Code | 34.9% | PHPStorm | 9.0% |
| Visual Studio | 34.3% | NetBeans | 8.2% |
| Notepad++ | 34.2% | IPython / Jupyter | 7.4% |
| Sublime Text | 28.9% | Emacs | 4.1% |
| Vim | 25.8% | Rstudio | 3.3% |
| IntelliJ | 24.9% | RubyMine | 1.6% |
| Android Studio | 19.3% | TextMate | 1.1% |
| Eclipse | 18.9% | Coda | 0.6% |
| Atom | 18.0% | Komodo | 0.6% |
| PyCharm | 12.0% | Zend | 0.4% |
| Xcode | 10.6% | Light Table | 0.2% |

**Các nhóm công nghệ liên quan**

Hình sau là các nhóm công nghệ liên quan,





**Một số hướng đang được quan tâm nhiều**

– Web

– Mobile

– IoT

– Machine Learning

– Data Science

## Theo nghề web, nên học gì?

Muốn làm web cần học chuyên môn và tối thiểu phải biết sử dụng tiếng Anh.

Về khả năng sử dụng tiếng Anh, bước đầu tiên cần phải có là khả năng đọc để hiểu tài liệu chuyên ngành bằng tiếng Anh. Để hiểu được tiếng Anh chuyên ngành, cần hai thứ: từ vựng là quan trọng nhất, sau đó đến ngữ pháp (cấu trúc câu, mệnh đề, chia động từ, thể bị động, thể chủ động, các từ loại).

Ví dụ một cách học tiếng Anh chuyên ngành:

– Chọn một chủ đề bạn quan tâm (về HTML chẳng hạn)

– Đọc sách/tài liệu tiếng Việt về HTML trước, chọn cuốn sách/tài liệu nào mà tác giả hay ghi các thuật ngữ tiếng Anh đi kèm là tốt nhất

– Đọc hiểu hoặc làm theo sách/tài liệu tiếng Anh về HTML

– Nếu được, nên tập dịch tài liệu từ tiếng Anh ra tiếng Việt

– Trong quá trình học, nên viết các từ tiếng Anh chưa biết nghĩa ra giấy, tra nghĩa của nó, viết một hai lần thôi, chứ đừng viết liền một lúc cả trăm lần từ đó, điều này làm mất thời gian và cũng sẽ quên. Sau này nếu gặp lại từ này mà mình chưa nhớ nghĩa, thì lại viết lại và tra nghĩa lại, cho tới khi nào thực sự nhớ được từ và nghĩa của nó thì thôi, không phải viết và tra nghĩa nữa. Qua việc viết, từ mới cùng với nghĩa của nó sẽ đi vào đầu mình lúc nào không hay!

Ban đầu cần học để có thể đọc hiểu được tài liệu chuyên môn bằng tiếng Anh, sau đó học nghe, nói, viết; nếu học thêm được các ngoại ngữ khác càng tốt (tiếng Nhật chẳng hạn).

Về chuyên môn, bảng dưới đây là liệt kê một số những thứ cần học mang tính tham khảo, người học sẽ cảm thấy việc học nhẹ nhàng và hiệu quả hơn nếu học đúng theo trình tự. Tuy nhiên, nếu tự học hoặc vì điều kiện nào đó mà không thể đáp ứng được đúng thứ tự như trong bảng cũng không sao. Cứ tự tin học, thấy còn thiếu kiến thức gì thì bổ sung từ từ, quan trọng là đam mê, kiên trì và dám bước tới.

Xem bảng minh họa một số thứ cần học (chỉ mang tính tham khảo),

|  |  |
| --- | --- |
| **Kiến thức** | **Mô tả chi tiết** |
| Kiến thức chung (dù theo front-end hay back-end) | - Kiến thức nền về Hệ điều hành, Kĩ thuật lập trình, Cấu trúc dữ liệu và thuật toán, Lập trình hướng đối tượng, Cơ sở dữ liệu  - Kiến thức nền về WWW, Mạng máy tính  - Kĩ năng và thái độ về khía cạnh kĩ thuật (cách viết mã, lựa chọn giải thuật, tổ chức mã nguồn chuyên nghiệp), cũng như khía cạnh nghề nghiệp (cách làm việc chuyên nghiệp, hiệu quả và hạnh phúc với nghề)  - Phần mềm Quản lý phiên bản/quản lý mã nguồn: Git, SVN, TFS  - AJAX / Web API  - RESTful API / HTTP method  - Cơ bản về lập trình front-end và back-end |
| Front-end | - HTML, CSS, JavaScript  - Các thư viện và framework phổ biến (ví dụ: jQuery, Bootstrap, AngularJS, ReactJS, VueJS)  - Responsive design, UI/UX  - Các kiến thức nâng cao (ví dụ: mô hình MVC, design pattern, lập trình bất đồng bộ trong JavaScript) |
| Back-end | - Các ngôn ngữ phổ biến: Java, C#, PHP, Ruby, Python, NodeJS  - Cơ sở dữ liệu (SQL, mySQL, MongoDB)  - Mô hình MVC, Design pattern  - Các framework của mỗi ngôn ngữ  - Các kiến thức nâng cao (ví dụ Dependency Injection, Inversion of Control, SOLID)  - Các kiến thức về triển khai, bảo mật |

Bảng trên là liệt kê các thứ cần học để tham khảo thôi. Để nắm hết các nội dung đó quả là thử thách thực sự, và cũng không cần thiết.

Phương pháp học là:

– Học từ dễ đến khó, học chắc từng khái niệm cơ bản, khi có cái nền tốt rồi, cứ từng bước bồi đắp thêm các kiến thức mới, mọi thứ rất logic và nhẹ nhàng

– Bắt đầu là HTML, CSS, JavaScript, một vài thư viện/framework phổ biến (ví dụ: jQuery, Bootstrap, AngularJS, VueJS, ReactJS)

– Sau đó là một ngôn ngữ back-end nào đó cùng với cơ sở dữ liệu tương ứng (Java, C#, PHP, Python, Ruby, mySQL, SQL), một vài framework phổ biến

– Mục tiêu là hiểu được cách thức để làm được một sản phẩm web hoàn chỉnh (đây chính là học front-end và back-end cơ bản)

– Trong quá trình học sẽ gặp một số khái niệm, mô hình, cố gắng tìm hiểu để nắm bắt

– Làm thực hành thật nhiều, tự làm các dự án từ dễ đến khó

– Học từ các trang dạy làm web tại Facebook/Github/Youtube/Website/Blog hoặc học từ chia sẻ của các anh/chị trong nghề

– Cuối cùng, quan trọng nhất là phải kiên trì, chăm chỉ, “cứ đi ắt sẽ đến!”.

## Giới thiệu môn học

– Mã học phần: CT3111D

– Tên học phần: Lập trình cơ sở dữ liệu

– Số tín chỉ: 4 (2 LT + 2 TH)

– Hình thức đánh giá:

­­ – Chuyên cần (10%) + tập tin ghi chép (10%) + kiểm tra thực hành: 30%

– Thi cuối kỳ: 50% (trên máy)

– Tài liệu tham khảo: bài giảng, tài liệu, bài lab, Internet

– Email giáo viên: [legiacong@gmail.com](mailto:legiacong@gmail.com)

– Tài nguyên môn học lấy tại đây (cập nhật thường xuyên): <https://drive.google.com/open?id=1CgBCZg44WUhd1hRBbOsGSlRVNEnmOovx>

Nội dung môn học:

– Nắm được các khái niệm liên quan đến CSDL

– Lập trình cơ sở dữ liệu với Windows Form

– CRUD (Create, Read, Update & Delete)

– LinQ

– Entity Framework

# Lập trình WinForms

## Cài đặt Visual Studio

Danh sách các phiên bản Visual Studio, C# và Framework .NET[[7]](#footnote-7)

|  |  |  |  |
| --- | --- | --- | --- |
| ***Visual Studio*** | ***Phiên bản*** | ***.NET framework*** | ***Năm*** |
| Visual Studio 2019 | 16.0 | Chưa biết | Chưa biết |
| Visual Studio 2017 | 15.0 | 3.5 – 4.7 | March 7, 2017 |
| Visual Studio 2015 | 14.0 | 2.0 – 4.6 | July 20, 2015 |
| Visual Studio 2013 | 12.0 | 2.0 – 4.5.2 | October 17, 2013 |
| Visual Studio 2012 | 11.0 | 2.0 – 4.5.2 | September 12, 2012 |
| Visual Studio 2010 | 10.0 | 2.0 – 4.0 | April 12, 2010 |
| Visual Studio 2008 | 9.0 | 2.0, 3.0, 3.5 | November 19, 2007 |
| Visual Studio 2005 | 8.0 | 2.0, 3.0 | November 7, 2005 |
| Visual Studio .NET 2003 | 7.1 | 1.1 | April 24, 2003 |
| Visual Studio .NET (2002) | 7.0 | 1.0 | February 13, 2002 |
| Visual Studio 6.0 | 6.0 | N/A | June 1998 |
| Visual Studio 97 | 5.0 | N/A | February 1997 |

**Bản Visual studio nào miễn phí, bản nào phải trả tiền?**

Ví dụ, với Visual Studio 2017 thì nên chọn bản miễn phí là Community.

Bản phải trả tiền là Professional và Enterprise

Đọc thêm tại đây: <https://visualstudio.microsoft.com/vs/compare/>

Lab . Chọn phiên bản và cài đặt Visual Studio

Câu hỏi . Bạn đã và sẽ áp dụng mô hình bloom vào việc học như thế nào?

Câu hỏi . Bạn dự định làm gì sau khi ra trường, bạn đã, đang và sẽ tập trung học những kiến thức gì?

Câu hỏi . Phân biệt Visual Studio, .NET framework và C#?

Câu hỏi . Phân biệt code editor và IDE?

## Tổng quan về lập trình WinForms

Bài 1: Tổng quan về lập trình WinForms

<https://www.youtube.com/watch?v=dtYVRWfGhzI&index=1&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

WinForms[[8]](#footnote-8) hay Windows Forms là một thư viện hỗ trợ lập trình các ứng dụng chạy trên desktop, laptop và tablet.

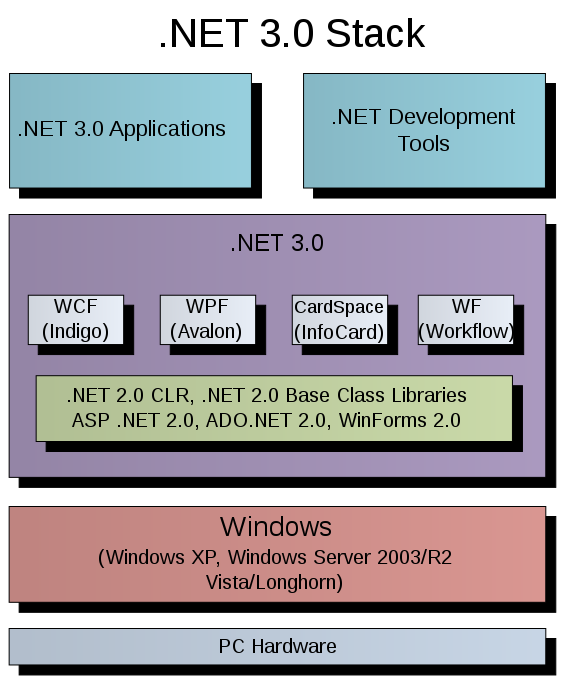
Nó là một thành phần trong Framework .NET của Microsoft.

Ra đời năm 2003, tới năm 2014 Microsoft ngưng cập nhật các chức năng mới cho WinForms.

Ưu điểm của WinForms là đơn giản, lập trình dễ dàng.

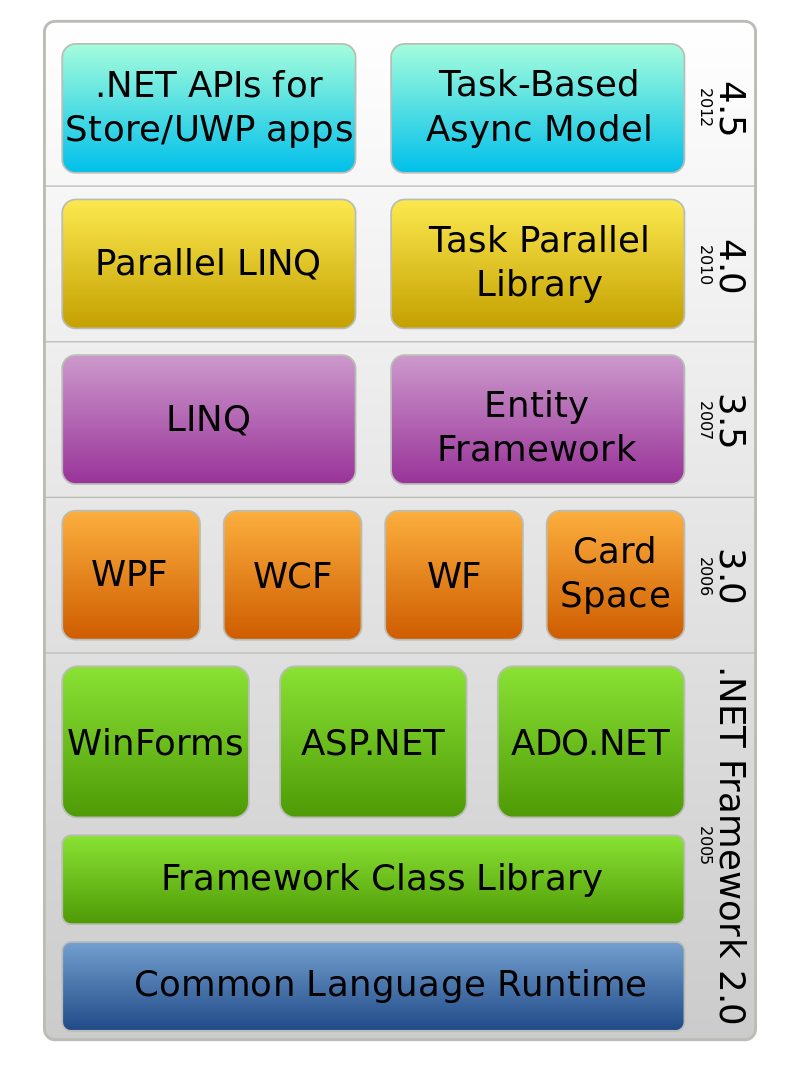
Xem hình minh họa nền tảng .NET 3.0,

*.NET 3.0*



Hình sau là các phiên bản của Framework .NET[[9]](#footnote-9)

*Các phiên bản của .NET framework*



**WPF**

WPF (Windows Presentation Foundation) là công nghệ kế tiếp của WinForms, dùng để xây dựng các ứng dụng cho các máy tính chạy Windows.

WPF ra đời năm 2006, sử dụng XAML để làm giao diện.

Có nhiều tính năng nổi trội hơn so với WinForms.

## Form

Bài 2: Tổng quan về Form

<https://www.youtube.com/watch?v=YDbbrrHEnfM&index=2&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

</////hết buổi 1

**Phân biệt Solution và Project**

<https://stackoverflow.com/questions/2389226/visual-studio-project-vs-solution>

[A solution is a container for projects, and tracks dependencies between projects.]

[A solution is like a house, a project like a room. Each room provides a piece of functionality whereas the house, a container of rooms, provides the means to link the rooms together and organize them appropriately.]

[It doesn't help that Visual Studio seems to make things more confusing. "New Project" actually creates a new SOLUTION containing a project. "Open Project" actually opens a solution containing one (or many) project. (The file menu says "Open Project/Solution" but it really is opening solutions. There is no "Close Project" only "Close Solution" which is accurate.

So, in VS you are always working within a solution. Many solutions contain only one project and newer developers are likely to think of them as the same thing. However you can add other projects into a solution.]

Lab . Thực hiện thuần thục các thao tác sau: tạo project, lưu project/solution, đóng solution, mở lại project/solution, thêm project vào solution

Ngôn ngữ biên dịch (compiled language) là loại ngôn ngữ cần được dịch toàn bộ mã nguồn chương trình ra mã máy hoặc bytecode mới có thể thực thi được. Có đầu vào (chương trình) tường minh. Ví dụ: C#, C++, Java

Ngôn ngữ thông dịch (interpreted language) là loại ngôn ngữ khi nào thực thi mới dịch mã nguồn sang mã máy. Có đầu vào tường minh. Ví dụ: PHP, Python, JavaScript

Ngôn ngữ kịch bản (scripting language) là loại ngôn ngữ mà mã nguồn sẽ được thực thi từ đầu đến cuối như một “tờ kịch bản”, không có đầu vào tường minh. Ngôn ngữ kịch bản thuộc kiểu ngôn ngữ thông dịch. Ví dụ: JavaScript, PHP

Một số câu hỏi cần tìm hiểu:

Câu hỏi . Hàm static void main() để làm gì? Nó nằm ở tập tin nào trong một project?

Câu hỏi . Mối liên hệ giữa hai tập tin FormXYZ.Designer.cs và FormXYZ.cs là gì?

Câu hỏi . Cách lấy một thiết kế form từ nơi khác đưa vào project?

Câu hỏi . Thứ tự các hàm sẽ được chạy của một chương trình đơn giản?

Có thể định dạng và thiết lập các thuộc tính cho form và các control của nó bằng properties.

</////hết buổi 2

## Label

Bài 3: Label

<https://www.youtube.com/watch?v=hzmW2PlNQ5c&index=3&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

Sử dụng thuộc tính anchor để neo label theo các cạnh của form.

Phương thức Show() sẽ hiển thị form “được gọi” ngang hàng với form “gọi”, có thể cùng lúc thao tác trên hai form. Phương thức ShowDialog() chỉ cho người dùng thao tác trên form “được gọi”.

Muốn truy cập control từ form khác thiết lập thuộc tính Modifiers của control là Public.

Đoạn mã thêm một control bằng cách viết mã (không dùng kéo thả):

public Form1()

{

InitializeComponent();

loadLabel();

}

void loadLabel()

{

Label lb = new Label();

lb.Text = "Nhãn mới thêm";

this.Controls.Add(lb);

lb.BackColor = Color.Aqua;

}

Một method mà không chỉ định kiểu truy cập (modifier) thì sẽ mặc định là private. Đọc thêm tại đây: <https://stackoverflow.com/questions/2917495/method-without-access-modifier>

## Button

Bài 4: Button

<https://www.youtube.com/watch?v=hzbvLiCrg-A&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS&index=4>

Đổi màu một nút khi rê chuột, khi bấm, thiết lập trong thuộc tính FlatAppearance.

Để nút không đổi kích thước theo hình ảnh, thiết lập thuộc tính AutoSize là True.

Để xem đoạn mã của lớp cha, sử dụng phím F12.

Thiết lập hiển thị giữa hình nền và chữ của button, sử dụng thuộc tính TextImageRelation.

Đoạn mã: bấm vào một nút, tạo các nút mới có vị trí ngẫu nhiên nằm trong form, khi bấm vào nút vừa tạo sẽ xuất giá trị Text của button:

(ép kiểu tham số sender, thêm một phương thức cho một control: *btn.Click += Btn\_Click*)

int i = 0;

void addButton()

{

Random rand = new Random();

Button btn = new Button() {

Text = i.ToString(),

Location = new Point(rand.Next(0, this.Size.Width), rand.Next(0, this.Size.Height))

};

btn.Click += Btn\_Click;

this.Controls.Add(btn);

i++;

}

private void Btn\_Click(object sender, EventArgs e)

{

//ép sender về kiểu button

Button btn = sender as Button;

MessageBox.Show(btn.Text);

}

private void button1\_Click(object sender, EventArgs e)

{

addButton();

}

## TextBox

Bài 5: Textbox

<https://www.youtube.com/watch?v=MsSds2bDqKA&index=5&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

Phân biệt hai thuộc tính enable và readonly của một textbox.

Thuộc tính tabindex để thiết lập thứ tự control sẽ được chọn (focus), giá trị nhỏ sẽ được chọn trước.

Đoạn mã sau sẽ cho nhập nội dung vào một textbox, bấm nút, xuất nội dung của textbox ra một label:

private void button1\_Click(object sender, EventArgs e)

{

lbValue.Text = txbValue.Text;

}

Sử dụng thuộc tính Acceptbutton của form để thiết lập phím Enter, nhằm thực hiện chức năng của một button (gán phím Enter cho một nút).

Sử dụng thuộc tính Cancelbutton của form để thiết lập phím Esc, nhằm thực hiện chức năng của một button (gán phím Esc cho một nút).

Để xóa một dòng mã, sử dụng phím Ctrl + L.

Nhập một số vào textbox, tính bình phương, rồi xuất ra label:

private void button1\_Click(object sender, EventArgs e)

{

int num = Convert.ToInt32(txbValue.Text);

lbValue.Text = (num \* num).ToString();

}

Sử dụng TryParse để kiểm tra kiểu dữ liệu có phù hợp hay không:

private void button1\_Click(object sender, EventArgs e)

{

int num = 0;

if (Int32.TryParse(txbValue.Text, out num))

lbValue.Text = (num \* num).ToString();

else

lbValue.Text = "Vui lòng nhập số!";

}

Sử dụng event TextChanged,

private void txbValue\_TextChanged(object sender, EventArgs e)

{

int num = 0;

if (Int32.TryParse(txbValue.Text, out num))

lbValue.Text = (num \* num).ToString();

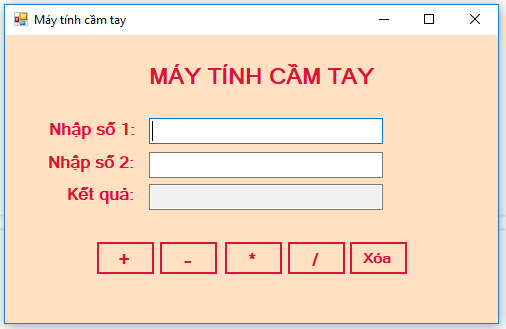
else

lbValue.Text = "Vui lòng nhập số!";

}

Lab . Làm một máy tính cầm tay (calculator) đơn giản.

1. Thiết kế form như sau,



*lbTieuDe, txtSo1, txtSo2, txtKetQua, btnCong, btnTru, btnNhan, btnChia, btnXoa*

2. Mô tả chức năng:

– Khi chạy chương trình dấu nhắc (focus) sẽ nằm tại ô nhập liệu Nhập số 1. Khi người dùng bấm phím Tab trên bàn phím, thứ tự focus sẽ lần lượt là: Nhập số 2, +, -, \*, /, Xóa, Kết Quả

– Ô Kết quả không cho nhập dữ liệu, chỉ để xuất kết quả, người dùng có thể chọn và copy kết quả

– Người dùng nhập hai giá trị số vào hai ô (Nhập số 1 và Nhập số 2), khi bấm các nút +, -, \*, /, chương trình sẽ thực hiện phép toán tương ứng và xuất kết quả vào ô Kết quả

– Khi người dùng bấm nút Xóa, hoặc phím Esc (trên bàn phím) toàn bộ thông tin trong các ô nhập liệu sẽ được xóa, đặt focus vào ô Nhập số 1

– Khi người dùng bấm vào các phép toán cần kiểm tra để đảm bảo các đầu vào phải là các số. Với phép chia, cần kiểm tra để đảm bảo không có phép chia cho 0

– Nếu có lỗi nhập liệu đặt lại focus tại ô bị lỗi

-----

Đoạn mã tham khảo:

namespace Calculator

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void btnCong\_Click(object sender, EventArgs e)

{

double so1, so2;

//so1 = Convert.ToDouble(txtSo1.Text);

//so2 = Convert.ToDouble(txtSo2.Text);

//txtKetQua.Text = (so1 + so2).ToString();

if (Double.TryParse(txtSo1.Text, out so1) && Double.TryParse(txtSo2.Text, out so2))

txtKetQua.Text = (so1 + so2).ToString();

else if (!Double.TryParse(txtSo1.Text, out so1))

txtSo1.Focus();

else

txtSo2.Focus();

}

private void btnTru\_Click(object sender, EventArgs e)

{

double so1, so2;

//so1 = Convert.ToDouble(txtSo1.Text);

//so2 = Convert.ToDouble(txtSo2.Text);

//txtKetQua.Text = (so1 - so2).ToString();

if (Double.TryParse(txtSo1.Text, out so1) && Double.TryParse(txtSo2.Text, out so2))

txtKetQua.Text = (so1 - so2).ToString();

else if (!Double.TryParse(txtSo1.Text, out so1))

txtSo1.Focus();

else

txtSo2.Focus();

}

private void btnNhan\_Click(object sender, EventArgs e)

{

double so1, so2;

//so1 = Convert.ToDouble(txtso1.Text);

//so2 = Convert.ToDouble(txtSo2.Text);

//txtKetQua.Text = (so1 \* so2).ToString();

if (Double.TryParse(txtSo1.Text, out so1) && Double.TryParse(txtSo2.Text, out so2))

txtKetQua.Text = (so1 \* so2).ToString();

else if (!Double.TryParse(txtSo1.Text, out so1))

txtSo1.Focus();

else

txtSo2.Focus();

}

private void btnChia\_Click(object sender, EventArgs e)

{

double so1, so2;

//so1 = Convert.ToDouble(txtso1.Text);

//so2 = Convert.ToDouble(txtSo2.Text);

//txtKetQua.Text = (so1 / so2).ToString();

if (Double.TryParse(txtSo1.Text, out so1) && Double.TryParse(txtSo2.Text, out so2))

{

if (so2 > 0)

txtKetQua.Text = (so1 / so2).ToString();

else

txtSo2.Focus();

}

else if (!Double.TryParse(txtSo1.Text, out so1))

txtSo1.Focus();

else

txtSo2.Focus();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

txtSo1.Clear();

txtSo2.Clear();

txtKetQua.Clear();

txtSo1.Focus();

}

}

}

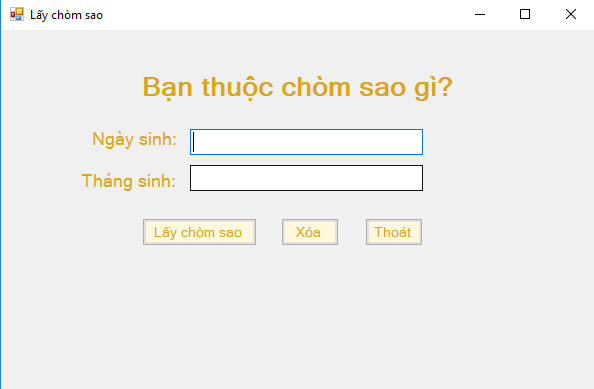
</////hết buổi 3

Câu hỏi . Properties của form hoặc control là gì?

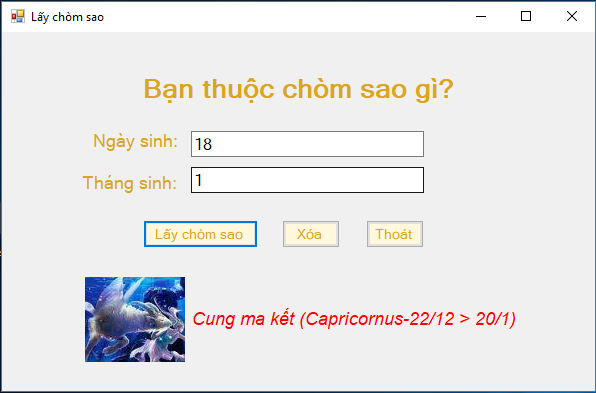
Câu hỏi . Events của form hoặc control là gì?

Lab . Viết chương trình “cho biết bạn thuộc chòm sao nào, dựa vào ngày/tháng sinh”.

Với giao diện như sau:



Sau khi bấm vào nút “Lấy chòm sao”:



Yêu cầu chức năng:

– Đặt TabIndex cho các ô nhập liệu và các nút cho hợp lý

– Cho người dùng nhập ngày + tháng sinh

– Nếu ngày, tháng sinh hợp lệ, khi người dùng bấm vào nút “Lấy chòm sao!” thì hiển thị: hình, cung hoàng đạo (tên chòm sao) và ngày tháng của cung hoàng đạo đó

– Nếu người dùng nhập dữ liệu không hợp lệ thì đặt lại focus vào ô bị lỗi nhập liệu

– Nếu người dùng bấm vào nút “Xóa” hoặc phím Esc thì xóa hết dữ liệu trên giao diện

– Nếu người dùng bấm nút “Thoát” thì đóng chương trình

Danh sách các chòm sao:

– Chòm sao ARIES – DƯƠNG CƯU sinh từ ngày 21/3 – 20/4

– Chòm sao TAURUS – KIM NGƯU sinh từ ngày 21/4 – 21/5

– Chòm sao GEMINI – SONG TỬ sinh từ ngày 22/5 – 21/6

– Chòm sao CANCER – CỰ GIẢI sinh từ ngày 22/6-23/7

– Chòm sao LEO – HẢI SƯ sinh từ ngày 24/7 – 23/8

– Chòm sao VIRGO – XỬ NỬ sinh từ ngày 24/8 – 23/9

– Chòm sao LIBRA – THIÊN XỨNG sinh từ ngày 24/9 – 23/10

– Chòm sao SCORPIO – HỔ CÁP sinh từ ngày 24/10 – 22/11

– Chòm sao SAGITTARIUS – NHÂN MÃ sinh từ ngày 23/11 – 21/12

– Chòm sao CAPRICORNUS – MA KẾT sinh từ ngày 22/12 – 20/1

– Chòm sao AQUARIUS – BẢO BÌNH sinh từ ngày 21/1 – 19/2

– Chòm sao PISCES – SONG NGƯ sinh từ ngày 20/2 – 20/3

[Gợi ý]

– Để hiển thị hình ảnh: sử dụng control Picture Box

– Thuật toán:

Khi người dùng bấm nút “Lấy chòm sao”:

if (ngày và tháng đều là số)

{

switch(tháng)

case (tháng 1)

if (ngày < 1 hoặc ngày > 31) > lỗi nhập dữ liệu

if (ngày <= 19) { lấy hình ảnh, tên chòm sao và thời gian để hiển thị }

if(ngày >= 20) { lấy hình ảnh, tên chòm sao và thời gian để hiển thị }

break;

case (tháng 2)

…

}

else // dữ liệu không phải là số

{

Đặt lại focus tại ô bị lỗi nhập dữ liệu

}

[đoạn mã tham khảo]

private void btnThoat\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void btnXoa\_Click(object sender, EventArgs e)

{

txtNgay.Clear();

txtThang.Clear();

}

private void btnChomSao\_Click(object sender, EventArgs e)

{

int ngay, thang;

if (Int32.TryParse(txtNgay.Text, out ngay) && Int32.TryParse(txtThang.Text, out thang))

{

switch (thang)

{

case 1:

if (ngay < 1 || ngay > 31)

txtNgay.Focus();

else

{

if (ngay <= 20)

{

// tạo và chèn hình ảnh

PictureBox pic = new PictureBox

{

Name = "pictureBox",

Size = new Size(100, 85),

Location = new Point(83, 245),

ImageLocation = @"..\..\..\images\cung\_ma\_ket.jpg",

SizeMode = PictureBoxSizeMode.CenterImage

};

this.Controls.Add(pic);

// tạo nội dung văn bản

Label lbKetQua = new Label

{

Name = "labelKetQua",

Location = new Point(186, 276),

AutoSize = true,

ForeColor = Color.Red,

Font = new Font("Arial", 13, FontStyle.Italic),

Text = "Cung ma kết (Capricornus-21/1 > 19/2)"

};

this.Controls.Add(lbKetQua);

}

if (ngay >= 20)

{

}

}

break;

default:

break;

}

}

else

{

if (!Int32.TryParse(txtNgay.Text, out ngay))

txtNgay.Focus();

if (!Int32.TryParse(txtThang.Text, out thang))

txtThang.Focus();

}

}

<///// hết buổi 4

Câu hỏi . Thuộc tính anchor dùng để làm gì?

Câu hỏi . Thuộc tính Modifiers dùng để làm gì?

Câu hỏi . Phân biệt phương thức Show() và ShowDialog()?

## CheckBox

Bài 6. CheckBox

<https://www.youtube.com/watch?v=7mTR8t9qFKI&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS&index=6>

Checkalign để thay đổi vị trí của ô check so với phần chữ

Checked: 2 trạng thái (checked, unchecked)

CheckState: 3 trạng thái (checked, unchecked, indeterminate). Cần thiết lập thuộc tính ThreeState là True. Trạng thái checked và indeterminate đều trả về giá trị True, unchecked trả về giá trị False.

Vậy, ngoài khác biệt về số trạng thái (2 và 3), thì Checked và CheckState còn khác nhau điều gì? Chúng khác nhau ở hai sự kiện là CheckedChanged và CheckStateChanged. Ví dụ,

private void checkBox1\_Click(object sender, EventArgs e)

{

MessageBox.Show("Clicked");

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

MessageBox.Show("CheckedChanged");

}

private void checkBox1\_CheckStateChanged(object sender, EventArgs e)

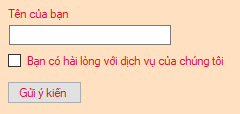
{

MessageBox.Show("CheckStateChanged");

}

Lab . Chương trình ví dụ sử dụng checkbox

[Giao diện]



[Mô tả chức năng]

– Khi người dùng bấm vào nút Gửi ý kiến: lấy “tên” và ý kiến của người dùng, xuất ra một MessageBox

[Đoạn mã tham khảo]

private void btnGui\_Click(object sender, EventArgs e)

{

string name = txtName.Text;

string haiLong = chkHaiLong.CheckState == CheckState.Checked ? "hài lòng" : chkHaiLong.CheckState == CheckState.Unchecked ? "không hài lòng" : "phân vân";

string strShow = string.Format("Bạn {0} cảm thấy {1} về dịch vụ!", name, haiLong);

MessageBox.Show(strShow);

}

Để ý cách sử dụng hàm string.Format("abcdef {0} ghijk {1} mnop!", noiDung0, noiDung1). Nội dung của biến noiDung0 sẽ được chèn vào {0}, nội dung của biến noiDung1 sẽ được chèn vào {1}.

## MessageBox

Bài 7. MessageBox

<https://www.youtube.com/watch?v=2cfnkRI2E8Y&index=7&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

MessageBox không phải là một control mà là một lớp hỗ trợ.

Tại một thời điểm chỉ có một MessageBox được xuất hiện trên giao diện.

Để thêm tiêu đề cho cửa sổ MessageBox, sử dụng tham số caption, ví dụ:

MessageBox.Show("Nội dung thông báo", "Nội dung của thanh tiêu đề (caption)");

MessageBox.Show("Bạn chưa nhập dữ liệu!", "Thông báo");

Để tùy chỉnh các nút cho cửa sổ MessageBox, sử dụng tham số MessageBoxButton, ví dụ:

MessageBox.Show("Bạn có muốn đóng chương trình không?", "Thông báo", MessageBoxButtons.YesNoCancel);

Để thêm biểu tượng (icon) cho cửa sổ MessageBox, sử dụng tham số MessageBoxIcon, ví dụ:

MessageBox.Show("Bạn có muốn đóng chương trình không?", "Thông báo", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);

Để thiết lập nút được chọn mặc định (focus) cho cửa sổ MessageBox, sử dụng tham số MessageBoxDefaultButton, ví dụ:

MessageBox.Show("Bạn có muốn đóng chương trình không?", "Thông báo", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question, MessageBoxDefaultButton.Button3);

Có thể nhận kết quả trả về của cửa sổ MessageBox, để sử dụng cho các xử lý khác, ví dụ:

DialogResult result = MessageBox.Show("Bạn có muốn đóng chương trình không?", "Thông báo", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question, MessageBoxDefaultButton.Button3);

if ( result == System.Windows.Forms.DialogResult.Yes )

{

MessageBox.Show("Người dùng vừa bấm nút Yes");

}

Thực hiện Insert Snippets cho lệnh switch *(Với các lệnh khác cách làm tương tự, chỉ cần thay tên lệnh)*, cách làm như sau:

– Để dấu nháy chuột ở chỗ cần chèn lệnh switch,

– Sử dụng tổ hợp phím tắt Ctrl+K, + X

– Chọn Visual C#

– Chọn switch

– Thay chữ switch\_on bằng tên biến (result)

– Gõ phím Enter 2 lần

## Panel và FlowLayoutPanel

Bài 8. Panel và FlowLayoutPanel

<https://www.youtube.com/watch?v=Cljvl3ur1wg&index=8&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

Panel là một control dùng để chứa (gom nhóm) các control khác, giúp thuận lợi trong việc thiết kế và xử lý.

Thêm một control và panel,

Button btn = new Button();

btn.Text = "New Button";

pnlTest.Controls.Add(btn);

Sử dụng FlowLayoutPanel để sắp xếp các control trong nó có thứ tự.

## RadioButton

Bài 9. RadioButton

<https://www.youtube.com/watch?v=dnY3DchYdwI&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS&index=9>

RadioButton dùng để chọn một mục trong một nhóm các mục chọn. Để tạo ra một nhóm các mục chọn, sử dụng control Panel.

Với RadioButton, kĩ năng cơ bản nhất cần nắm là: cách duyệt các radio trong một panel, để biết được người dùng đã chọn radio nào.

Để gỡ bỏ chế độ chọn (checked) mặc định cho một radio, thiết lập thuộc tính Tabstop là false.

Để khóa các radio trong một panel, thiết lập thuộc tính Enable là false.

Để duyệt các radio trong một panel sử dụng cú pháp sau:

RadioButton rb = null;

// duyệt panel Câu hỏi 1

foreach(RadioButton item in pnlCauHoi1.Controls)

{

if(item.Checked)

{

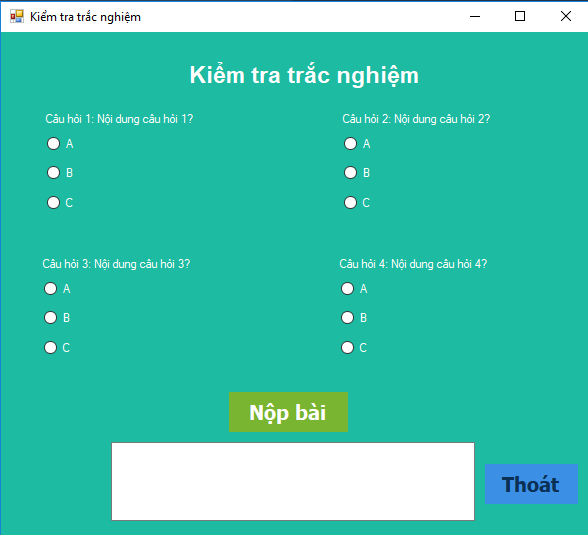
rb = item;

}

}

Lab . Tạo một chương trình trắc nghiệm đơn giản

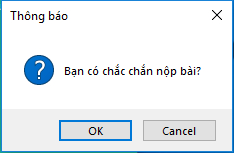
[Giao diện khi chạy chương trình – GiaoDien1]



[Giao diện sau khi người dùng đã làm bài và bấm nút Nộp bài – GiaoDien2]



[Xác nhận chắc chắn nộp bài từ người dùng – GiaoDien3]



[Mô tả chức năng]

– Chương trình cho phép người dùng chọn 01 câu trả lời cho từng câu hỏi, không nhất thiết phải làm hết các câu hỏi, có thể không trả lời câu nào cũng được

­– Nếu người dùng chưa bấm nút “Nộp bài” thì nút “Thoát” bị vô hiệu, để đảm bảo người dùng phải nộp bài mới được đóng chương trình

– Sau khi người dùng bấm nút Nộp bài thì:

+ Hiển thị cửa sổ thông báo như GiaoDien3

+ Nếu người dùng bấm Cancel thì cho tiếp tục làm bài

+ Nếu người dùng xác nhận (OK) thì thực hiện các bước sau:

++ Nếu họ không trả lời câu hỏi nào, thì xuất câu thông báo “Bạn chưa làm bài!”

++ Ngược lại, xuất kết quả như trong GiaoDien2

++ Vô hiệu (khóa) nút “Nộp bài”, khóa các mục chọn như trong GiaoDien2

++ Mở nút “Thoát” để người dùng có thể đóng chương trình

[Gợi ý cách làm]

– Khi khởi tạo form sẽ khóa nút btnThoat

– Khi người dùng bấm nút *btnNopBai*

{

Khai báo các biến

string kq = null; // để chứa chuỗi kết quả

RadioButton rb = null; // để chứa nút người dùng đã chọn

int flagXuat = 0; // để kiểm tra khi nào thì xuất dòng thông báo “Bạn chưa làm bài”, khi nào thì xuất kết quả chọn của mỗi câu

Kiểm tra kết quả trả về của GiaoDien3, nếu Cancel thì không làm gì, nếu là OK thì: duyệt qua từng panel, kiểm tra xem nút radio nào được chọn (checked), lấy Tên câu hỏi nối vào chuỗi kq, lấy phần Text của nút được chọn nối vào chuỗi kq, nối thêm kí tự xuống hàng “\r\n”

Xuất kq vào txtKetQua

}

<///// hết buổi 5

Câu hỏi . Làm sao để gỡ bỏ chế độ chọn (checked) mặc định cho một radiobutton?

## ComboBox

Bài 10. ComboBox

<https://www.youtube.com/watch?v=Uw3vymbSPO0&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS&index=10>

ComboBox là một dạng danh sách xổ xuống và cho phép người dùng lựa chọn các mục trong đó. Ví dụ như mục chọn phông chữ, kích thước phông trong Microsoft Word.

Các mục con, hiển thị trong ComboBox được thiết lập ở thuộc tính Items.

Để sử dụng chức năng autocomplete (gợi ý mục chọn) thiết lập thuộc tính AutoCompleteMode là SuggestAppend và thuộc tính AutoCompleteSource là ListItems.

Để xác lập độ rộng và kiểu của ComboBox thiết lập trong thuộc tính DropDownHeight và DropDownStyle.

Khi người dùng chọn một mục trong danh sách xổ xuống, nên quan tâm tới hai sự kiện là SelectedIndexChanged và SelectedValueChanged.

– SelectedIndexChanged: khi có sự thay đổi chỉ mục

– SelectedValueChanged: khi có sự thay đổi giá trị

Về sự khác nhau của hai sự kiện SelectedIndexChanged và SelectedValueChanged, đọc thêm tại đây: <https://stackoverflow.com/questions/4818312/combobox-events-selectedindexchanged-vs-selectedvaluechanged>

Sử dụng thuộc tính SelectedIndex để lấy giá trị index mà người dùng vừa chọn. Ví dụ,

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

ComboBox cb = sender as ComboBox;

MessageBox.Show(cb.SelectedIndex.ToString());

}

Sử dụng thuộc tính SelectedItem để lấy item trong ListItem mà người dùng vừa chọn . Ví dụ,

private void comboBox1\_SelectedValueChanged(object sender, EventArgs e)

{

ComboBox cb = sender as ComboBox;

MessageBox.Show(cb.SelectedItem.ToString());

}

Có hai cách để đưa dữ liệu vào ComboBox, một là sử dụng ListItem, hai là sử dụng DataSource.

Ví dụ, sử dụng DataSource để đưa dữ liệu vào ComboBox:

List<string> listItems;

private void button1\_Click(object sender, EventArgs e)

{

listItems = new List<string>() { "CTK40", "CTK41", "CTK42" };

comboBox1.DataSource = listItems;

}

Đọc thêm về cách khai báo và sử dụng kiểu dữ liệu List tại đây:

<https://www.howkteam.vn/course/list-trong-c/list-trong-c-1567>

Đọc thêm về Generic collection tại đây:

<https://toidicodedao.com/2015/03/05/series-c-hay-ho-generic-la-cai-thu-chi-chi/>

List là một kiểu dữ liệu generic collection, dùng để lưu trữ và quản lý một tập các đối tượng theo kiểu mảng.

Vì List là một lớp, nên trước khi sử dụng cần phải khởi tạo nó bằng từ khóa new. Kiểu dữ liệu được khai báo trong dấu <>.

Khai báo: public string MSSV { get; set; } sẽ chuyển MSSV thành một property, khi đó thay vì viết obj.get\_MSSV, thì viết thành obj.MSSV. Hoặc thay vì viết obj.set\_MSSV(001122), thì viết thành obj.MSSV = 001122. Đọc thêm ở đây: <https://daynhauhoc.com/t/cho-minh-hoi-ve-get-set-trong-c/23855/8>

Thuộc tính DisplayName của ComboBox được sử dụng để xác định trường được hiển thị. Ví dụ,

private void button1\_Click(object sender, EventArgs e)

{

listItems = new List<SinhVien>()

{

new SinhVien() { MSSV = "001122", HoTen = "Nguyễn Văn Tèo", Diem = 5 },

new SinhVien() { MSSV = "001133", HoTen = "Trần Văn Tí", Diem = 7 },

new SinhVien() { MSSV = "001144", HoTen = "Lê Huỳnh Sửu", Diem = 9 },

};

comboBox1.DataSource = listItems;

comboBox1.DisplayMember = "MSSV";

}

public class SinhVien

{

public string MSSV { get; set; }

public string HoTen { get; set; }

public float Diem { get; set; }

}

Ví dụ sau sẽ xuất MSSV và HoTen trên hai ComboBox khác nhau,

comboBox1.DataSource = listItems;

comboBox1.DisplayMember = "MSSV";

comboBox2.DataSource = listItems;

comboBox2.DisplayMember = "HoTen";

Ví dụ: xử lý khi có sự thay đổi mục chọn trong ComboBox,

private void comboBox1\_SelectedValueChanged(object sender, EventArgs e)

{

ComboBox cb = sender as ComboBox;

if ( cb.SelectedValue != null )

{

SinhVien sv = cb.SelectedValue as SinhVien;

textBox1.Text = sv.HoTen.ToString();

}

}

Có thể viết khác,

private void comboBox1\_SelectedValueChanged(object sender, EventArgs e)

{

ComboBox cb = sender as ComboBox;

if ( cb.SelectedItem != null )

{

SinhVien sv = cb.SelectedItem as SinhVien;

textBox1.Text = sv.HoTen.ToString();

}

}

SelectedItem: là lấy item trong listItem của ComboBox, là một Object???

SelectedValue: là lấy value trong DataSource của ComboBox, là một chuỗi???

Đọc thêm về sự khác nhau giữa SelectedItem và SelectedValue: <https://stackoverflow.com/questions/4902039/difference-between-selecteditem-selectedvalue-and-selectedvaluepath>

## Kĩ thuật Data Binding

Thực hiện kết buộc dữ liệu (data binding) giữa ComboBox và TextBox. Ví dụ,

Viết hàm thực hiện kết buộc dữ liệu:

void addBindings()

{

textBox1.DataBindings.Add("Text", comboBox1.DataSource, "HoTen");

}

Ý nghĩa của đoạn mã trên: lấy trường HoTen của DataSource thuộc comboBox1 kết buộc với trường Text của textBox1. Khi giá trị trong HoTen thay đổi thì cập nhật luôn giá trị tương ứng bên Text.

Gọi hàm addBindings() trong hàm khởi tạo của Form:

public Form1()

{

InitializeComponent();

listItems = new List<SinhVien>()

{

new SinhVien() { MSSV = "001122", HoTen = "Nguyễn Văn Tèo", Diem = 5 },

new SinhVien() { MSSV = "001133", HoTen = "Trần Văn Tí", Diem = 7 },

new SinhVien() { MSSV = "001144", HoTen = "Lê Huỳnh Sửu", Diem = 9 },

};

comboBox1.DataSource = listItems;

comboBox1.DisplayMember = "MSSV";

addBindings();

}

Ví dụ: tạo một ComboBox chứa danh sách các lớp, khi người dùng chọn một lớp sẽ hiển thị danh sách sinh viên của lớp ở một ComboBox khác. Sử dụng kĩ thuật data binding.

– Tạo lớp Lop,

public class Lop

{

public string Ten { get; set; }

public List<string> DanhSachSinhVien { get; set; }

}

– Tạo một danh sách chứa các lớp,

List<Lop> DanhSachLop;

– Khởi tạo dữ liệu, tạo ba lớp, mỗi lớp ba sinh viên,

DanhSachLop = new List<Lop>();

DanhSachLop.Add(new Lop()

{

Ten = "CTK40",

DanhSachSinhVien = new List<string>() { "Tí", "Sửu", "Dần" }

});

DanhSachLop.Add(new Lop()

{

Ten = "CTK41",

DanhSachSinhVien = new List<string>() { "Mão", "Thìn", "Tỵ" }

});

DanhSachLop.Add(new Lop()

{

Ten = "CTK42",

DanhSachSinhVien = new List<string>() { "Ngọ", "Mùi", "Thân" }

});

– Hiển thị dữ liệu trong comboBox1,

comboBox1.DataSource = DanhSachLop;

comboBox1.DisplayMember = "Ten";

– Hàm kết buộc dữ liệu: giữa DataSource của comboBox2 với DanhSachSinhVien của comboBox1.DataSource,

void addBinding\_Lop\_DanhSachSinhVien()

{

comboBox2.DataBindings.Add("DataSource", comboBox1.DataSource, "DanhSachSinhVien");

}

– Gọi hàm kết buộc dữ liệu giữa DataSource của comboBox2 với DanhSachSinhVien của comboBox1.DataSource,

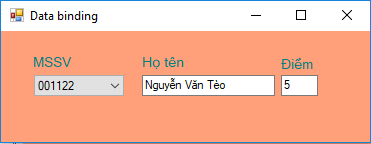
addBinding\_Lop\_DanhSachSinhVien();

Lab . Data binding giữa ComboBox và TextBox

[Giao diện khi thiết kế]



[Giao diện khi chạy]



[Yêu cầu]

– Tạo danh sách gồm ít nhất 5 sinh viên

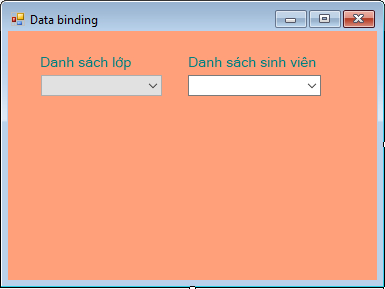
– Mỗi Sinh viên gồm có: MSSV, Họ tên, Điểm

– Sử dụng kĩ thuật data binding giữa ComboBox “MSSV” và hai TextBox “Họ tên” và “Điểm”

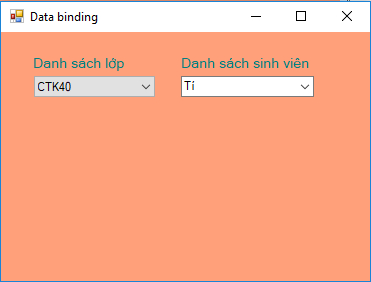
­– Khi chạy: chương trình sẽ hiển thị danh sách các MSSV trong ComboBox “MSSV”, nếu người dùng chọn một MSSV bất kì, họ tên và điểm sẽ được hiển thị ở hai TextBox tương ứng.

Lab . Data binding giữa ComboBox và ComboBox

[Giao diện khi thiết kế]



[Giao diện khi chạy]



[Yêu cầu]

– Tạo danh sách gồm 5 lớp (ví dụ: CTK38, CTK39, CTK40,…)

– Mỗi lớp gồm ít nhất 5 sinh viên

– Sử dụng kĩ thuật data binding giữa hai ComboBox “Danh sách lớp” và “Danh sách sinh viên”

­– Khi chạy: chương trình sẽ hiển thị danh sách các lớp trong ComboBox “Danh sách lớp”, nếu người dùng chọn một lớp bất kì, danh sách sinh viên của lớp đó sẽ được hiển thị trong ComboBox “Danh sách sinh viên”

Câu hỏi 15. Kiểu dữ liệu Generics là gì?

Câu hỏi 16. Trong ComboBox, sự khác nhau giữa SelectedItem và SelectedValue là gì?

Câu hỏi . Kĩ thuật data binding là gì?

<///// hết buổi 6

## PictureBox

Bài 10. PictureBox

<https://www.youtube.com/watch?v=57QobqDr8HM&index=11&list=PL33lvabfss1y2T7yK--YZJHCsU7LZVzBS>

## Cài đặt SQL Server, Visual Studio

### Cơ sở dữ liệu là gì?

Đọc trên vi.wiki: <https://vi.wikipedia.org/wiki/C%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u>

Cơ sở dữ liệu (database) là một tập hợp thông tin có cấu trúc. Các thông tin này được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành, hoặc được lưu trữ trong các hệ quản trị cơ sở dữ liệu.

Ưu điểm của cơ sở dữ liệu:

– Giảm sự trùng lặp thông tin xuống mức thấp nhất

– Có thể truy xuất thông tin theo nhiều cách

– Cho phép nhiều người cùng sử dụng một lúc

Cơ sở dữ liệu phản ánh một phần của thế giới thật.

Có nhiều loại cơ sở dữ liệu, ví dụ:

|  |  |
| --- | --- |
| **Loại cơ sở dữ liệu** | **Đặc điểm** |
| Cơ sở dữ liệu dạng file | Dữ liệu được lưu trữ dưới dạng các file (text, ascii, .dbf). |
| Cơ sở dữ liệu quan hệ | Dữ liệu được lưu trữ trong các bảng dữ liệu gọi là các thực thể, giữa các thực thể này có mối liên hệ với nhau gọi là các quan hệ, mỗi quan hệ có các thuộc tính, trong đó có một thuộc tính là khóa chính. Các hệ quản trị hỗ trợ cơ sở dữ liệu quan hệ như: MS SQL server, Oracle, MySQL,…v.v. |
| Cơ sở dữ liệu hướng đối tượng | Dữ liệu cũng được lưu trữ trong các bảng dữ liệu, nhưng các bảng có bổ sung thêm các tính năng hướng đối tượng như lưu trữ thêm các hành vi, nhằm thể hiện hành vi của đối tượng. Mỗi bảng xem như một lớp dữ liệu, mỗi dòng dữ liệu trong bảng là một đối tượng. Các hệ quản trị có hỗ trợ cơ sở dữ liệu hướng đối tượng như: MS SQL server, Oracle, PostgreSQL (viết tắt Postgres). |
| Cơ sở dữ liệu bán cấu trúc | Dữ liệu được lưu dưới dạng XML, với định dạng này, thông tin mô tả về đối tượng thể hiện trong các thẻ (tag). Đây là cơ sở dữ liệu có nhiều ưu điểm, do lưu trữ được hầu hết các loại dữ liệu khác nhau. Đây là hướng mới trong nghiên cứu và ứng dụng |

Bảng dưới đây là danh sách các cơ sở dữ liệu phổ biến đang được mọi người sử dụng,

|  |  |  |  |
| --- | --- | --- | --- |
| ***Cơ sở dữ liệu*** | ***Tỉ lệ*** | ***Cơ sở dữ liệu*** | ***Tỉ lệ*** |
| MySQL | 58.7% | Memcached | 5.5% |
| SQL Server | 41.2% | Amazon DynamoDB | 5.2% |
| PostgreSQL | 32.9% | Amazon RDS/Aurora | 5.1% |
| MongoDB | 25.9% | Cassandra | 3.7% |
| SQLite | 19.7% | IBM Db2 | 2.5% |
| Redis | 18.0% | Neo4j | 2.4% |
| Elasticsearch | 14.1% | Amazon Redshift | 2.2% |
| MariaDB | 13.4% | Apache Hive | 2.2% |
| Oracle | 11.1% | Google BigQuery | 2.1% |
| Microsoft Azure (Tables, CosmosDB, SQL, etc) | 7.9% | Apache Hbase | 1.7% |
| Google Cloud Storage | 5.5% |  |  |

### Hệ quản trị cơ sở dữ liệu là gì?

Theo vi.wiki:

<https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u>

Hệ quản trị cơ sở dữ liệu (tiếng Anh: Database Management System – DBMS) là phần mềm hay hệ thống được thiết kế để quản trị một cơ sở dữ liệu.

Các DBMS cho phép lưu trữ, sửa chữa, xóa và tìm kiếm thông tin trong một cơ sở dữ liệu. Có nhiều loại DBMS khác nhau: từ phần mềm nhỏ chạy trên máy tính cá nhân đến những hệ quản trị phức tạp chạy trên một hoặc nhiều siêu máy tính.

Đa số các DBMS đều sử dụng ngôn ngữ truy vấn theo cấu trúc, tiếng Anh gọi là Structured Query Language (SQL).

Các DBMS phổ biến gồm: mySQL, Oracle, PostgreSQL, SQL server, DB2, Infomix.

Phần lớn các DBMS hoạt động tốt trên nhiều hệ điều hành khác nhau như Linux, Unix mà MacOS, trừ SQL server của Microsoft chỉ chạy trên hệ điều hành Windows.

### Cài đặt SQL Server

**Chọn phiên bản nào[[10]](#footnote-10)?**

SQL Server hiện có nhiều phiên bản khác nhau. Dưới đây là danh sách các phiên bản cùng với tính năng của từng phiên bản.

|  |  |
| --- | --- |
| Enterprise | bản cao cấp nhất với đầy đủ tính năng |
| Standard | ít tính năng hơn Enterprise, sử dụng khi không cần dùng tới các tính năng nâng cao |
| Workgroup | phù hợp cho các công ty lớn với nhiều văn phòng làm việc từ xa |
| Web | thiết kế riêng cho các ứng dụng web |
| Developer | tương tự như Enterprise nhưng chỉ cấp quyền cho một người dùng duy nhất để phát triển, thử nghiệm, demo. Có thể dễ dàng nâng cấp lên bản Enterprise mà không cần cài lại |
| ***Express*** | ***bản này chỉ dùng ở mức độ đơn giản, tối đa 1 CPU và bộ nhớ 1GB, kích thước tối đa của cơ sở dữ liệu là 10GB (miễn phí, nhẹ, phù hợp để học)*** |
| Compact | nhúng miễn phí vào các môi trường phát triển ứng dụng web. Kích thước tối đa của cơ sở dữ liệu là 4GB |
| Datacenter | thay đổi lớn trên SQL Server 2008 R2 chính là bản Datacenter Edition. Không giới hạn bộ nhớ và hỗ trợ hơn 25 bản cài |
| Business Intelligence | Business Intelligence Edition mới được giới thiệu trên SQL Server 2012. Phiên bản này có các tính năng của bản Standard và hỗ trợ một số tính năng nâng cao về BI như Power View và PowerPivot nhưng không hỗ trợ những tính năng nâng cao về mức độ sẵn sàng như AlwaysOn Availability Groups… |
| Enterprise Evaluation | bản SQL Server Evaluation Edition là lựa chọn tuyệt vời để dùng được mọi tính năng và có được bản cài miễn phí của SQL Server để học tập và phát triển. Phiên bản này có thời gian hết hạn là 6 tháng từ ngày cài |

Danh sách các bản SQL server theo thời gian[[11]](#footnote-11)

|  |
| --- |
| SQL Server 2017 |
| SQL Server 2016 |
| SQL Server 2014 |
| SQL Server 2012 |
| SQL Server 2008 R2 |
| SQL Server 2008 |
| SQL Server 2005 |
| SQL Server 2000 |
| SQL Server 7.0 |

Tải nguồn

SQL server 2008 express (SQLEXPRADV\_x64\_ENU.exe, dung lượng ~1GB)

*Lưu ý: nếu lấy SQLEXPR\_x86\_ENU.exe (89.1 MB) hoặc SQLEXPR\_x64\_ENU.exe (82.5 MB hoặc SQLEXPR32\_x86\_ENU.exe (61.1 MB) sẽ bị thiếu Microsoft SQL Server Management Studio.*

<https://www.microsoft.com/en-us/download/details.aspx?id=30438>

Hướng dẫn cài đặt SQL server 2008 express: <https://www.youtube.com/watch?v=mxWiiyb-rBE>

SQL server 2017 express (~1.5GB): <https://www.microsoft.com/en-us/sql-server/sql-server-downloads>

Hướng dẫn cài đặt SQL server 2017 express: <https://viblo.asia/p/tong-quan-va-cach-cai-dat-cua-ms-sql-server-3P0lPzJGKox>

# Phần mềm quản lý quán café

Bài 0: Giới thiệu

<https://www.youtube.com/watch?v=tu2k9ZrDlWA&list=PL33lvabfss1xnPhBJHjM0A8TEBBcGCTsf>

Bài 1: Thiết kế CSDL

<https://www.youtube.com/watch?v=R8HRA72Orkk&index=2&list=PL33lvabfss1xnPhBJHjM0A8TEBBcGCTsf>

### Hướng dẫn kết nối, tạo database, tạo table và tạo liên kết giữa các table trong Sql Server

<https://www.youtube.com/watch?v=z4sRV45PKow&list=PLMI-61XivrDr3W22JAyK-H7kfq0MueNUg&index=1>

Chạy chương trình SQL Server Management Studio.

Cách chứng thực để kết nối vào một SQL server:

– Server type là gì?

– Server name là gì?

– Authentication là gì?

Chuột phải vào Databases\chọn New Database để tạo database mới, đặt tên là Demo.

Mở cơ sở dữ liệu Demo, chuột phải vào Tables\chọn New Table để tạo hai bảng dữ liệu mới.

**PhongBan**

ID int auto increment primary key

tenPB nvarchar(50)

**NhanVien**

maNV int auto increment primary key

hoTen nvarchar(50)

queQuan nvarchar(50)

IDPhongBan int

Chuột phải vào tên bảng\chọn Design để thiết lập khóa cho bảng. Chuột phải vào trường sẽ làm khóa chọn Set Primary Key.

Chuột phải vào Database Diagrams\chọn New Database Diagram để tạo liên kết 1-n giữa bảng PhongBan và NhanVien.

Trước khi tiếp tục với loạt bài làm việc với cơ sở dữ liệu, cần học một chút về lập trình Winforms.

### Kết nối Visual Studio với SQL server

<https://www.youtube.com/watch?v=TjMZIVo_L4I&list=PLMI-61XivrDr3W22JAyK-H7kfq0MueNUg&index=2>

Chuột phải vào bảng\chọn Edit…để nhập dữ liệu cho bảng.

# WPF

Bài 1: Tổng quan về WPF

(WPF: Windows Presentation Foundation)

<https://www.youtube.com/watch?v=5awClGNtbfU&list=PL33lvabfss1ywgHcDF2aB8YBxwtj1_Rjk>

1. https://tusach.thuvienkhoahoc.com/wiki/H%E1%BB%87\_th%E1%BB%91ng\_ph%C3%A2n\_lo%E1%BA%A1i\_c%C3%A1c\_m%E1%BB%A5c\_ti%C3%AAu\_c%E1%BB%A7a\_Bloom [↑](#footnote-ref-1)
2. <http://travisnguyen.net/selfpondering/2018/04/28/stackoverflow-history/> [↑](#footnote-ref-2)
3. <https://en.wikipedia.org/wiki/Joel_Spolsky> [↑](#footnote-ref-3)
4. <https://en.wikipedia.org/wiki/Jeff_Atwood> [↑](#footnote-ref-4)
5. <https://en.wikipedia.org/wiki/Stack_Overflow> [↑](#footnote-ref-5)
6. <https://insights.stackoverflow.com/survey/2018> [↑](#footnote-ref-6)
7. https://en.wikipedia.org/wiki/Microsoft\_Visual\_Studio [↑](#footnote-ref-7)
8. https://en.wikipedia.org/wiki/Windows\_Forms [↑](#footnote-ref-8)
9. https://en.wikipedia.org/wiki/.NET\_Framework\_version\_history [↑](#footnote-ref-9)
10. https://quantrimang.com/cac-phien-ban-ms-sql-server-145297 [↑](#footnote-ref-10)
11. https://sqlserverbuilds.blogspot.com/ [↑](#footnote-ref-11)