NHẬP MÔN
TƯ DUY TÍNH TOÁN

Bài 1: Giới thiệu môn học

Nội dung

1. Thông tin chung về môn học
2. “Tư duy tính toán” là gì?
3. Thuật toán
   - Sơ đồ khối
   - Lưu đồ Nassi–Shneiderman
   - Mã giả
4. Bài tập và thảo luận

Phần 1

Thông tin chung về môn học

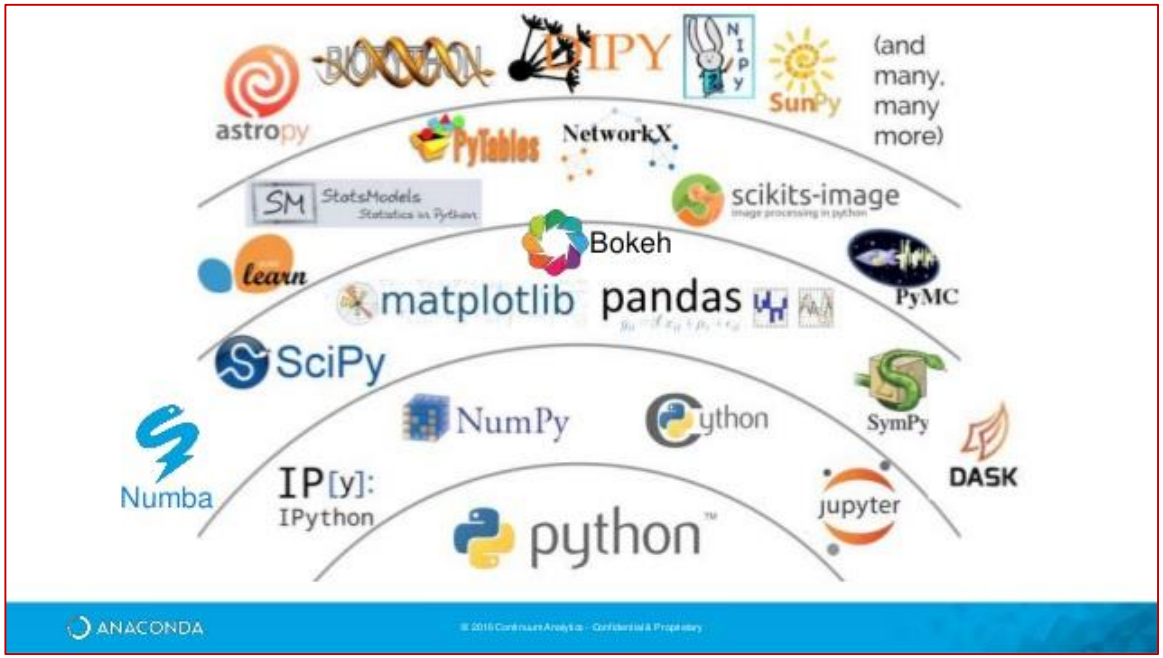
Giới thiệu môn học

- Tên môn: Nhập môn Tư duy Tính toán (Introduction to Computational Thinking)
- Số tín chỉ: 2 (15 tiết lý thuyết + 15 tiết bài tập)
- Nội dung chính:
  - Thế nào là "tư duy tính toán"
  - Một số cách biểu diễn thuật toán
  - Cơ bản về ngôn ngữ lập trình python
  - Một số thuật toán đơn giản sử dụng python
- Giảng viên: Trương Xuân Nam, khoa CNTT
- Email: namtx@wru.vn / truongxuannam@gmail.com

Tài liệu môn học và phần mềm học tập

- Tài liệu chính: bài giảng của giáo viên
  - Sách giáo trình đang được biên soạn, hiện chưa có
- Phần mềm học tập: python 3.x
  - Có thể sử dụng bất kỳ phần mềm nào, miễn là nó hỗ trợ ngôn ngữ python 3.5 trở lên
  - Trên lớp, thầy giáo sẽ minh họa bằng phần mềm tiêu chuẩn, lấy từ site https://www.python.org
- Bài giảng, bài tập, mã nguồn, điểm số,… sẽ được đưa lên site https://txnam.net mục BÀI GIẢNG
  - Bài giảng và bài tập sẽ được đưa lên trước giờ học
  - Trong giờ thực hành, sinh viên vào website lấy bài tập về để làm, giáo viên sẽ không gửi cho lớp
  - Điểm quá trình cũng sẽ được công bố trên website

Phần mềm học tập



Kiến thức yêu cầu

- Đã biết và sử dụng được một ngôn ngữ lập trình nào đó (C/C++, C#, Java, Javascript, Pascal,…)
  - Vì chúng ta sẽ học khá nhanh, nhiều kiến thức
  - Sử dụng được tức là có thể viết chương trình với ngôn ngữ đó
- Có kiến thức về các khái niệm cơ bản trong lập trình
  - Môn học này giúp sinh viên hiểu hơn về những khái niệm đó
- Biết sử dụng email
  - Nộp bài tập vào email của thầy giáo: cần ghi rõ tên sinh viên, bài nộp là bài nào, của buổi bài tập số mấy
  - Có thể email cho thầy giáo để hỏi thêm các vấn đề về môn học

- Chú ý: copy bài của bạn khác để nộp sẽ bị cấm thi

Đánh giá kết quả

- Điểm môn học:
  - Điểm quá trình: **50%**
  - Điểm thi cuối kỳ: **50%**
- Điểm quá trình:
  - Điểm danh
  - Bài làm trên lớp, trong phòng lab
  - Bài tập về nhà (nộp qua email)
- Thi cuối kỳ:
  - Thi trắc nghiệm
  - Học gì thi nấy, không hỏi ngoài môn học
  - Không có giới hạn nội dung thi
  - Không sử dụng tài liệu tham khảo

Mục tiêu của môn học này

- Làm quen với cách các thuật toán có thể ứng dụng vào bài toán thực tế như thế nào
  - Mô tả bài toán theo cách của dân máy tính
  - Lựa chọn phương pháp xử lý phù hợp
- Biết cài đặt thuật giải bằng ngôn ngữ lập trình (python)
- Rèn luyện thói quen lập trình một cách “trong sáng” ;)
- Học lập trình python phần cơ bản, để có thể sử dụng trong các môn học sau này

Lời khuyên

ĐI HỌC ĐẦY ĐỦ

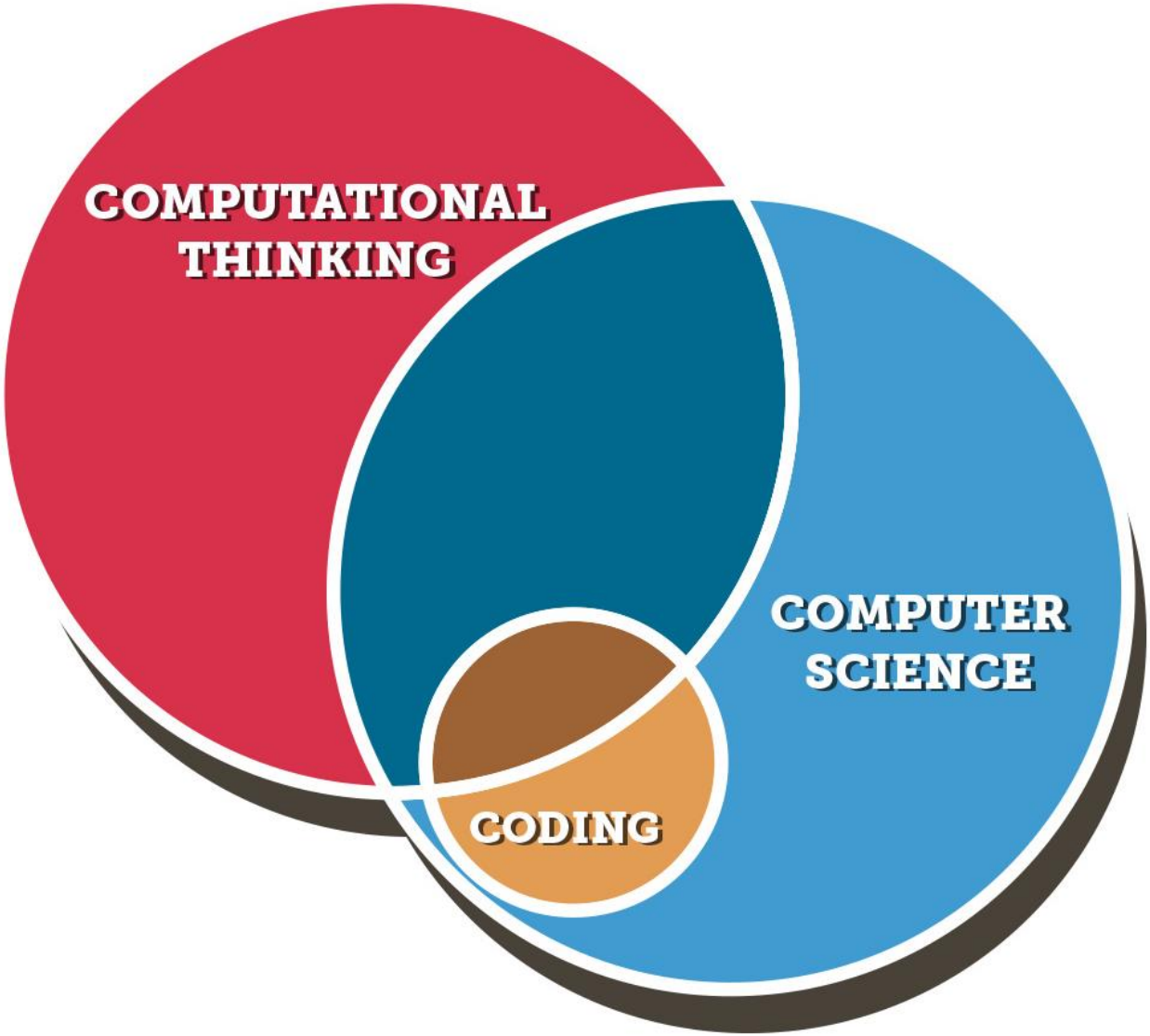
LÀM HẾT TẤT CẢ CÁC BÀI TẬP

Chỉ thế thôi!!!

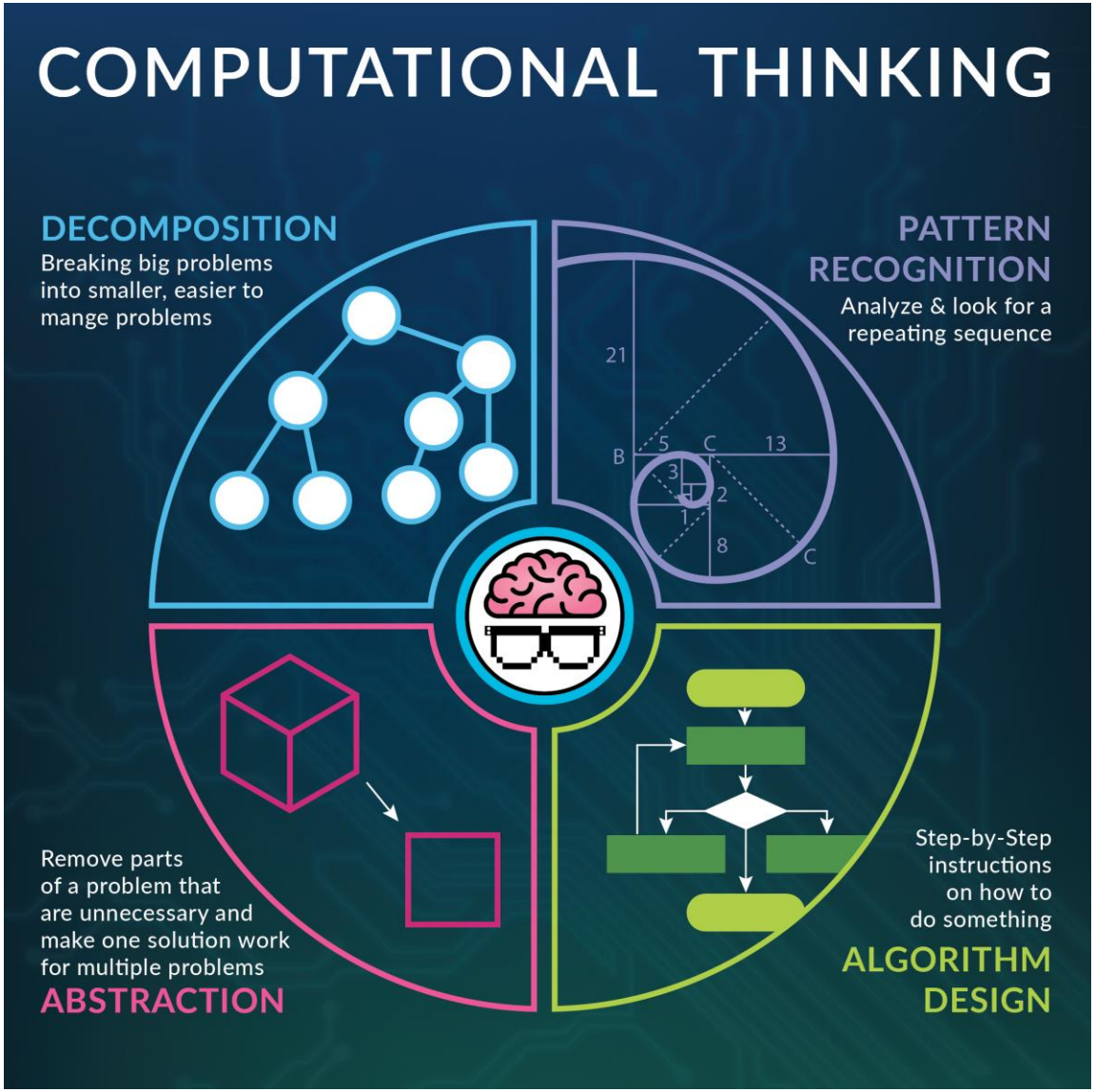
Phần 2

“Tư duy tính toán” là gì?

“Tư duy tính toán” là gì?



“Tư duy tính toán” là gì?



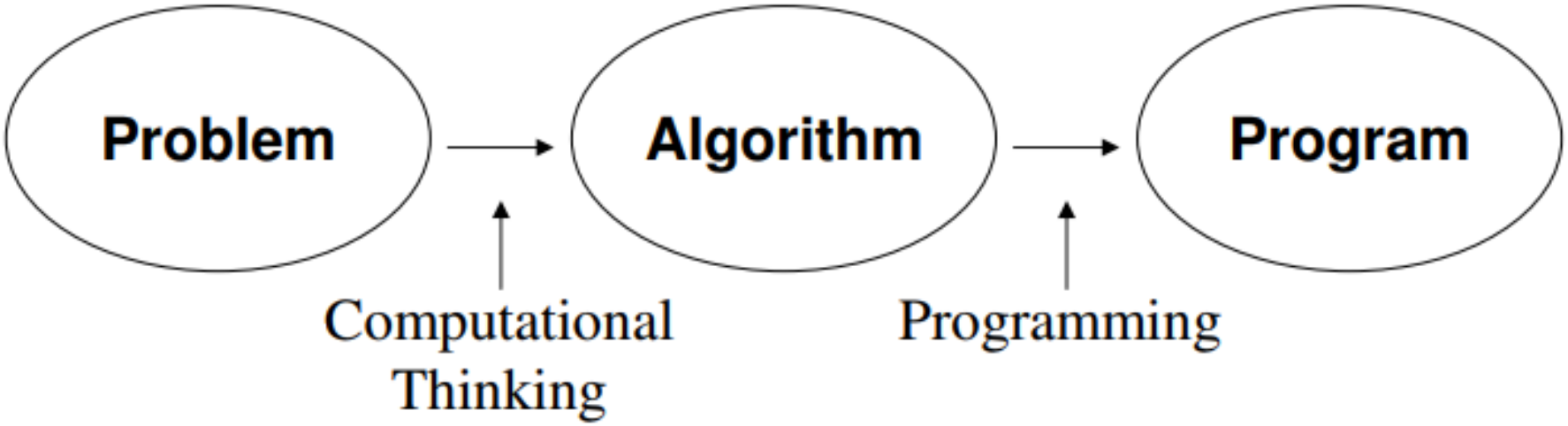
“Tư duy tính toán” là gì?

- “Tư duy”: có nhiều loại “tư duy” khác nhau
- “Tư duy tính toán”: duy phù hợp với người làm CNTT
- “Quy trình xử lý” khi đối mặt với một vấn đề cần được giải quyết bằng máy tính
  - **Decomposition** (phân rã): chia nhỏ bài toán
  - **Pattern recognition** (nhận dạng mẫu): phân tích và phân loại vấn đề thành nhóm, áp dụng các phương pháp phù hợp cho từng nhóm
  - **Abstraction** (trừu tượng hóa): tạo ra lời giải chung cho nhiều vấn đề tương tự nhau
  - **Algorithm design** (thiết kế thuật giải): xây dựng lời giải bằng máy tính

“Tư duy tính toán” với ngành CNTT

- Sự thể hiện của “tư duy tính toán” thường xuất hiện trong quá trình xây dựng lời giải (problem solving), đặc biệt là đối với ngành lập trình
  - Phân tích và mô hình hóa các vấn đề cần giải quyết cùng với dữ liệu của chúng
  - Chia nhỏ bài toán đó thành các bài toán con thông dụng đã được giải quyết
  - Tổ hợp các lời giải đó thành một giải pháp hoàn chỉnh
  - Hiểu một cách rõ ràng, chi tiết cách máy tính xử lý các dữ liệu của bài toán như thế nào
  - Phân tích được tính hiệu quả của lời giải
  - Chỉ ra được điểm yếu, rủi ro với lời giải hiện tại
  - …

“Tư duy tính toán” với ngành CNTT



Ví dụ

- Câu đố: tìm 9 chữ số cuối cùng của 2018!

- "Bài này khó, bỏ luôn"
  - "không có tư duy" ☹
- "2018! chia hết cho 1 tỉ, vậy 9 chữ số cuối là 9 chữ số 0"
  - Tư duy logic, nhưng lời giải không tổng quát
- "Dùng máy tính tính xem 2018! bằng bao nhiêu"
  - Tư duy tính toán, nhưng thiếu hiệu quả và không hiểu rõ về cách thức máy tính thực hiện công việc

- Câu hỏi: muốn tính 9 chữ số đầu tiên của 2018! thì sao?

Phần 3

Thuật toán

Thuật toán

- Thuật toán: các bước để giải quyết một vấn đề
- Đặc trưng quan trọng:
  - Tính máy móc: không nhập nhằng, không thể hiểu theo nhiều cách khác nhau
  - Tính dừng: sau một số bước hữu hạn sẽ cho ra kết quả
  - Tính tổng quát: giải quyết một lớp các vấn đề chứ không phải một vấn đề cụ thể
  - Có đầu vào / đầu ra xác định: dữ liệu đầu vào và đầu ra rõ ràng
  - Tính đúng đắn: hoạt động ổn định, kết quả không bị phụ thuộc bất kỳ yếu tố nào khác ngoại trừ đầu vào
- Như vậy chúng ta thấy có nhiều thứ tương tự thuật toán, nhưng không phải là thuật toán (ví dụ: công thức nấu ăn)

Mô tả thuật toán

- Giải phương trình $ax^2 + bx + c = 0$ với điều kiện $a \neq 0$

  Bước 1: nhập các giá trị a, b, c

  Bước 2: nếu a = 0 thì thông báo lỗi và dừng

  Bước 3: tính $d = b^2 - 4 \times a \times c$

  Bước 4: nếu d < 0 thì thông báo vô nghiệm và dừng

  Bước 5: nếu d > 0 thì thực hiện Bước 7

  Bước 6: thông báo có nghiệm kép $x = \frac{-b}{2a}$ và dừng

  Bước 7: thông báo có hai nghiệm

  Bước 8: tính và in ra nghiệm $x_1 = \frac{-b+\sqrt{d}}{2a}$

  Bước 9: tính và in ra nghiệm $x_2 = \frac{-b-\sqrt{d}}{2a}$

Mô tả thuật toán

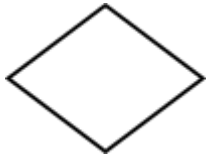
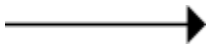
- Mô tả thuật toán đơn giản nhất bằng ngôn ngữ tự nhiên
  - Mô tả từng bước một
  - Có thể xuất hiện rẽ nhánh (lựa chọn một trong nhiều việc)
  - Có thể xuất hiện lặp (làm một việc cho đến khi đạt điều kiện)
- Cách làm này dễ hiểu (tương tự như công thức nấu ăn), nhưng thường không đủ chặt chẽ
- Một vài phương pháp mô tả thuật toán thông dụng
  - Sơ đồ khối
  - Lưu đồ Nassi–Shneiderman
  - Mã giả
- Mỗi phương pháp có điểm hay dở riêng
- Không có cách mô tả thuật toán hoàn toàn tránh được lỗi

Phần 3.1

Sơ đồ khối

Sơ đồ khối

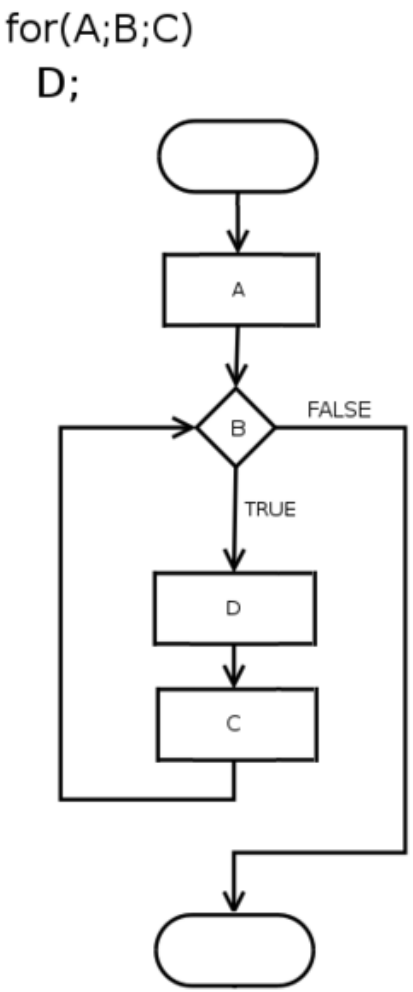
- Sử dụng hệ thống kí hiệu

	Mở đầu / Kết thúc
	Khối xử lý
	Rẽ nhánh (đúng / sai)
	Nhập / xuất dữ liệu
	Bước chuyển

Sơ đồ khối

- Luôn mở đầu và kết thúc bởi khối cạnh tròn
- Khối có thể có nhiều bước chuyển vào
- Khối quả trám luôn có 2 bước ra
  - Ứng với hai tình huống đúng / sai
- Các khối khác chỉ có 1 bước ra duy nhất

- Ưu điểm:
  - Đơn giản, dễ kiểm tra
  - Phù hợp với bài toán đơn giản
  - Chương trình phức tạp sẽ có sơ đồ rất lớn

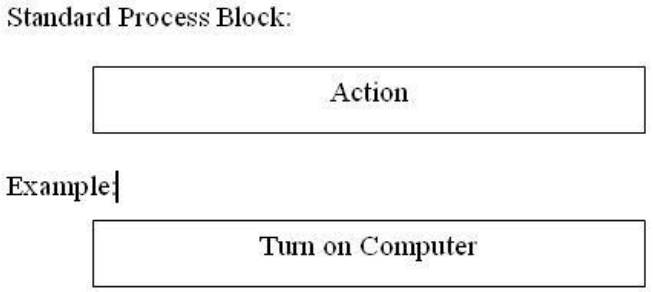


Phần 3.2

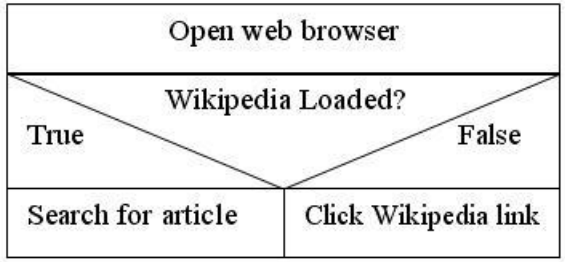
Lưu đồ Nassi–Shneiderman

Lưu đồ Nassi–Shneiderman

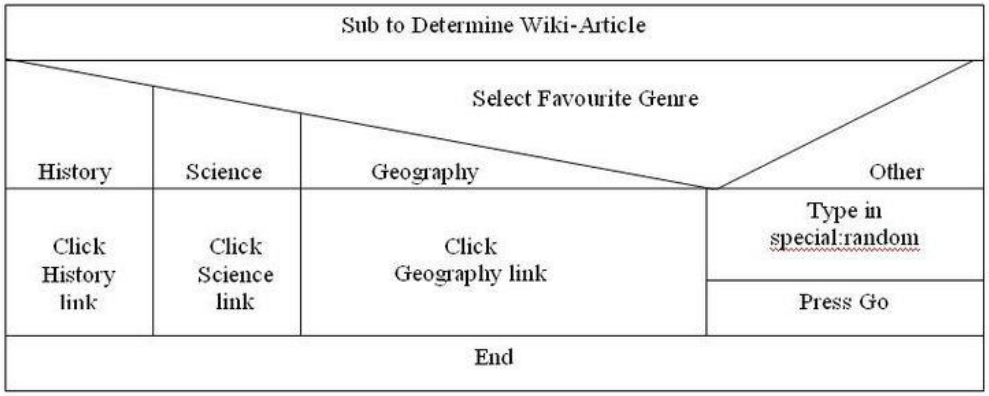
- Mô tả các khối xử lý bằng ngôn ngữ tự nhiên



- Khối rẽ nhánh đơn (2 nhánh đúng/sai)

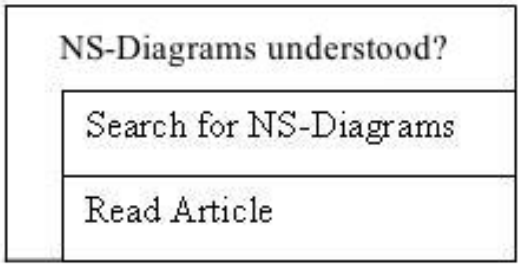


- Khối rẽ nhiều nhánh con

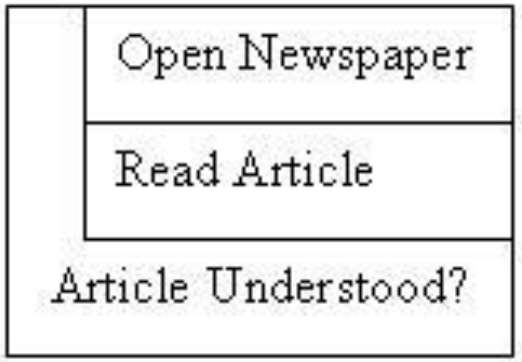


Lưu đồ Nassi–Shneiderman

- Lặp kiểm tra điều kiện trước



- Lặp kiểm tra điều kiện sau



Phần 3.3

Mã giả

Mã giả

- Mã giả mô tả thuật toán bằng một ngôn ngữ pha trộn giữa ngôn ngữ lập trình và ngôn ngữ tự nhiên đơn giản
- Ví dụ:

```
READ N from user input
IF N <= 1
    nfact = 1
ELSE
    nfact = 2
    REPEAT I = 3 to N
        nfact = nfact x I
END IF
OUTPUT nfact
```

Phần 4

Bài tập và thảo luận

Bài tập và thảo luận

1. Viết sơ đồ khối và lưu đồ Nassi–Shneiderman cho thuật toán tính n! (đã có mã giả ở slide 29).
2. Viết sơ đồ khối, lưu đồ Nassi–Shneiderman và mã giả cho thuật toán giải phương trình bậc 2 (slide 20).
3. Viết sơ đồ khối, lưu đồ Nassi–Shneiderman và mã giả cho thuật toán euclid (thuật toán tính ước số chung lớn nhất của hai số tự nhiên).
4. (về nhà) Tải phần mềm thông dịch python 3.7.0 từ website https://www.python.org và cài đặt trên máy tính của bạn.
5. (về nhà) Hãy tìm thuật toán cho câu hỏi ở slide 17, nếu tìm được thuật toán hiệu quả, hãy gửi cho thầy giáo để nhận điểm thưởng điểm quá trình.