



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

TÓM TẮT BÀI GIẢNG
KỸ THUẬT LẬP TRÌNH
(TÀI LIỆU LƯU HÀNH NỘI BỘ)

Đơn vị: Viện Kỹ thuật và Công nghệ

Nghệ An, 2022

NỘI DUNG HỌC PHẦN

- **CHƯƠNG 1** - Giới thiệu về máy tính và lập trình
- **CHƯƠNG 2** - Ngôn ngữ lập trình Python
- **CHƯƠNG 3** - Lập trình hàm
- **CHƯƠNG 4** - Các kiểu dữ liệu có cấu trúc
- **CHƯƠNG 5** - Thiết kế module
- **CHƯƠNG 6** - Lập trình hướng đối tượng
- **CHƯƠNG 7** - Thao tác trên tập tin và thư mục
- **CHƯƠNG 8** - Lập trình giao diện



TRƯỜNG ĐẠI HỌC VINH
VIỆN KỸ THUẬT VÀ CÔNG NGHỆ

CHƯƠNG 1

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH

Nghe An, 2022

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH



NỘI DUNG GIẢNG DẠY:

- 1.1. Phần cứng và phần mềm máy tính
- 1.2. Ngôn ngữ lập trình
- 1.3. Giải quyết vấn đề và phát triển phần mềm
- 1.4. Thuật toán

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH



NỘI DUNG GIẢNG DẠY:

1.1. Phần cứng và phần mềm máy tính

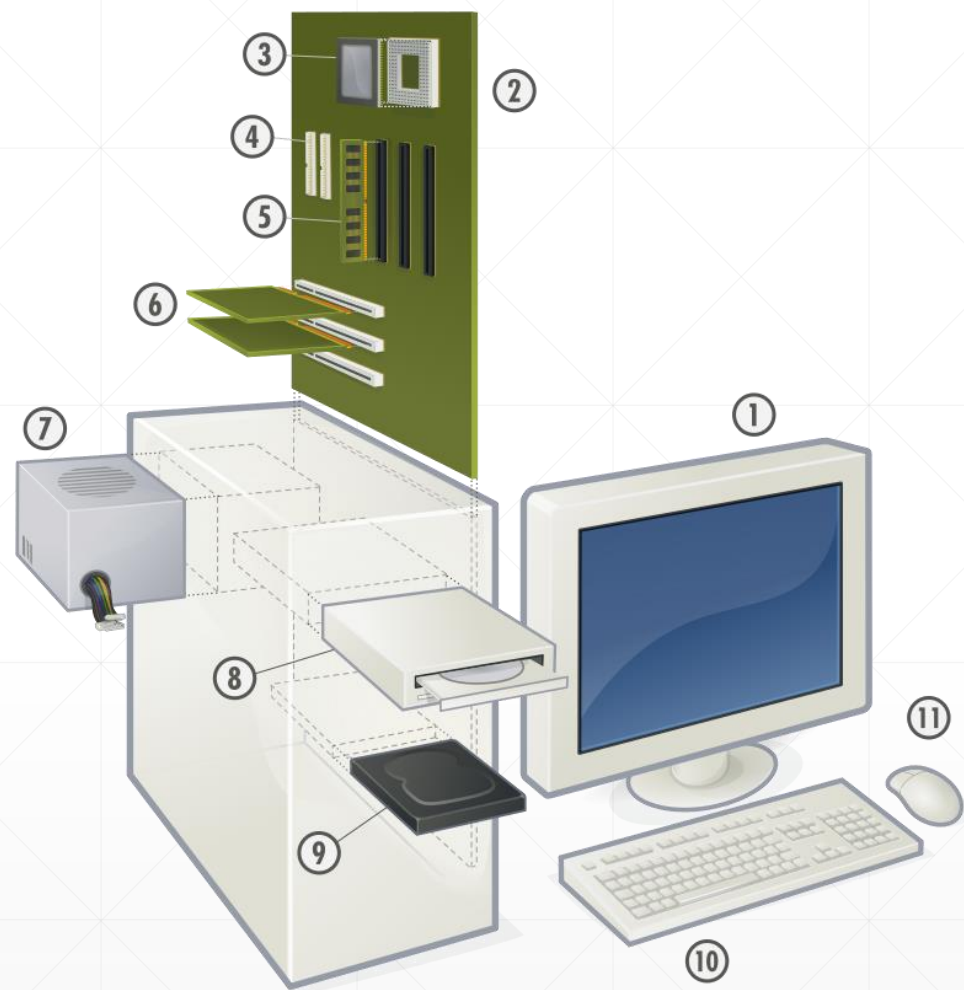
1.2. Ngôn ngữ lập trình

1.3. Giải quyết vấn đề và phát triển phần mềm

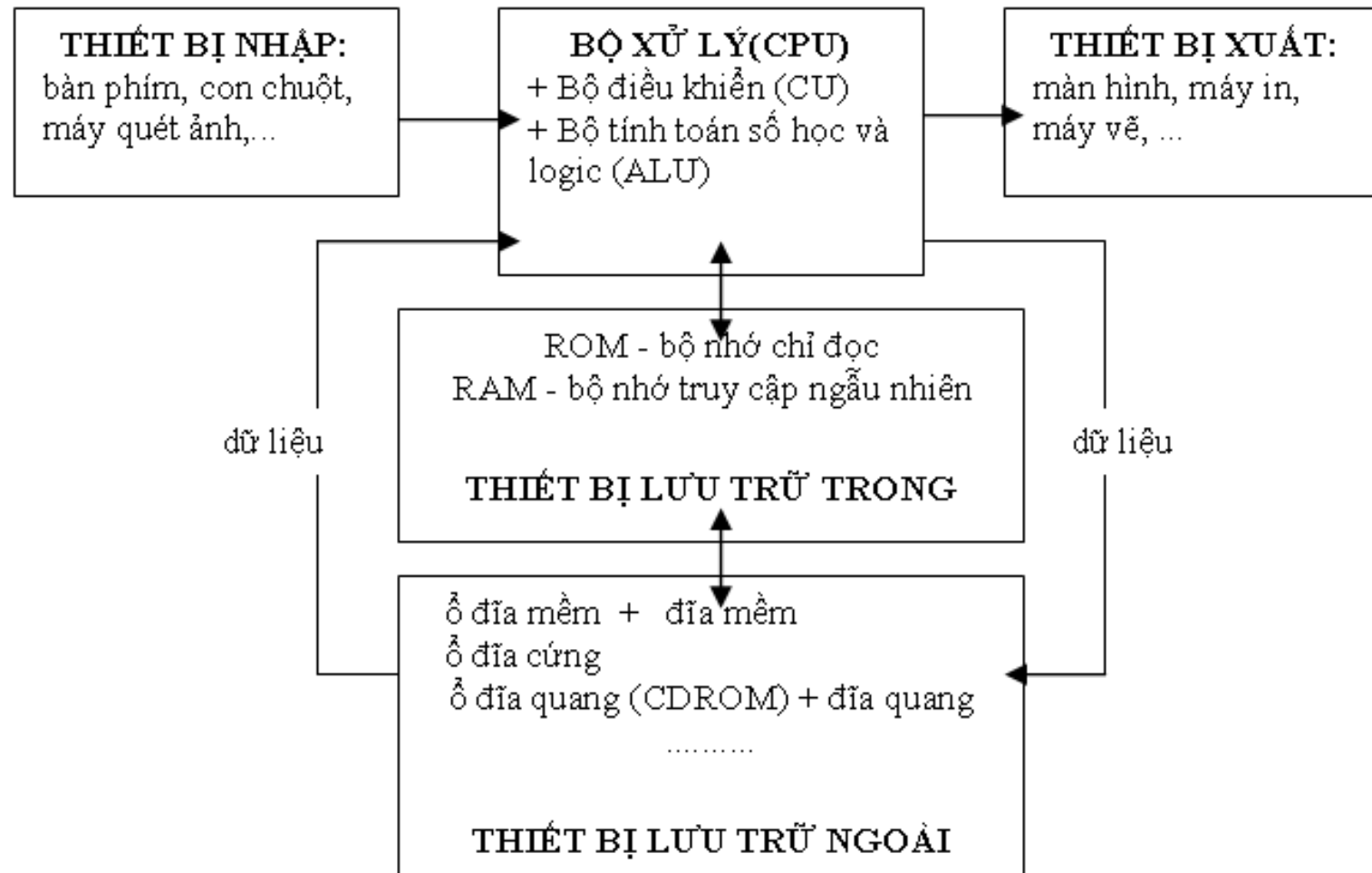
1.4. Thuật toán

PHẦN CỨNG VÀ PHẦN MỀM MÁY TÍNH

Phần cứng (Hardware): là các cơ phận (vật lý) cụ thể của máy tính hay hệ thống máy tính như là màn hình, chuột, bàn phím, máy in, máy quét, vỏ máy tính, bộ nguồn, bộ vi xử lý CPU, bo mạch chủ, các loại dây nối, loa, ổ đĩa mềm, ổ đĩa cứng, ổ CDROM, ổ DVD, card đồ họa VGA, card wifi, card âm thanh, bộ phận tản nhiệt Cooler,...



Các thành phần chính của máy tính cá nhân để bàn



Các thành phần chính của máy tính cá nhân để bàn

PHẦN CỨNG VÀ PHẦN MỀM MÁY TÍNH

Phần mềm máy tính (Computer Software) hay gọi tắt là **Phần mềm (Software)** là một tập hợp những câu lệnh hoặc chỉ thị (Instruction) được viết bằng một hoặc nhiều ngôn ngữ lập trình theo một trật tự xác định, và các dữ liệu hay tài liệu liên quan nhằm tự động thực hiện một số nhiệm vụ hay chức năng hoặc giải quyết một vấn đề cụ thể nào đó

Phần mềm thực hiện các chức năng của nó bằng cách gửi các chỉ thị trực tiếp đến phần cứng (hay phần cứng máy tính) hoặc bằng cách cung cấp dữ liệu để phục vụ các chương trình hay phần mềm khác.

PHẦN CỨNG VÀ PHẦN MỀM MÁY TÍNH

Phần mềm là một khái niệm trừu tượng, nó khác với phần cứng ở chỗ là "phần mềm không thể sờ hay đụng vào", và nó cần phải có phần cứng mới có thể thực thi được.

Chương trình máy tính thường được tạo ra bởi con người, những người này được gọi là **lập trình viên**, tuy nhiên cũng tồn tại những chương trình được sinh ra bởi các chương trình khác.

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH



NỘI DUNG GIẢNG DẠY:

1.1. Phần cứng và phần mềm máy tính

1.2. Ngôn ngữ lập trình

1.3. Giải quyết vấn đề và phát triển phần mềm

1.4. Thuật toán

NGÔN NGỮ LẬP TRÌNH

Ngôn ngữ lập trình là một tập con của ngôn ngữ máy tính, được thiết kế và chuẩn hóa để truyền các chỉ thị cho các máy có bộ xử lý (CPU), nói riêng là máy tính.

Ngôn ngữ lập trình được dùng để lập trình máy tính, tạo ra các chương trình máy nhằm mục đích điều khiển máy tính hoặc mô tả các thuật toán để người khác đọc hiểu.

NGÔN NGỮ LẬP TRÌNH

Một ngôn ngữ lập trình phải thỏa mãn được hai điều kiện cơ bản sau:

- Dễ hiểu và dễ sử dụng đối với người lập trình, để có thể dùng để giải quyết nhiều bài toán khác nhau.
- Miêu tả một cách đầy đủ và rõ ràng các tiến trình (tiếng Anh: process), để chạy được trên các hệ máy tính khác nhau.

NGÔN NGỮ LẬP TRÌNH

Trước đây, để tạo ra chương trình máy tính người ta phải làm việc trực tiếp với các con số 0 hoặc 1, hay còn gọi là ngôn ngữ máy. Công việc này vô cùng khó khăn, chiếm nhiều thời gian, công sức và đặc biệt dễ gây ra lỗi.

Để khắc phục nhược điểm này, người ta đề xuất ra hợp ngữ, một ngôn ngữ cho phép thay thế dãy 0 hoặc 1 này bởi các từ gợi nhớ. Tuy nhiên, cải tiến này vẫn còn chưa thật thích hợp với đa số người dùng.

Từ những năm 1950, người ta đã xây dựng những ngôn ngữ lập trình mà câu lệnh của nó gần với ngôn ngữ tự nhiên. Các ngôn ngữ này được gọi là ngôn ngữ lập trình bậc cao.

NGÔN NGỮ LẬP TRÌNH

Một tập hợp các chỉ thị được biểu thị qua ngôn ngữ lập trình nhằm mục đích thực hiện các thao tác máy tính nào đó được gọi là một **chương trình**. Khái niệm này còn có những tên khác như **chương trình máy tính** hay **chương trình điện toán**.

Khái niệm **lập trình** dùng để chỉ quá trình con người tạo ra chương trình máy tính thông qua ngôn ngữ lập trình. Người ta còn gọi đó là quá trình **mã hoá thông tin** tự nhiên thành ngôn ngữ máy.

Văn bản được viết bằng ngôn ngữ lập trình để tạo nên chương trình được gọi là **mã nguồn**.

NGÔN NGỮ LẬP TRÌNH

Thao tác chuyển đổi từ **mã nguồn** thành **chuỗi các chỉ thị máy tính** được thực hiện tương tự như việc chuyển đổi qua lại giữa các ngôn ngữ tự nhiên của con người.

Các thao tác này gọi là biên dịch, hay ngắn gọn hơn là dịch:

- Nếu quá trình dịch diễn ra đồng thời với quá trình thực thi, ta gọi đó là thông dịch.
- Nếu diễn ra trước, ta gọi đó là biên dịch.

NGÔN NGỮ LẬP TRÌNH

ĐẶC ĐIỂM CHUNG CỦA NGÔN NGỮ LẬP TRÌNH:

Mỗi ngôn ngữ lập trình có thể được xem như là một tập hợp của các chi tiết kỹ thuật chú trọng đến cú pháp, từ vựng, và ý nghĩa của ngôn ngữ. Những chi tiết kỹ thuật này thường bao gồm:

- Dữ liệu và cấu trúc dữ liệu;
- Câu lệnh và dòng điều khiển;
- Các tên và các tham số;
- Các cơ chế tham khảo và sự tái sử dụng.

NGÔN NGỮ LẬP TRÌNH

Đối với các ngôn ngữ phổ biến hoặc có lịch sử lâu dài, người ta thường tổ chức các hội thảo chuẩn hoá nhằm tạo ra và công bố các tiêu chuẩn chính thức cho ngôn ngữ đó, cũng như thảo luận về việc mở rộng, bổ sung cho các tiêu chuẩn trước đó.

Ví dụ: Với ngôn ngữ C++, hội đồng tiêu chuẩn ANSI C++ và ISO C++ đã tổ chức đến 13 cuộc hội thảo để điều chỉnh và nâng cấp ngôn ngữ này.

THẢO LUẬN NHÓM

NỘI DUNG:

1. Phân biệt giữa ngôn ngữ máy, hợp ngữ và ngôn ngữ lập trình cấp cao.
2. Giới thiệu về một số ngôn ngữ lập trình thông dụng hiện nay về các nội dung:
 - Lịch sử phát triển;
 - Đặc điểm;
 - Ứng dụng.

BÀI TẬP

NỘI DUNG:

1. Tạo tài khoản trên trang web <https://github.com/>.
2. Tải các tài liệu của học phần đã được giới thiệu.

CHUẨN BỊ CHO BUỔI HỌC TIẾP THEO:

1. Đọc các tài liệu về nội dung mục 1.3 và 1.4.
2. Nghiên cứu về khái niệm **mã giả**, cách sử dụng **mã giả**.

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH



NỘI DUNG GIẢNG DẠY:

1.1. Phần cứng và phần mềm máy tính

1.2. Ngôn ngữ lập trình

1.3. Giải quyết vấn đề và phát triển phần mềm

1.4. Thuật toán

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



CÁC MÔ HÌNH PHÁT TRIỂN SẢN PHẨM PHẦN MỀM

- Quá trình phát triển phần mềm là tập hợp các thao tác và các kết quả tương quan để sản xuất ra một sản phẩm phần mềm.
- Hầu hết các thao tác này được tiến hành bởi các kỹ sư phần mềm.
- Các công cụ hỗ trợ máy tính về kỹ thuật phần mềm có thể được dùng để giúp trong một số thao tác.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



Có 4 thao tác nền tảng của hầu hết các quá trình phần mềm:

- + Đặc tả phần mềm: Các chức năng của phần mềm và điều kiện để nó hoạt động phải được định nghĩa.
- + Phát triển phần mềm: Để phần mềm đạt được đặc tả thì phải có quá trình phát triển này.
- + Đánh giá phần mềm: Phần mềm phải được đánh giá để chắc chắn rằng nó làm những gì mà khách hàng muốn.
- + Sự tiến hóa của phần mềm: Phần mềm phải tiến hóa để thỏa mãn sự thay đổi các yêu cầu của khách hàng.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



Có nhiều loại mô hình phát triển phần mềm khác nhau:

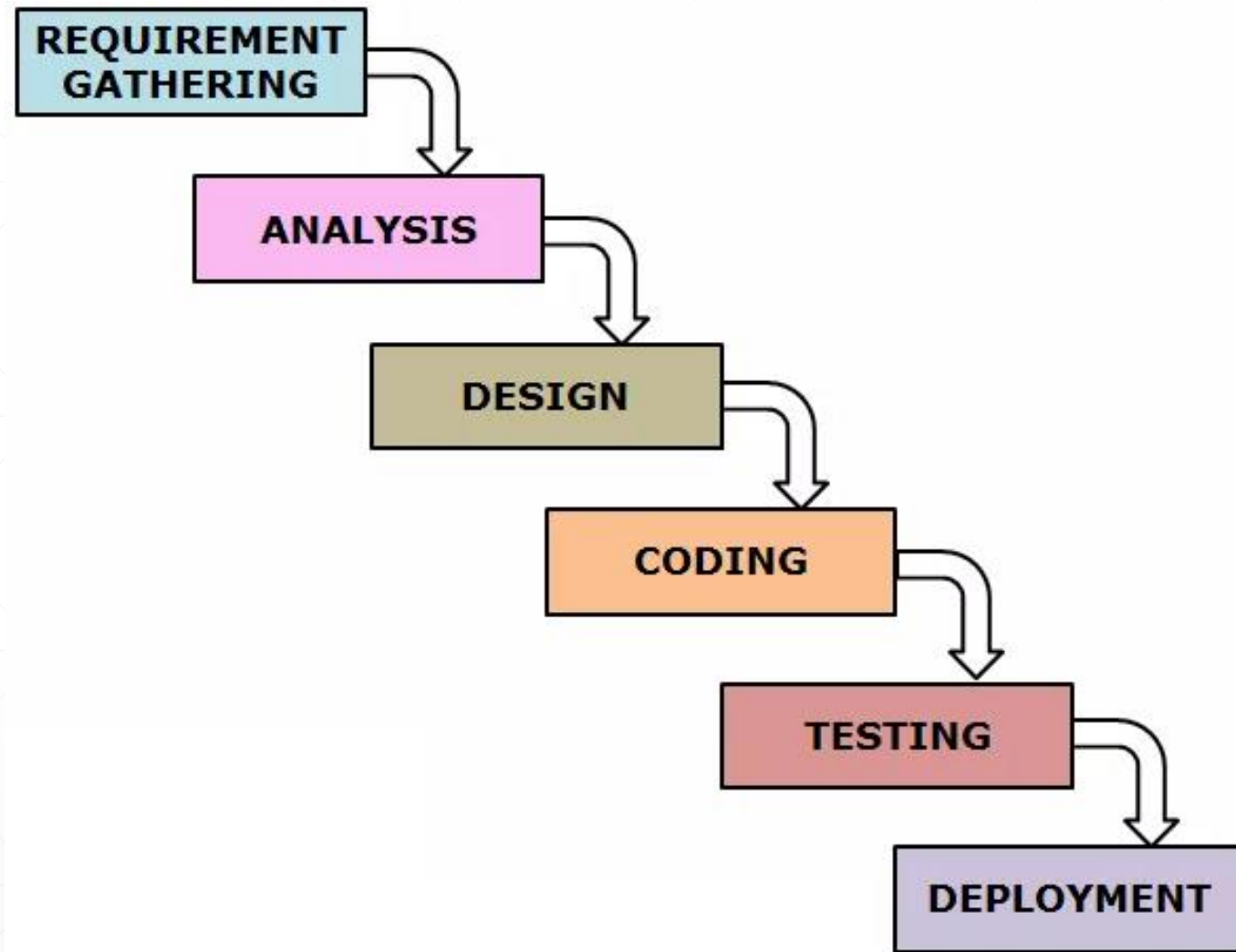
- Mô hình thác nước (Waterfall model)
- Mô hình xoắn ốc (Spiral model)
- Mô hình agile
- Mô hình tiếp cận lặp (Iterative model)
- Mô hình tăng trưởng (Incremental model)
- Mô hình chữ V (V model)
- Mô hình Scrum
- RAD model (Rapid Application Development)

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH THÁC NƯỚC (WATERFALL MODEL):

- Đây được coi như là mô hình phát triển phần mềm đầu tiên được sử dụng.
- Mô hình này áp dụng tuần tự các giai đoạn của phát triển phần mềm.
- Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau. Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. Đặc biệt không được quay lại giai đoạn trước để xử lý các yêu cầu khi muốn thay đổi.



Mô hình thác nước (Waterfall model)

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH THÁC NƯỚC (WATERFALL MODEL):

- **Ứng dụng**

Mô hình thường được áp dụng cho các dự án phần mềm như sau:

- Các dự án nhỏ, ngắn hạn.
- Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH THÁC NƯỚC (WATERFALL MODEL):

- **Ưu điểm**

- Dễ sử dụng, dễ tiếp cận, dễ quản lý.
- Sản phẩm phát triển theo các giai đoạn được xác định rõ ràng.
- Xác nhận ở từng giai đoạn, đảm bảo phát hiện sớm các lỗi.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH THÁC NƯỚC (WATERFALL MODEL):

• Nhược điểm

- Ít linh hoạt, phạm vi điều chỉnh hạn chế.
- Rất khó để đo lường sự phát triển trong từng giai đoạn.
- Mô hình không thích hợp với những dự án dài, đang diễn ra, hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.
- Khó quay lại khi giai đoạn nào đó đã kết thúc.

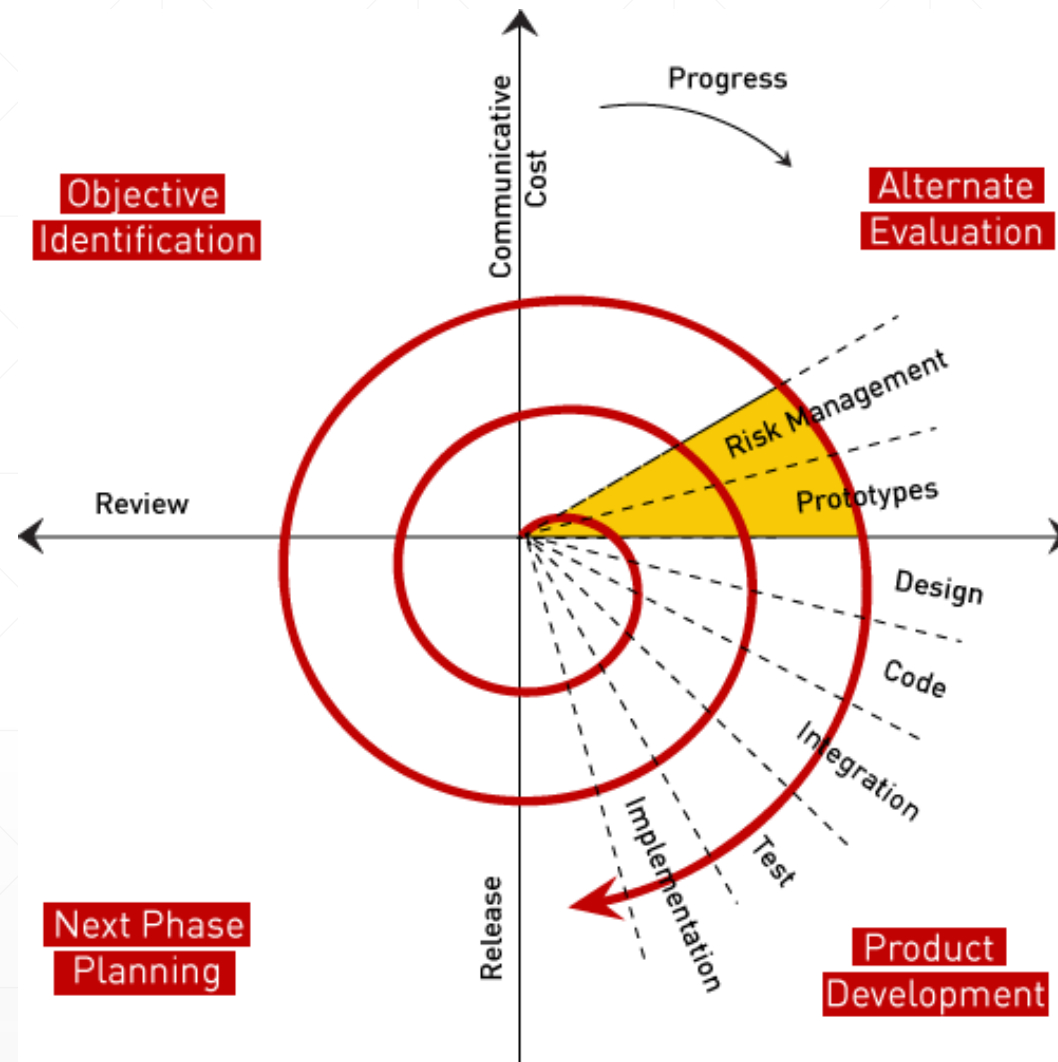
GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM

MÔ HÌNH XOẮN ỐC (SPIRAL MODEL):

Là mô hình kết hợp giữa các tính năng của mô hình prototyping và mô hình thác nước.

Mô hình xoắn ốc được ưa chuộng cho các dự án lớn, đắt tiền và phức tạp.

Mô hình này sử dụng những giai đoạn tương tự như mô hình thác nước, về thứ tự, plan, đánh giá rủi ro, ...



Mô hình xoắn ốc (Spiral model)

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH XOẮN ỐC (SPIRAL MODEL):

Ứng dụng

Mô hình này thường được sử dụng cho các ứng dụng lớn và các hệ thống được xây dựng theo các giai đoạn nhỏ hoặc theo các phân đoạn.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH XOẮN ỐC (SPIRAL MODEL):

Ưu điểm

- Tốt cho các hệ phần mềm quy mô lớn.
- Dễ kiểm soát các mạo hiểm ở từng mức tiến hóa.
- Đánh giá thực tế hơn như là một quy trình làm việc, bởi vì những vấn đề quan trọng đã được phát hiện sớm hơn.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH XOẮN ỐC (SPIRAL MODEL):

Nhược điểm

- Manager cần có kỹ năng tốt để quản lý dự án, đánh giá rủi ro kịp thời.
- Chi phí cao và mất nhiều thời gian để hoàn thành dự án.
- Phức tạp và không thích hợp với các dự án nhỏ và ít rủi ro.
- Yêu cầu thay đổi thường xuyên dẫn đến lặp vô hạn.
- Chưa được dùng rộng rãi.

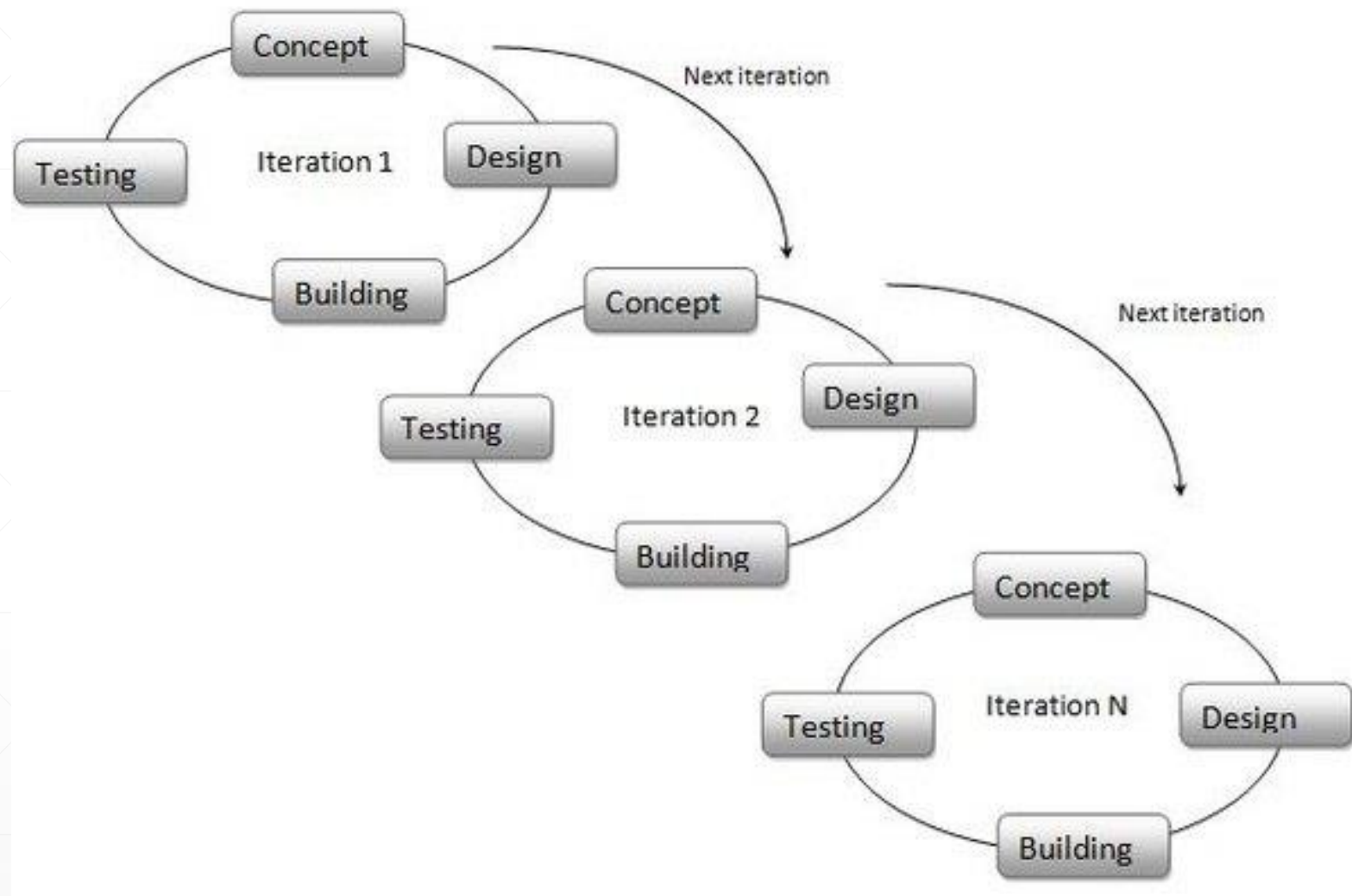
GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH AGILE:

Agile là một phương pháp phát triển phần mềm linh hoạt để làm sao đưa sản phẩm đến tay người dùng càng nhanh càng tốt và được xem như là sự cải tiến so với những mô hình cũ như mô hình “Thác nước (waterfall)” hay “CMMI”.

Phương thức phát triển phần mềm Agile là một tập hợp các phương thức phát triển lặp và tăng dần trong đó các yêu cầu và giải pháp được phát triển thông qua sự liên kết cộng tác giữa các nhóm tự quản và liên chức năng.



Mô hình Agile

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH AGILE:

Ứng dụng

Có thể được sử dụng với bất kỳ loại hình dự án nào, nhưng cần sự tham gia và tính tương tác của khách hàng.

Sử dụng khi khách hàng yêu cầu chức năng sẵn sàng trong khoảng thời gian ngắn.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH AGILE:

Ưu điểm

- Tăng cường tinh thần làm việc nhóm và trao đổi công việc hiệu quả.
- Các chức năng được xây dựng nhanh chóng và rõ ràng, dễ quản lý.
- Dễ dàng bổ sung, thay đổi yêu cầu.
- Quy tắc tối thiểu, tài liệu dễ hiểu, dễ sử dụng.

GIẢI QUYẾT VẤN ĐỀ VÀ PHÁT TRIỂN PHẦN MỀM



MÔ HÌNH AGILE:

Nhược điểm

- Không thích hợp để xử lý các phụ thuộc phức tạp.
- Có nhiều rủi ro về tính bền vững, khả năng bảo trì và khả năng mở rộng.
- Cần một team có kinh nghiệm.
- Phụ thuộc rất nhiều vào sự tương tác rõ ràng của khách hàng.
- Chuyển giao công nghệ cho các thành viên mới trong nhóm có thể khá khó khăn do thiếu tài liệu.

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH

NỘI DUNG GIẢNG DẠY:

- 1.1. Phần cứng và phần mềm máy tính
- 1.2. Ngôn ngữ lập trình
- 1.3. Giải quyết vấn đề và phát triển phần mềm
- 1.4. Thuật toán**

THUẬT TOÁN

Thuật toán, còn gọi là **giải thuật**, là một tập hợp hữu hạn của các chỉ thị hay phương cách được định nghĩa rõ ràng cho việc hoàn tất một số sự việc từ một trạng thái ban đầu cho trước; khi các chỉ thị này được áp dụng triệt để thì sẽ dẫn đến kết quả sau cùng như đã dự đoán.

Nói cách khác, thuật toán là một bộ các qui tắc hay qui trình cụ thể nhằm giải quyết một vấn đề trong một số bước hữu hạn, hoặc nhằm cung cấp một kết quả từ một tập hợp của các dữ kiện đưa vào.

THUẬT TOÁN

Ví dụ: Thuật toán để giải phương trình bậc nhất $P(x): ax + b = c$, (a, b, c là các số thực), trong tập hợp các số thực có thể là một bộ các bước sau đây:

1. Nếu $a = 0$

- $b = c$ thì $P(x)$ có nghiệm bất kì
- $b \neq c$ thì $P(x)$ vô nghiệm

2. Nếu $a \neq 0$

- $P(x)$ có duy nhất một nghiệm $x = (c - b)/a$

THUẬT TOÁN

Lưu ý:

Khi một thuật toán đã hình thành thì ta không xét đến việc chứng minh thuật toán đó mà chỉ chú trọng đến việc áp dụng các bước theo sự hướng dẫn sẽ có kết quả đúng.

Việc chứng minh tính đầy đủ và tính đúng của các thuật toán phải được tiến hành xong trước khi có thuật toán. Nói rõ hơn, thuật toán có thể chỉ là việc áp dụng các công thức hay qui tắc, qui trình đã được công nhận là đúng hay đã được chứng minh về mặt toán học.

THUẬT TOÁN

"Thuật toán" hiện nay thường được dùng để chỉ thuật toán giải quyết các vấn đề tin học. Hầu hết các thuật toán tin học đều có thể viết thành các chương trình máy tính mặc dù chúng thường có một vài hạn chế (khả năng của máy tính, khả năng của người lập trình).

Trong nhiều trường hợp, một chương trình khi thiết kế bị thất bại là do lỗi ở các thuật toán mà người lập trình đưa vào là không chính xác, không đầy đủ, hay không ước định được trọn vẹn lời giải vấn đề.

Tuy nhiên cũng có một số bài toán mà hiện nay người ta chưa tìm được lời giải triệt để, những bài toán ấy gọi là những bài toán NP - không đầy đủ.

THUẬT TOÁN

Tính chất của thuật toán:

- Tính chính xác: để đảm bảo kết quả tính toán hay các thao tác mà máy tính thực hiện được là chính xác.
- Tính rõ ràng: Thuật toán phải được thể hiện bằng các câu lệnh minh bạch; các câu lệnh được sắp xếp theo thứ tự nhất định.
- Tính khách quan: Một thuật toán dù được viết bởi nhiều người trên nhiều máy tính vẫn phải cho kết quả như nhau.

THUẬT TOÁN

- Tính phổ dụng: Thuật toán không chỉ áp dụng cho một bài toán nhất định mà có thể áp dụng cho một lớp các bài toán có đầu vào tương tự nhau.
- Tính kết thúc: Thuật toán phải gồm một số hữu hạn các bước tính toán.

THUẬT TOÁN

Độ phức tạp thuật toán:

Thời gian mà máy tính khi thực hiện một thuật toán không chỉ phụ thuộc vào bản thân thuật toán đó, ngoài ra còn tùy thuộc từng máy tính. Để đánh giá hiệu quả của một thuật toán, có thể xét số các phép tính phải thực hiện khi thực hiện thuật toán này.

Thông thường số các phép tính được thực hiện phụ thuộc vào cỡ của bài toán, tức là độ lớn của đầu vào. Vì thế độ phức tạp thuật toán là một hàm phụ thuộc đầu vào. Tuy nhiên trong những ứng dụng thực tiễn, chúng ta không cần biết chính xác hàm này mà chỉ cần biết một ước lượng đủ tốt của chúng.

Chương 1:

GIỚI THIỆU VỀ MÁY TÍNH VÀ LẬP TRÌNH



TÓM TẮT

1.1. Phần cứng và phần mềm máy tính

1.2. Ngôn ngữ lập trình

1.3. Giải quyết vấn đề và phát triển phần mềm

1.4. Thuật toán

THẢO LUẬN NHÓM

NỘI DUNG:

1. Xác định vai trò của thuật toán trong lập trình.
2. Khái niệm về mã giả và cách sử dụng mã giả;
3. Ứng dụng viết mã giả để giải quyết các vấn đề:
 - Giải phương trình bậc nhất;
 - Giải phương trình bậc 2.

BÀI TẬP



NỘI DUNG:

1. Tìm hiểu về phần mềm Flowgorithm.
 2. Tải và cài đặt phần mềm: Flowgorithm.
 3. Sử dụng phần mềm Flowgorithm để thiết kế các thuật toán:
 - Giải phương trình bậc nhất;
 - Giải phương trình bậc 2;
 - Giải thuật tìm dãy số Fibonacci.
- Báo cáo kết quả thực hiện lên github.com.

BÀI TẬP

CHUẨN BỊ CHO BUỔI HỌC TIẾP THEO:

1. Đọc các tài liệu về nội dung mục 2.1; 2.2 và 2.3.
2. Nghiên cứu về ngôn ngữ lập trình Python và đặc điểm của ngôn ngữ lập trình Python.