1. **IDE là gì?**

**IDE**, hay còn gọi là **Môi trường phát triển tích hợp (Integrated Development Environment)** là một ứng dụng phần mềm cung cấp một bộ công cụ toàn diện để phát triển phần mềm. IDE được thiết kế để làm cho quá trình phát triển phần mềm hiệu quả hơn, bằng cách cung cấp một giao diện duy nhất để quản lý tất cả các khía cạnh (aspects).

IDE cung cấp nhiều công cụ và tính năng giúp các lập trình viên có thể viết, test và debug dễ dàng hơn. IDE cũng cải thiện chất lượng code với các công cụ highlight cú pháp, hoàn thành code (code completion) và tái cấu trúc (refactoring). Ngoài ra, IDE cũng thường bao gồm hỗ trợ tích hợp cho các hệ thống kiểm soát nguồn, giúp các lập trình viên dễ dàng cộng tác và quản lý các thay đổi trong dự án.

Hiện nay có hai loại IDE chính, bao gồm **IDE một ngôn ngữ** và **IDE đa ngôn ngữ**.

Ngoài ra, còn có các loại IDE khác như:

+ IDE dựa trên web

+ IDE di động

1. **Các tính năng chính của IDE**

Ba tính năng phổ biến trong hầu hết các IDE là **trình soạn thảo mã nguồn**, **tự động hóa việc xây dựng** và **gỡ lỗi**. Các tính năng bổ sung có thể khác nhau và bao gồm:

- Cải tiến UI của trình soạn thảo mã

- Tính năng kiểm thử tự động

- Hỗ trợ triển khai mã thông qua tích hợp phần bổ trợ

- Hỗ trợ tái cấu trúc mã

- Hỗ trợ đóng gói ứng dụng

1. **IDE dành cho iOS:**

Có rất nhiều loại IDE dành cho iOS như Xcode, AppCode, Visual Studio Code,… nhưng

Xcode là IDE (Integrated Development Environment) được sử dụng rộng rãi để phát triển ứng dụng cho hệ điều hành iOS và macOS. Dưới đây là một số thông tin về Xcode:

1. **Xcode:**

◦ Xcode là môi trường phát triển tích hợp của Apple.

◦ Nó cho phép bạn viết mã, thiết kế giao diện và chạy ứng dụng trên máy ảo (Simulator) hoặc trên các thiết bị thật như iPhone, iPad, iPod và Apple Watch.

◦ Bạn có thể tải Xcode từ Mac App Store.

1. **Chức năng của Xcode:**

◦ Thiết kế giao diện người dùng bằng “Interface Builder”.

◦ Viết mã điều khiển bằng ngôn ngữ Swift.

◦ Chạy và kiểm tra ứng dụng.

◦ Gỡ lỗi với Xcode Debugger.

1. **Tính Năng Nổi Bật:**

◦ Tự động hoàn thành mã.

◦ Giao diện tương tác trực quan.

◦ Hỗ trợ Git để quản lý mã nguồn.

◦ Phân tích hiệu suất và tối ưu hóa ứng dụng.

1. **Tương Thích Với Swift:**

◦ Swift là ngôn ngữ lập trình của Apple, được sử dụng để phát triển ứng dụng cho iOS và macOS.

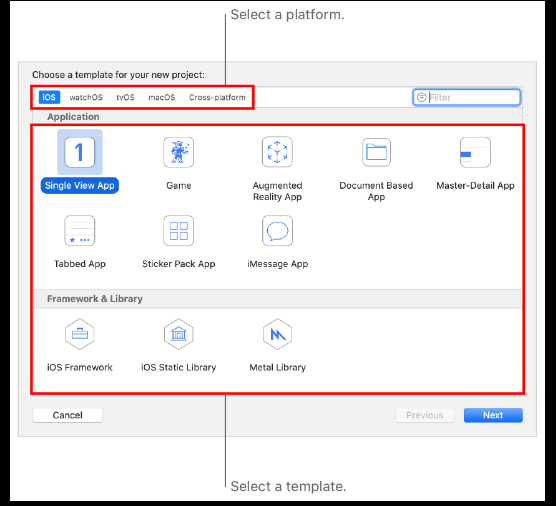
◦ Xcode hỗ trợ viết và kiểm tra mã Swift.

1. **Tạo, chạy và debug cho iOS**

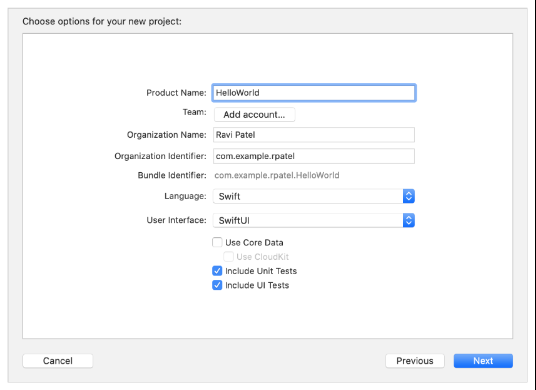
**Tạo:**

Để tạo một ứng dụng iOS, bạn cần sử dụng Xcode, IDE chính thức của Apple dành cho phát triển iOS. Sau khi tải xuống và cài đặt Xcode, bạn có thể tạo một dự án mới bằng cách khởi chạy Xcode, sau đó nhấp vào **Create a new Xcode project** hoặc chọn **File > New > Project.**

Trong bảng xuất hiện, chọn hệ điều hành hoặc platform và template thích hợp cho project (vd: iOS, macOS, v.v.).



Điền thông tin cơ bản như Product Name, Organization Name ( nếu không có thì hãy để tên của bản thân), ngôn ngữ lập trình (Swift hoặc Objective-C), v.v.



**Chạy:**

1. **Mở Project:**

◦ Mở Xcode.

◦ Chọn Project mà bạn muốn chạy từ danh sách các dự án đã tạo hoặc tạo một dự án mới bằng cách tạo New project như hướng dẫn ở trên.

1. **Chọn Thiết Bị Hoặc Máy Ảo:**

◦ Trong thanh công cụ, chọn thiết bị hoặc máy ảo mà bạn muốn chạy ứng dụng.

◦ Nếu bạn chưa có thiết bị hoặc muốn sử dụng máy ảo, bạn có thể chọn **Simulator**.

1. **Chạy Ứng Dụng:**

◦ Nhấn phím **Run** (hoặc sử dụng tổ hợp phím **Command + R**).

◦ Xcode sẽ biên dịch mã nguồn và chạy ứng dụng trên thiết bị hoặc máy ảo bạn đã chọn.

**Debug:**

Trong Xcode, **debugging** là quá trình tìm và sửa lỗi trong mã nguồn của bạn. Một trong những công cụ quan trọng để thực hiện việc này là **breakpoint**. Dưới đây là hướng dẫn cách sử dụng breakpoint trong Xcode:

1. **Tạo Breakpoint:**

◦ Đặt breakpoint bằng cách click vào số dòng code ở phía bên trái của khu vực code.

◦ Breakpoint sẽ tạo ra một điểm dừng, cho phép bạn ngưng việc thực thi mã nguồn tại vị trí đó.

1. **Chạy Ứng Dụng Với Breakpoint:**

◦ Nhấn phím Run (hoặc sử dụng tổ hợp phím Command + R).

◦ Khi ứng dụng chạy đến dòng code có breakpoint, nó sẽ dừng lại.

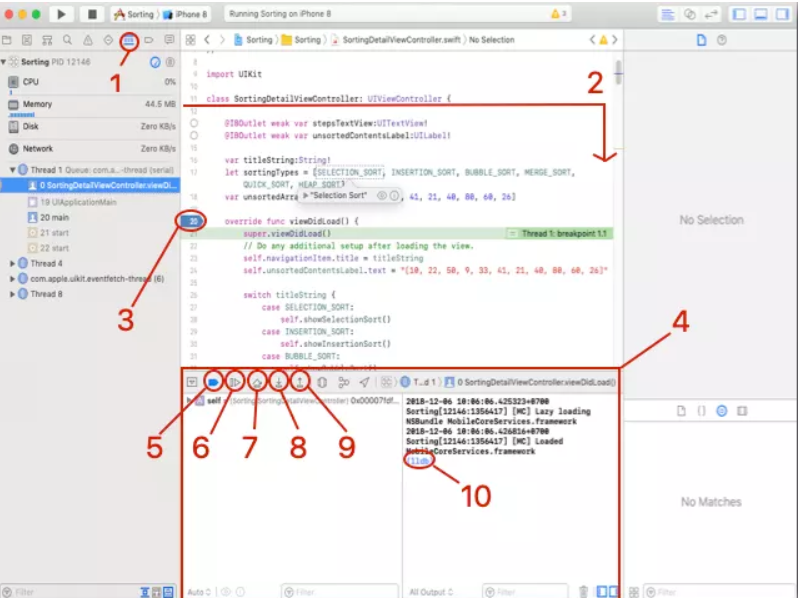
1. **Kiểm Tra Giá Trị Biến:**

◦ Trong khu vực Variables View, bạn có thể xem giá trị của các biến tại thời điểm breakpoint.

1. **Tiếp Tục Thực Thi Hoặc Bước Tiếp Theo:**

◦ Click vào biểu tượng Continue để tiếp tục thực thi mã nguồn.

◦ Sử dụng các phím tắt như Step Over (F6), Step Into (F7), và Step Out (F8) để điều hướng qua các dòng code.



**Ghi chú:**

1. Debug Navigator: hiển thị tất cả threads và breakpoints.
2. Thời điểm thread được tạo ra và xử lí code.
3. Có thể dễ dàng đặt breakpoint bằng các click vào số thứ tự của dòng code ở phía bên trái khu vực code
4. Khu vực debug.
5. Active và Deactive Breakpoints.
6. Tiếp tục hoặc dừng xử lí (keyboard shortcut: ⌃ + ⌘ + Y).
7. Step Over: thực thi dòng code tiếp theo (keyboard shortcut F6).
8. Step Into: đi vào bên trong function được gọi xem xét (keyboard shortcut F7).
9. Step Out: nhảy ra khỏi function (keyboard shortcut F8).
10. "lldb" nghĩa là "low level debugger"

Ngoài ra, còn một số công cụ dùng để **debug** như:

**Debug Bar:**

◦ Debug bar cung cấp các công cụ để debug ứng dụng:

◦ Step Over: Bước qua từng dòng code.

◦ Step Into: Bước vào một hàm.

◦ Step Out: Chạy đến dòng code tiếp theo.

◦ Pause: Tạm dừng chương trình.

◦ Continue: Tiếp tục chạy chương trình.

◦ Variables: Xem giá trị của các biến.

◦ Expression: Đánh giá biểu thức.

◦ Memory: Xem thông tin về bộ nhớ.

◦ Threads: Xem các thread đang chạy.

**Sử dụng LLDB:**

◦ LLDB là trình gỡ lỗi dòng lệnh tích hợp trong Xcode:

◦ Mở tab "Console" trong Xcode.

◦ Gõ lệnh LLDB để truy cập trình gỡ lỗi.

◦ Sử dụng các lệnh LLDB để kiểm tra và sửa lỗi.

**Xem Log:**

◦ Log cung cấp thông tin về hoạt động của chương trình:

◦ Xem log trong tab "Console".

◦ Lọc log theo mức độ (tất cả, lỗi, cảnh báo, etc.).

◦ Tìm kiếm trong log.

**\* Một số thủ thuật debug**:

◦ Sử dụng breakpoints để xác định vị trí lỗi.

◦ Sử dụng LLDB để kiểm tra giá trị biến và thay đổi giá trị biến.

◦ Xem log để tìm kiếm lỗi và cảnh báo.

◦ Sử dụng debugger để theo dõi luồng chương trình.

◦ Sử dụng các công cụ debug của Xcode để tìm kiếm và sửa lỗi.