



LẬP TRÌNH TRỰC QUAN

GV: VÕ TUẤN KIẾT
2025

Giới thiệu chung

❖ Môn học trước:

- Nhập môn lập trình
- Lập trình hướng đối tượng

❖ Thông tin giảng viên giảng dạy:

- GV: Võ Tuấn Kiệt
- Email: kietvt@uit.edu.vn
- Phone/Zalo: 0907107385

Thông tin môn học

- ❖ **Tên môn học:** Lập trình trực quan (Visual Programming)
- ❖ **Khối kiến thức:** Cơ sở ngành
- ❖ **Số tín chỉ:** 4
- ❖ **Số tiết:**
 - Lý thuyết: 45
 - Thực hành: 30
 - Tự học: 90
- ❖ **Ngôn ngữ lập trình:** C# (C Sharp)

Mục tiêu môn học

- ✓ Nắm vững kiến thức, phương pháp lập trình trên môi trường windows và các công nghệ có liên quan
- ✓ Nắm được các kỹ thuật lập trình cao cấp với nhiều tính năng mới cùng ngôn ngữ C# trong lập trình ứng dụng trực quan
- ✓ Rèn luyện kỹ năng làm việc nhóm trong việc khai triển đồ án cuối kỳ

Tài liệu tham khảo

- [1] Microsoft Visual C# Step by Step,
Microsoft Press , John Sharp, 2022

Cách đánh giá môn học

- ❖ Quá trình: 20%
- ❖ Thực hành (thi trên máy): 30%
- ❖ Đồ án cuối kỳ: 50% (theo nhóm, tối đa 4 SV/nhóm)



Các đề án tham khảo

- Xây dựng chương trình Explorer (tương tự Window Explorer)
- Xây dựng chương trình Wordpad
- Xây dựng chương trình Paint (Windows)
- Xây dựng chương trình hỗ trợ dạy và minh họa cho các cấu trúc dữ liệu: Danh sách liên kết, stack, queue, cây nhị phân,...
- Xây dựng chương trình hỗ trợ dạy và minh họa cho các thuật toán sắp xếp
- Xây dựng chương trình hỗ trợ dạy và minh họa cho cấu trúc dữ liệu: Cây đồ đen
- Xây dựng phần mềm hỗ trợ giải bài tập khảo sát hàm số

Các đề án tham khảo

- Xây dựng phần mềm hỗ trợ giải bài tập hình học giải tích
- Xây dựng phần mềm thí nghiệm quang hình học
- Xây dựng phần mềm thí nghiệm mạch điện
- Xây dựng trò chơi cờ tướng
- Xây dựng trò chơi cờ vua
- Xây dựng trò chơi rắn săn mồi
- Xây dựng trò chơi Line
- Xây dựng trò chơi xếp hình
- Xây dựng trò chơi xếp gạch

Các đồ án tham khảo

- Nhóm các chương trình quản lý
 - Phần mềm quản lý thư viện
 - Phần mềm quản lý nhà sách
 - Phần mềm quản lý khách sạn
 - Phần mềm quản lý phòng mạch tư
 - Phần mềm quản lý sổ tiết kiệm
 - Phần mềm quản lý bán vé máy bay
 - Phần mềm quản lý cây gia phả
 - Phần mềm quản lý vật tư
 - Phần mềm quản lý bệnh viện
 - Phần mềm quản lý nhân sự cho công ty
 - Phần mềm quản lý phân phối dược phẩm

Nội dung môn học

- | | |
|---|-------------------------------------|
| 1 | Tổng quan về lập trình trên Windows |
| 2 | Tổng quan về ngôn ngữ lập trình C# |
| 3 | Lập trình giao diện đồ họa |
| 4 | Xử lý sự kiện |
| 5 | Thư viện đồ họa GDI |
| 6 | Multi-Document Interface (MDI) |
| 7 | Lập trình đa luồng |
| 8 | Multimedia |

Các công cụ hỗ trợ

❖ Môi trường phát triển tích hợp (IDE):

- Microsoft Visual Studio, Community Edition



Các công cụ hỗ trợ

❖ GitHub:

GitHub là một **nền tảng lưu trữ và cộng tác dựa trên Git**, hoạt động như một “mạng xã hội” dành cho lập trình viên.

Bạn có thể:

- Lưu trữ repo của mình trên GitHub (public hoặc private).
- Làm việc nhóm: xem, bình luận, đề xuất thay đổi qua **Pull Request**.
- Quản lý công việc bằng **Issues, Projects**.
- Tích hợp CI/CD, chạy test tự động, khai triển code trực tiếp.

Các công cụ hỗ trợ

❖ GitHub: (liên kết local với remote repo.)

- Tạo một tài khoản trên GitHub
- Cài đặt Git Client trên máy (Git for Windows, TortoiseGit, SourceTree, GitHub Desktop, ...)
- Tạo một **repository** (thùng chứa) trên GitHub (remote), để ở chế độ **private**
- Tạo một repository cho project trên local: chạy command prompt, vào thư mục project chạy lệnh:

git init

- Tạo file **.gitignore** chứa những kiểu file/thư mục không cần đưa vào repo
- Tạo liên kết giữa local và remote repo, chạy lệnh:

git remote add origin <URL đến remote git>

Các công cụ hỗ trợ

❖ GitHub: (file .gitignore điển hình cho dự án C#)

Build directories

[Bb]in/

[Oo]bj/

[Dd]ebug*/

[Rr]elease*/

Visual Studio

.vs/

*.suo

*.user

NuGet packages

packages/

*.nupkg

Debug/Cache

*.pdb

*.log

*.cache

Windows junk

Thumbs.db

Desktop.ini

Các công cụ hỗ trợ

❖ GitHub: (đưa những thay đổi ở local lên remote repo.)

- 1) Mở cửa sổ Command Prompt và chuyển vào thư mục project
- 2) Chạy lệnh đưa các thay đổi vào staging (của local):
`git add .`
(nhớ có dấu chấm ở cuối)
- 3) Chạy lệnh commit các thay đổi:
`git commit -m "message"`
(phần message ta nên nói ngắn gọn tại sao ta làm những thay đổi đó)
- 4) Chạy lệnh đưa các thay đổi trong staging lên GitHub (remote):
`git push origin <nhánh>`
(phần <nhánh> (branch) thường là main hoặc master tùy theo. Có thể xem nhánh hiện tại bằng lệnh: `git branch`)
vd: `git push origin master`