BÁO CÁO
TÌM KIẾM VÀ ĐÁNH GIÁ THÔNG TIN HỌC THUẬT

Phát triển Tư duy Máy tính (Computational Thinking) cho sinh viên Công nghệ Thông tin năm nhất

Sinh viên: Nguyễn Minh Hiếu

Mã sinh viên: 25020152

Ngành: Công nghệ Thông tin

Kỳ học: Kỳ II - Năm nhất

I. Tổng quát

Bộ môn: Nhập môn Công nghệ số & Ứng dụng Trí tuệ nhân tạo

> **Buổi 2:** 2.4. Bài tập: Bài tập 2: Tìm kiếm và đánh giá thông tin học thuật với mục tiêu: Phát triển kỹ năng tìm kiếm và đánh giá thông tin học thuật từ các nguồn đáng tin cậy.

II. Giới thiệu chủ đề và phạm vi tìm kiếm

2.1. Lý do chọn chủ đề

Tư duy máy tính (Computational Thinking - CT) là nhóm năng lực quan trọng đối với sinh viên Công nghệ Thông tin năm nhất vì nó liên quan trực tiếp đến cách phân tích bài toán, thiết kế thuật toán, kiểm thử lời giải và chuyển vấn đề thực tế thành cấu trúc có thể xử lý bằng máy tính. Đối với sinh viên mới học lập trình, CT không chỉ là "biết viết code" mà còn là khả năng phân rã vấn đề, nhận dạng mẫu, trừu tượng hóa và xây dựng quy trình giải quyết vấn đề.

2.2. Phạm vi tìm kiếm

- **Đối tượng:** sinh viên đại học năm nhất, đặc biệt sinh viên CNTT hoặc ngành gần CNTT.
- **Khoảng thời gian ưu tiên:** 2017-2025; nguồn nền tảng trước 2017 vẫn được dùng khi có giá trị lý thuyết cao.
- **Khía cạnh nghiên cứu:** mức độ CT, yếu tố ảnh hưởng, công cụ đo lường, chiến lược giảng dạy và bối cảnh học CS hiện nay.
- **Ngôn ngữ:** chủ yếu tiếng Anh; thuật ngữ được diễn giải bằng tiếng Việt trong báo cáo.

III. Phương pháp tìm kiếm thông tin

Các nguồn được chọn theo bốn nhóm: cơ sở dữ liệu học thuật, tạp chí khoa học, sách/giáo trình chuyên khảo và nguồn mở đáng tin cậy. Từ khóa chính gồm: *computational thinking first-year university students*, *computational thinking programming education*, *algorithmic thinking computational thinking dimensions*, *object-oriented programming learning engagement*, và *computer science learning trends*.

Nhóm nguồn	Ví dụ nguồn đã dùng	Vai trò trong báo cáo
Cơ sở dữ liệu học thuật	ERIC, arXiv, cơ sở dữ liệu của trường/nhà xuất bản	Kiểm tra thông tin bài báo, tóm tắt, DOI, loại tài liệu.

Nhóm nguồn	Ví dụ nguồn đã dùng	Vai trò trong báo cáo
Tạp chí khoa học	Journal of Education and E-Learning Research, Computers & Education, Computers in Human Behavior, IJIET, Educational Researcher, Communications of the ACM	Cung cấp nghiên cứu thực nghiệm, tổng quan và khung lý thuyết.
Sách/giáo trình	Routledge, zyBooks	Bổ sung cơ sở sư phạm và tài liệu học lập trình tương tác.
Nguồn mở đáng tin cậy	JetBrains Academy, OECD	Cập nhật bối cảnh học CS, AI literacy và chính sách giáo dục.

IV. Tổng hợp và đánh giá 12 nguồn tham khảo

STT	Nguồn	Nội dung chính	Độ tin cậy	Ghi chú kiểm soát
1	**Cheng, N. & Osman, S.Z.M. (2025)** Bài báo nghiên cứu - ERIC / Journal of Education and E-Learning Research	Khảo sát 375 sinh viên năm nhất tại Trung Quốc; phân tích định lượng bằng thống kê mô tả, t-test và tương quan Pearson. Kết quả: CT mức trung bình, algorithmic thinking thấp nhất; kinh nghiệm lập trình và ngành học có ảnh hưởng đáng kể.	A+	Nguồn trọng tâm nhất, có DOI, có thông tin đạo đức nghiên cứu. Hạn chế: mẫu ở một trường/địa phương nên không khái quát tuyệt đối cho Việt Nam.
2	**Chin, X.J.L. et al. (2024)** Bài báo nghiên cứu - IJIET	Công cụ lập trình hướng đối tượng dạng block và game hóa; kết quả cho thấy điểm kiểm tra và mức độ ổn định học tập được cải thiện.	A	Có DOI, bài open access. Liên quan tốt đến dạy OOP nhưng không trực tiếp đo CT năm nhất.
3	**Grover, S. & Pea, R. (2013)** Bài báo tổng quan - Educational Researcher	Tổng quan lĩnh vực CT trong giáo dục K-12, làm rõ cách các nhà nghiên cứu diễn giải bài viết nền tảng của Wing và các hướng nghiên cứu cần phát triển.	A+	Nguồn nền tảng, peer-reviewed, DOI rõ ràng. Hạn chế: trọng tâm K-12 và đã xuất bản từ 2013.
4	**Korkmaz, Ö., Çakir, R. & Özden, M.Y. (2017)** Bài báo nghiên cứu - Computers in Human Behavior	Xây dựng và kiểm định thang đo Computational Thinking Scales (CTS); nghiên cứu kết luận thang đo có giá trị và độ tin cậy để đo kỹ năng CT của sinh viên/người học.	A+	Nguồn mạnh về công cụ đo lường. Dữ liệu trong file cũ ghi “21 mục” chưa chính xác với bản gốc CTS; báo cáo này không dùng con số sai đó.

STT	Nguồn	Nội dung chính	Độ tin cậy	Ghi chú kiểm soát
5	**Durak, H.Y. & Saritepeci, M. (2018)** Bài báo nghiên cứu - Computers & Education	Dùng mô hình quan hệ/SEM để xem các biến như phong cách tư duy, thành tích học tập và kinh nghiệm công nghệ giải thích kỹ năng CT như thế nào.	A	Tạp chí uy tín, DOI rõ. Hạn chế: mẫu là học sinh lớp 5-12, nên dùng như bằng chứng gián tiếp về yếu tố ảnh hưởng.
6	**Hsu, T.C., Chang, S.C. & Hung, Y.T. (2018)** Bài báo tổng quan - Computers & Education	Meta-review các nghiên cứu CT giai đoạn 2006-2017, tổng hợp công cụ, chiến lược học, ngôn ngữ lập trình và bối cảnh triển khai CT.	A+	Nguồn tốt để đề xuất chiến lược dạy CT. Hạn chế: phần lớn trọng tâm phổ thông, không chỉ đại học.
7	**Lockwood, J. & Mooney, A. (2018)** Bài báo tổng quan - International Journal of Computer Science Education in Schools	Tổng quan hệ thống về vị trí của CT trong giáo dục, nhấn mạnh CT là kỹ năng cần được đưa vào chương trình học và còn thiếu cấu trúc triển khai thống nhất.	A-	Có DOI, hữu ích cho khung lý thuyết. Hạn chế: tập trung secondary education.
8	**Wing, J.M. (2006)** Bài viết nền tảng - Communications of the ACM	Định nghĩa và phổ biến khái niệm CT như một cách tư duy dựa trên sức mạnh và giới hạn của quá trình tính toán, không chỉ dành riêng cho nhà khoa học máy tính.	A+	Nguồn kinh điển, có DOI. Hạn chế: bài khái niệm, không phải nghiên cứu thực nghiệm.
9	**Bers, M.U. (2021)** Sách chuyên khảo - Routledge	Trình bày cách dạy lập trình và CT qua hoạt động học tập mang tính chơi, nhấn mạnh học bằng tạo sản phẩm thay vì chỉ tiêu thụ công nghệ.	A	Được dùng để củng cố phần chiến lược sư phạm. File cũ ghi sai thành Bell & Wing (2016); báo cáo này đã sửa.
10	**McCown, F. & Ouellette, D. (2024)** Sách/giáo trình tương tác - zyBooks	Giáo trình tương tác về Android và Jetpack Compose; minh họa cách sách tương tác hỗ trợ người học lập trình mới bằng trực quan hóa, câu hỏi và hoạt động thực hành.	B+	Rất phù hợp với học lập trình ứng dụng cơ bản; không phải bài nghiên cứu CT độc lập.
11	**JetBrains Academy (2024)** Báo cáo mở/ industry report	Khảo sát 23.991 người học CS trên toàn cầu; cho thấy AI/ML là nhóm chủ đề đang thu hút mạnh người học CS và nhiều người học theo hướng tự học/online.	B+	Cập nhật và hữu ích cho bối cảnh; là khảo sát ngành, không peer-reviewed, có thiên lệch tự chọn mẫu.

STT	Nguồn	Nội dung chính	Độ tin cậy	Ghi chú kiểm soát
12	**Borgonovi, F. et al. (2025) - OECD** Báo cáo chính sách - OECD Artificial Intelligence Papers	Phân tích việc áp dụng AI trong hệ thống giáo dục, rủi ro, nguyên tắc và lộ trình chính sách; hữu ích để đặt CT trong bối cảnh năng lực số và AI literacy.	A-	Nguồn tổ chức quốc tế, có DOI. Hạn chế: trọng tâm chính là chính sách trường học và Italy, không trực tiếp về sinh viên CNTT năm nhất.

V. Phân tích tổng hợp

Khía cạnh	Phân tích
Mức độ CT của sinh viên năm nhất	Cheng & Osman (2025) cho thấy mức CT tổng thể ở mức trung bình (M = 3.59/5). Điều này ủng hộ việc cần dạy CT có hệ thống ngay từ học kỳ đầu.
Điểm yếu cần ưu tiên	Trong 5 chiều đo, algorithmic thinking có điểm trung bình thấp nhất (M = 3.33). Vì vậy sinh viên CNTT năm nhất nên được luyện phân rã bài toán, biểu diễn thuật toán, tracing và kiểm thử nhiều hơn.
Yếu tố ảnh hưởng	Kinh nghiệm lập trình trước và ngành học có ảnh hưởng đáng kể tới CT, trong khi khác biệt giới tính và nền tảng địa lý không có ý nghĩa thống kê trong nghiên cứu của Cheng & Osman.
Vai trò của công cụ học tập	Chin et al. (2024) và McCown & Ouellette (2024) gợi ý rằng công cụ trực quan, block-based, game hóa và giáo trình tương tác có thể giúp người mới học lập trình tiếp cận khái niệm khó dễ hơn.
Bối cảnh mới	JetBrains Academy (2024) và OECD (2025) cho thấy AI/ML và AI literacy đang làm thay đổi nhu cầu học CS; CT nên được dạy như năng lực nền để hiểu, đánh giá và sử dụng công nghệ mới.

5.1. Ý nghĩa đối với sinh viên CNTT năm nhất

Các tài liệu cho thấy CT nên được coi là năng lực nền trong học lập trình, không phải phần phụ trợ sau khi đã học xong ngôn ngữ lập trình. Với sinh viên năm nhất, điểm dễ yếu nhất thường không phải cú pháp đơn lẻ mà là khả năng thiết kế thuật toán, kiểm tra luồng xử lý, chọn cấu trúc dữ liệu phù hợp và diễn giải lỗi. Vì vậy, việc dạy CT nên đi song song với các môn nhập môn lập trình, OOP, cấu trúc dữ liệu và phát triển ứng dụng.

5.2. Đề xuất chiến lược học và dạy

- **Tăng bài tập thuật toán nhỏ nhưng đều:** luyện pseudocode, flowchart, tracing, dry-run và kiểm thử biên trước khi viết code.
- **Dạy theo dự án nhỏ:** mỗi chủ đề lập trình nên có bài mini-project để sinh viên tự phân rã yêu cầu, thiết kế giải pháp và phản tư sau khi làm.

- **Dùng công cụ trực quan:** block-based programming, animation, debugger, visualizer hoặc sách tương tác giúp sinh viên mới nhìn được trạng thái chương trình.
- **Phân tầng người học:** sinh viên đã có kinh nghiệm lập trình và sinh viên mới hoàn toàn cần bài tập khác mức độ; không nên dùng một lộ trình cứng cho tất cả.
- **Liên hệ AI literacy:** khi AI trở thành công cụ học lập trình phổ biến, sinh viên cần CT để kiểm tra, phản biện và sửa lời giải do AI gợi ý.

5.3. Hạn chế của bộ tài liệu

Bộ tài liệu có một nguồn trực tiếp về sinh viên năm nhất (Cheng & Osman, 2025) và nhiều nguồn bổ trợ về CT, OOP, công cụ học tập và bối cảnh AI. Vì vậy, kết luận không nên hiểu là mô tả chính xác toàn bộ sinh viên CNTT Việt Nam. Đây là tổng hợp học thuật có kiểm soát, phù hợp để định hướng chủ đề và đề xuất chiến lược học tập, nhưng vẫn cần thêm khảo sát tại Việt Nam nếu muốn đưa ra kết luận định lượng cho sinh viên trong nước.

VI. Kết luận

Sau khi kiểm chứng và làm sạch dữ liệu, có thể kết luận rằng chủ đề “Phát triển Tư duy Máy tính cho sinh viên CNTT năm nhất” có cơ sở học thuật phù hợp, có nguồn nền tảng rõ ràng và có ý nghĩa thực tiễn. Tài liệu trọng tâm cho thấy sinh viên năm nhất có mức CT trung bình, trong đó algorithmic thinking là chiều đo yếu cần được ưu tiên. Các nguồn tổng quan và công cụ học tập gợi ý rằng việc kết hợp bài tập thuật toán, học theo dự án, công cụ trực quan, game hóa và giáo trình tương tác có thể giúp sinh viên mới tiếp cận lập trình hiệu quả hơn.

VII. Danh mục tài liệu tham khảo

Bers, M.U. (2021) *Coding as a Playground: Programming and Computational Thinking in the Early Childhood Classroom*. 2nd edn. New York: Routledge. doi: 10.4324/9781003022602.

Borgonovi, F. et al. (2025) *AI adoption in the education system: International insights and policy considerations for Italy*. OECD Artificial Intelligence Papers, No. 52. Paris: OECD Publishing. doi: 10.1787/69bd0a4a-en.

Cheng, N. and Osman, S.Z.M. (2025) 'Level of computational thinking skills among first-year university students: Variation across gender, geographical background, academic discipline, and programming learning experience', *Journal of Education and E-Learning Research*, 12(2), pp. 231-238. doi: 10.20448/jeelr.v12i2.6757.

Chin, X.J.L., Seow, C.K., Cai, Y., Zhu, Y., Wang, M. and Cao, Q. (2024) 'Enhancing student learning and engagement with object-oriented block-based programming tool', *International Journal of Information and Education Technology*, 14(7), pp. 910-919. doi: 10.18178/ijiet.2024.14.7.2117.

Durak, H.Y. and Saritepeci, M. (2018) 'Analysis of the relation between computational thinking skills and various variables with the structural equation model', *Computers & Education*, 116, pp. 191-202. doi: 10.1016/j.compedu.2017.09.004.

Grover, S. and Pea, R. (2013) 'Computational thinking in K-12: A review of the state of the field', *Educational Researcher*, 42(1), pp. 38-43. doi: 10.3102/0013189X12463051.

Hsu, T.C., Chang, S.C. and Hung, Y.T. (2018) 'How to learn and how to teach computational thinking: Suggestions based on a review of the literature', *Computers & Education*, 126, pp. 296-310. doi: 10.1016/j.compedu.2018.07.004.

JetBrains Academy (2024) *Computer Science Learning Curve Survey 2024 Report*. Available at: https://lp.jetbrains.com/cs-learning-curve-report-2024/ (Accessed: 4 June 2026).

Korkmaz, Ö., Çakir, R. and Özden, M.Y. (2017) 'A validity and reliability study of the computational thinking scales (CTS)', *Computers in Human Behavior*, 72, pp. 558-569. doi: 10.1016/j.chb.2017.01.005.

Lockwood, J. and Mooney, A. (2018) 'Computational thinking in secondary education: Where does it fit? A systematic literary review', *International Journal of Computer Science Education in Schools*, 2(1), pp. 41-60. doi: 10.21585/ijcses.v2i1.26.

McCown, F. and Ouellette, D. (2024) *Mobile App Development with Android and Jetpack Compose*. zyBooks. Available at: https://scholarworks.harding.edu/computer-science-facpub/13/ (Accessed: 4 June 2026).

Wing, J.M. (2006) 'Computational thinking', *Communications of the ACM*, 49(3), pp. 33-35. doi: 10.1145/1118178.1118215.