

CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM (504073)

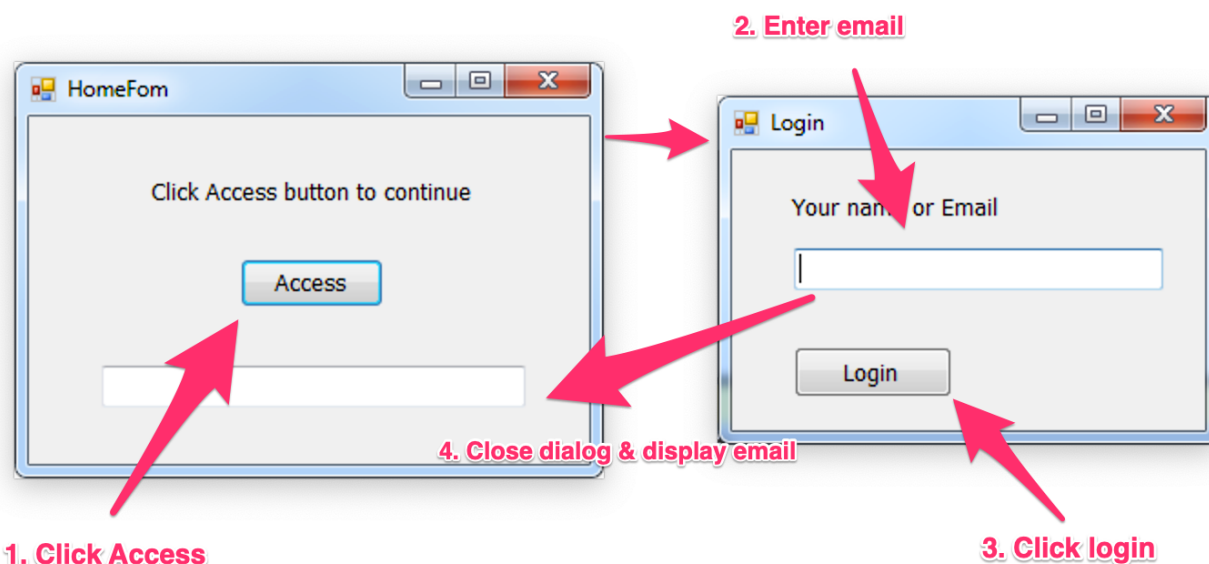
(TUẦN 01)

BIÊN SOẠN: MAI VĂN MẠNH

Sinh viên được cung cấp tập tin '**504073-Lab01.zip**' chứa các project mẫu để làm các bài tập bên dưới.

Bài 1. Sử dụng **delegate** để hoàn thành các bài tập này. Cho trước một project Windows Form gồm có hai forms: **home** và **login**.

Yêu cầu: Tại màn hình chính (home form), khi user clicks vào nút **Access**, màn hình đăng nhập (login form) xuất hiện **yêu cầu người dùng nhập tên** hoặc **địa chỉ email** của họ. Sau khi nhập tên hoặc địa chỉ email, **người dùng nhấp vào nút Login**, màn hình đăng nhập sẽ đóng và tên (hoặc địa chỉ email) đã nhập sẽ được hiển thị trong textbox của màn hình chính.

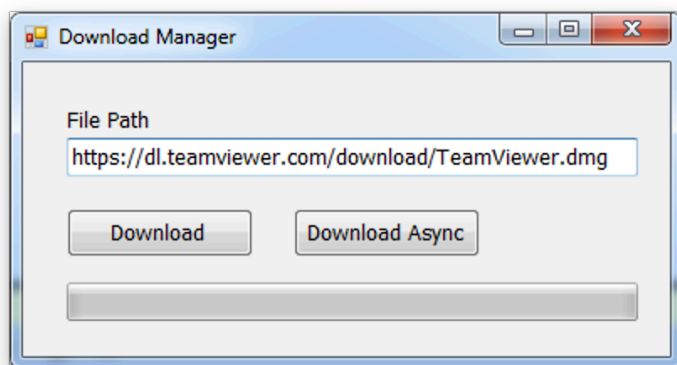


Hướng dẫn:

- Trong home form, **tạo một phương thức nhận một chuỗi làm tham số**, phương thức này đảm nhiệm việc **cập nhật giá trị textbox dựa trên tham số đầu vào**. **Tạo một đối delegate** tham chiếu đến phương thức này.
- **Tuyên đối tượng delegate** đến login form thông qua constructor hoặc setter.
- Trong **login form**, khi người dùng nhấp vào nút login, gọi (invoke) đối tượng **delegate** để gọi phương thức cập nhật giao diện của home form.

Bài 2. Bài tập sử dụng **events**. Cho trước một ứng dụng windows form thực hiện **chức năng download tập tin** có đường dẫn được nhập ở textbox. Tuy nhiên ví dụ

cung cấp sử dụng phương pháp **synchronous** khiến chương trình bị đứng, không phản hồi người dùng trong khi quá trình tải về đang diễn ra.

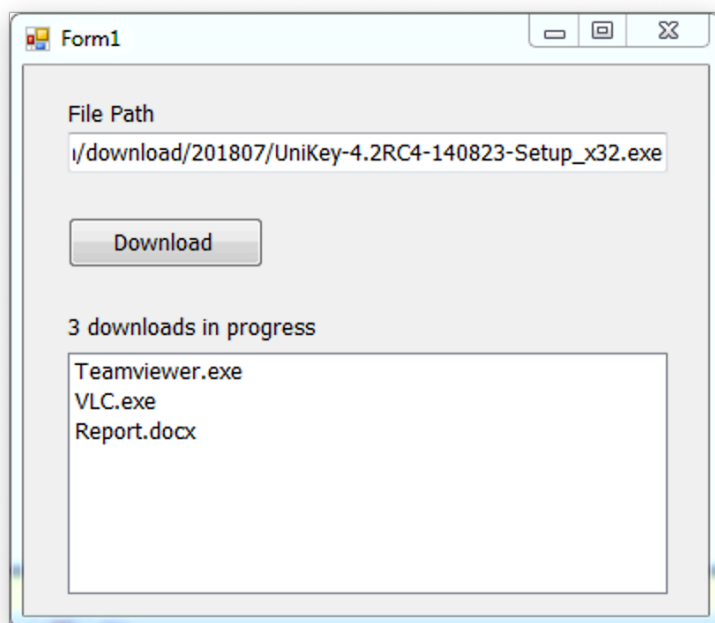


Yêu cầu: Chuyển đổi từ **phương pháp synchronous sang asynchronous**. Sử dụng các sự kiện được cung cấp bởi **WebClient** để hiển thị tiến độ trên **progress bar** và hiển thị dialog thông báo sau khi quá trình tải về hoàn tất. **Sử dụng kết hợp *lambda expression* để viết mã nguồn ngắn gọn hơn.**

Hướng dẫn:

- Chuyển từ `DownloadFile()` sang `DownloadFileAsync()`.
- Sử dụng sự kiện `DownloadProgressChanged` để cập nhật tiến độ tải về.
- Sử dụng sự kiện `DownloadFileCompleted` event để thông báo người dùng khi quá trình tải về hoàn tất.

Bài 3. Cho trước một ứng dụng Windows Form như hình bên dưới. Yêu cầu sử dụng **Thread** để hoàn thành một ứng dụng Downloader.



Yêu cầu: Khi người dùng điền liên kết vào textbox và nhấp vào nút **Download**, một **Thread** mới sẽ được tạo để thực hiện quá trình tải về. Nhiều tệp có thể được tải xuống cùng một lúc và chúng được tải xuống trong các Thread riêng biệt. Khi quá trình tải xuống đang diễn ra, tên của tệp sẽ được liệt kê trong listview. Sau khi tải xuống hoàn tất, tên tệp sẽ bị xóa khỏi listview.

Bởi vì quá trình tải mỗi tệp tin sử dụng một Thread khác nhau nên chúng ta có thể sử dụng `WebClient.DownloadFile()` để tải các tệp tin về mà không ảnh hưởng, không làm block các Thread khác.

Bài 4. List <T> là một lựa chọn rất tốt để lưu trữ mảng động trong C#, tuy nhiên nó không hỗ trợ thông báo sự kiện. Hãy viết một lớp mới mở rộng từ List<T> và cung cấp thêm một số sự kiện như OnAdd, OnRemove, OnInsert ... Viết một ứng dụng windows form để minh họa kết quả.