RBECT
Revista Brasileira de Ensino de Ciência e Tecnologia

Revista Brasileira de Ensino de Ciência e Tecnologia
ISSN: 1982-873X

https://periodicos.utfpr.edu.br/rbect

PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar.

Jeannette Wing
wing@cs.cmu.edu
0000-0002-1013-1990
Carnegie Mellon University, Pennsylvania, United States.

RESUMO

Apresenta-se aqui a tradução do trabalho intitulado “Computational Thinking”, da autora americana Jeannette Wing, professora de Ciência da Computação e chefe do Departamento de Ciência da Computação na Universidade de Carnegie Mellon, Pittsburgh, PA. O trabalho original foi publicado no número 3 da edição 49 do periódico “Communications of the ACM”, em março de 2006. DOI 0001-0782/06/0300.

O trabalho foi traduzido para o português por Cleverson Sebastião dos Anjos, professor da área de informática do Instituto Federal do Paraná.

Pensamento computacional baseia-se no poder e limites de processos computacionais, sejam eles executados por um humano ou por uma máquina. Métodos e modelos computacionais nos dão a coragem para resolver problemas e projetar sistemas que nenhum de nós seria capaz de enfrentar sozinhos. O pensamento computacional confronta o enigma da inteligência da máquina: O que humanos fazem melhor que computadores? E O que computadores fazem melhor que humanos? De forma mais fundamental, ele trata a questão: O que é computável? Hoje, sabemos apenas parte da resposta para essas perguntas.

Pensamento computacional é uma habilidade fundamental para todos, não somente para cientistas da computação. À leitura, escrita e aritmética, deveríamos incluir pensamento computacional na habilidade analítica de todas as crianças. Assim como a máquina impressora facilitou a divulgação dos três Rs[1], o que é apropriadamente incestuoso sobre essa visão é que a computação e os computadores facilitaram a divulgação do pensamento computacional.

Pensamento computacional envolve a resolução de problemas, projeção de sistemas, e compreensão do comportamento humano, através da extração de conceitos fundamentais da ciência da computação. O pensamento computacional inclui uma série de ferramentas mentais que refletem a vastidão do campo da ciência da computação.

Tendo que resolver um problema em particular, podemos perguntar: O quão difícil é de resolver? e Qual é a melhor forma de resolvê-lo? A ciência da computação se apoia em fundamentos teóricos sólidos para responder tais perguntas com precisão. Afirmar a dificuldade de um problema leva em conta a capacidade subjacente à máquina. Devemos considerar o conjunto de instruções da máquina, suas restrições de recursos e seu ambiente operacional.

Ao resolver um problema eficientemente, podemos questionar se uma solução aproximada é boa o suficiente e se falsos positivos ou falsos negativos são permitidos. O pensamento computacional é reformular um problema aparentemente difícil em um problema que sabemos como resolver, talvez por redução, incorporação, transformação ou simulação.

Pensamento computacional é pensar recursivamente. É processamento paralelo. É interpretar código como dado e dado como código. É verificação de tipo como uma generalização da análise dimensional. É reconhecer ambas as virtudes e os perigos da utilização de *alias*, ou dar a alguém ou alguma coisa mais de um nome. É reconhecer ambos o custo e o poder do endereçamento indireto e chamada de procedimento. É julgar um programa não apenas pela sua corretude e eficiência, mas pela sua estética, e a interface de um sistema pela sua simplicidade e elegância.

Pensamento computacional é usar abstração e decomposição ao atacar uma tarefa grande e complexa ou projetar um sistema complexo e grande. É a separação de interesses. É escolher uma representação apropriada para um

problema ou modelagem dos aspectos relevantes de um problema para torná-lo tratável. É usar invariantes para descrever o comportamento de um sistema de forma sucinta e declarativa. É ter a confiança de que podemos usar, modificar e influenciar um sistema grande e complexo sem entender todos os seus detalhes com segurança, modularizar algo em antecipação de múltiplos usuários ou *prefetching* e *caching* em antecipação de um uso futuro.

Pensamento computacional é pensar em termos de prevenção, proteção, e recuperação em cenários de pior caso através da redundância, contenção de danos e correção de erros. É chamar impasse de *deadlock* e contratos de interfaces. É aprender a evitar condições de corrida quando sincronizações se encontram.

Pensamento computacional é usar raciocínio heurístico na descoberta de uma solução. É planejar, aprender e agendar na presença da incerteza. É pesquisar, pesquisar e pesquisar mais, resultando em uma lista de páginas da web, uma estratégia para vencer um jogo ou um contraexemplo. Pensamento computacional é usar quantidades imensas de dados para aumentar a velocidade da computação. É fazer concessões entre tempo e espaço e entre poder de processamento e capacidade de armazenamento.

Considere esses exemplos do dia a dia: Quando sua filha vai para a escola pela manhã, ela coloca em sua mala as coisas que precisará para o dia; isso é *prefatching* e *caching*. Quando seu filho perde suas luvas, você sugere que ele refaça seus passos; isso é *backtracking*. Em que ponto você pára de alugar esquis e compra seu próprio?; isso são algoritmos *on line*. Em qual fila do mercado você fica?; isso é modelagem de performance para sistemas multisservidores. Por que seu telefone continua funcionando mesmo com falta de energia?; isso é independência de falha e redundância de projeto. Como um Teste de Turing Público Completamente Automatizado para Diferenciação entre Computadores e Humanos, ou CAPTCHA, autentica humanos?; isso é exploração da dificuldade de resolução de problemas difíceis de inteligência artificial para enganar agentes computacionais.

O pensamento computacional vai ter se tornado impregnado na vida de todo mundo quando palavras como algoritmos e precondição tornarem-se parte do vocabulário; quando não determinismo e coleta de lixo tomarem o significado usado por cientistas da computação; e quando árvores forem desenhadas de ponta cabeça.

Testemunhamos a influência da computação em outras disciplinas. Por exemplo, a aprendizagem de máquina tem transforado a estatística. A aprendizagem estatística está sendo usada para problemas em escala em termos de tamanho e dimensão de dados inimagináveis até apenas alguns anos atrás. Departamentos de estatística em todos os tipos de organizações estão contratando cientistas da computação. Escolas de ciência da computação estão apoiando ou abrindo novos departamentos de estatísticas.

O interesse recente de cientistas da computação pela biologia é dado pela crença de que biólogos podem se beneficiar de pensamento computacional. A contribuição do pensamento computacional para a biologia vai além da habilidade de pesquisar em grandes quantidades de sequências de dados em busca de padrões. A esperança é de que estruturas de dados e algoritmos – nossas

abstrações computacionais e métodos – possam representar a estrutura de proteínas de forma a elucidar suas funções. A biologia computacional está mudando a forma como os biólogos pensam. Similarmente, a teoria de jogos computacional está mudando a forma como os economistas pensam; nano computação, a forma como químicos pensam; e computação quântica, a forma como os físicos pensam.

Esse tipo de pensamento será parte do conjunto de habilidades não somente de outros cientistas, mas de todas as pessoas. A computação ubíqua está para o hoje assim como o pensamento computacional está para o amanhã. A computação ubíqua era o sonho de ontem que se tornou a realidade de hoje; pensamento computacional é a realidade do amanhã.

O QUE ELE É, E O QUE NÃO É

Ciência computacional é o estudo da computação – o que pode ser computado e como pode ser computado. Sendo assim, o pensamento computacional possui as seguintes características:

Conceptualização, não programação. Ciência da computação não é programação. Pensar como um cientista da computação significa mais do que ser capaz de programar um computador. É preciso pensar em múltiplos níveis de abstração;

Uma habilidade fundamental, não mecânica. Uma habilidade fundamental é algo que todo ser humano deve saber para atuar na sociedade moderna. Habilidade mecânica significa rotina mecânica. Ironicamente, somente quando a ciência da computação resolver o Grande Desafio da Inteligência Artificial, fizer com que computadores pensem como seres humanos, o pensamento será considerado mecânico;

Uma forma que humanos, não computadores, pensam. Pensamento computacional é uma forma para seres humanos resolverem problemas; não é tentar fazer com que seres humanos pensem como computadores. Computadores são tediosos e enfadonhos; humanos são espertos e imaginativos. Nós humanos tornamos a computação empolgante. Equipados com aparelhos computacionais, usamos nossa inteligência para resolver problemas que não ousaríamos sequer tentar antes da era da computação e construir sistemas com funcionalidades limitadas apenas pela nossa imaginação;

Complementa e combina pensamento matemático e de engenharia. A ciência da computação baseia-se inerentemente no pensamento de engenharia, uma vez que construímos sistemas que interagem com o mundo real. As limitações do dispositivo de computação subjacente forçam cientistas da computação a pensar de forma computacional, não somente matematicamente. Ser livre para construir mundos virtuais nos permite engenhar sistemas além do mundo físico;

Ideias, não artefatos. Não são apenas os artefatos de software e hardware que produzimos que estarão presentes fisicamente em todos os lugares e tocarão nossas vidas todo o tempo, serão os conceitos computacionais que usamos para

abordar e resolver problemas, gerenciar nossas vidas diárias e comunicar e interagir com outras pessoas; e

Para todas as pessoas, em todos os lugares. Pensamento computacional será uma realidade quando for tão essencial aos empreendimentos humanos que acaba por desaparecer como uma filosofia explícita.

Muitas pessoas igualam a ciência da computação à programação de computadores. Alguns pais enxergam apenas uma faixa pequena de oportunidades de trabalho para seus filhos que se graduam em ciência da computação. Muitas pessoas acham que a pesquisa fundamental na ciência da computação já foi realizada e que apenas a engenharia restou. Pensamento computacional é uma grande visão para guiar educadores, pesquisadores, e praticantes da ciência da computação para mudarmos a imagem que a sociedade tem da área. Nós precisamos alcançar em especial o público pré-universitário, incluindo professores, pais e alunos, enviando a eles duas mensagens:

Problemas intelectualmente desafiadores e interessantes permanecem sem solução e compreensão. O domínio do problema e o domínio da solução ainda são limitados apenas pela nossa criatividade e curiosidade; e

Aquele que se gradua em ciência da computação pode fazer qualquer coisa que desejar. Pode-se graduar em inglês ou matemática e seguir uma grande variedade de carreiras diferentes. O mesmo pode ser dito da ciência da computação. Pode-se graduar em ciência da computação e seguir uma carreira em medicina, direito, administração, política, qualquer tipo de ciência ou engenharia e até mesmo artes.

Professores de ciência da computação deveriam lecionar uma disciplina chamada “Formas de Pensar como um Cientista da Computação” para calouros na faculdade, tornando-a disponível não apenas para alunos do curso de ciência da computação. Devemos expor os estudantes pré-universitários aos métodos e modelos computacionais. Ao invés de lamentar a diminuição no interesse pela ciência da computação ou a redução no financiamento de pesquisas na ciência da computação, devemos buscar formas de inspirar o interesse do público na aventura intelectual do campo. Espalharemos assim a alegria, admiração e poder da ciência da computação buscando fazer do pensamento computacional um lugar comum.

RBECT
Revista Brasileira de Ensino de Ciência e Tecnologia
Revista Brasileira de Ensino de Ciência e Tecnologia
ISSN: 1982-873X

COMPUTATIONAL THINKING - It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

ABSTRACT

Here we present the Portuguese translation of the paper called “Computational Thinking”, from the American author Jeannette Wing, President’s Professor of Computer Science and head of the Computer Science Department at Carnegie Mellon University, PA. The original paper was published in the “Communications of the ACM”, number 3, edition 49, in March of 2006. DOI 0001-0782/06/0300.

The paper was translated into Portuguese by Cleverson Sebastião dos Anjos, professor of Computer Science of the Federal Institute of Paraná.

Computational thinking builds on the power and limits of computing processes, whether they are executed by a human or by a machine. Computational methods and models give us

the courage to solve problems and design systems that no one of us would be capable of tackling alone. Computational thinking confronts the riddle of machine intelligence: What can humans do better than computers? and What can computers do better than humans? Most fundamentally it addresses the question: What is computable? Today, we know only parts of the answers to such questions.

Computational thinking is a fundamental skill for everyone, not just for computer scientists. To reading, writing, and arithmetic, we should add computational thinking to every child's analytical ability. Just as the printing press facilitated the spread of the three Rs, what is appropriately incestuous about this vision is that computing and computers facilitate the spread of computational thinking.

Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science.

Having to solve a particular problem, we might ask: How difficult is it to solve? and What's the best way to solve it? Computer science rests on solid theoretical underpinnings to answer such questions precisely. Stating the difficulty of a problem accounts for the underlying power of the machine—the computing device that will run the solution. We must consider the machine's instruction set, its resource constraints, and its operating environment.

In solving a problem efficiently, we might further ask whether an approximate solution is good enough, whether we can use randomization to our advantage, and whether false positives or false negatives are allowed. Computational thinking is reformulating a seemingly difficult problem into one we know how to solve, perhaps by reduction, embedding, transformation, or simulation.

Computational thinking is thinking recursively. It is parallel processing. It is interpreting code as data and data as code. It is type checking as the generalization of dimensional analysis. It is recognizing both the virtues and the dangers of aliasing, or giving someone or something more than one name. It is recognizing both the cost and power of indirect addressing and procedure call. It is judging a program not just for correctness and efficiency but for aesthetics, and a system's design for simplicity and elegance.

Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system. It is separation of concerns. It is choosing an appropriate representation for a problem or modeling the relevant aspects of a problem to make it tractable. It is using invariants to describe a system's behavior succinctly and declaratively. It is having the confidence we can safely use, modify, and influence a large complex system without understanding its every detail. It is modularizing something in anticipation of multiple users or prefetching and caching in anticipation of future use.

Revista Brasileira de Ensino de Ciência e Tecnologia
ISSN: 1982-873X

Computational thinking is thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction. It is calling gridlock deadlock and contracts interfaces. It is learning to avoid race conditions when synchronizing meetings with one another.

Computational thinking is using heuristic reasoning to discover a solution. It is planning, learning, and scheduling in the presence of uncertainty. It is search, search, and more search, resulting in a list of Web pages, a strategy for winning a game, or a counterexample. Computational thinking is using massive amounts of data to speed up computation. It is making trade-offs between time and space and between processing power and storage capacity.

Consider these everyday examples: When your daughter goes to school in the morning, she puts in her backpack the things she needs for the day; that's prefetching and caching. When your son loses his mittens, you suggest he retrace his steps; that's backtracking. At what point do you stop renting skis and buy yourself a pair?; that's online algorithms. Which line do you stand in at the supermarket?; that's performance modeling for multi-server systems. Why does your telephone still work during a power outage?; that's independence of failure and redundancy in design. How do Completely Automated Public Turing Test(s) to Tell Computers and Humans Apart, or CAPTCHAs, authenticate humans?; that's exploiting the difficulty of solving hard AI problems to foil computing agents.

Computational thinking will have become ingrained in everyone's lives when words like algorithm and precondition are part of everyone's vocabulary; when nondeterminism and garbage collection take on the meanings used by computer scientists; and when trees are drawn upside down.

We have witnessed the influence of computational thinking on other disciplines. For example, machine learning has transformed statistics. Statistical learning is being used for problems on a scale, in terms of both data size and dimension, unimaginable only a few years ago. Statistics departments in all kinds of organizations are hiring computer scientists. Schools of computer science are embracing existing or starting up new statistics departments.

Computer scientists' recent interest in biology is driven by their belief that biologists can benefit from computational thinking. Computer science's contribution to biology goes beyond the ability to search through vast amounts of sequence data looking for patterns. The hope is that data structures and algorithms—our computational abstractions and methods—can represent the structure of proteins in ways that elucidate their function. Computational biology is changing the way biologists think. Similarly, computational game theory is changing the way economists think; nanocomputing, the way chemists think; and quantum computing, the way physicists think.

This kind of thinking will be part of the skill set of not only other scientists but of everyone else. Ubiquitous computing is to today as computational thinking is to tomorrow. Ubiquitous computing was yesterday's dream that became today's reality; computational thinking is tomorrow's reality.

WHAT IT IS, AND ISN'T

Computer science is the study of computation— what can be computed and how to compute it. Computational thinking thus has the following characteristics:

Conceptualizing, not programming. Computer science is not computer programming. Thinking like a computer scientist means more than being able to program a computer. It requires thinking at multiple levels of abstraction;

Fundamental, not rote skill. A fundamental skill is something every human being must know to function in modern society. Rote means a mechanical routine. Ironically, not until computer science solves the AI Grand Challenge of making computers think like humans will thinking be rote;

A way that humans, not computers, think. Computational thinking is a way humans solve problems; it is not trying to get humans to think like computers. Computers are dull and boring; humans are clever and imaginative. We humans make computers exciting. Equipped with computing devices, we use our cleverness to tackle problems we would not dare take on before the age of computing and build systems with functionality limited only by our imaginations;

Complements and combines mathematical and engineering thinking. Computer science inherently draws on mathematical thinking, given that, like all sciences, its formal foundations rest on mathematics. Computer science inherently draws on engineering thinking, given that we build systems that interact with the real world. The constraints of the underlying computing device force computer scientists to think computationally, not just mathematically. Being free to build virtual worlds enables us to engineer systems beyond the physical world;

Ideas, not artifacts. It's not just the software and hardware artifacts we produce that will be physically present everywhere and touch our lives all the time, it will be the computational concepts we use to approach and solve problems, manage our daily lives, and communicate and interact with other people; and

For everyone, everywhere. Computational thinking will be a reality when it is so integral to human endeavors it disappears as an explicit philosophy

Many people equate computer science with computer programming. Some parents see only a narrow range of job opportunities for their children who major in computer science. Many people think the fundamental research in computer science is done and that only the engineering remains. Computational thinking is a grand vision to guide computer science educators, researchers, and practitioners as we act to change society's image of the field. We especially need to reach the pre-college audience, including teachers, parents, and students, sending them two main messages:

Intellectually challenging and engaging scientific problems remain to be understood and solved. The problem domain and solution domain are limited only by our own curiosity and creativity; and

One can major in computer science and do anything. One can major in English or mathematics and go on to a multitude of different careers. Ditto computer science. One can major in computer science and go on to a career in medicine, law, business, politics, any type of science or engineering, and even the arts.

Revista Brasileira de Ensino de Ciência e Tecnologia
ISSN: 1982-873X

Professors of computer science should teach a course called “Ways to Think Like a Computer Scientist” to college freshmen, making it available to non-majors, not just to computer science majors. We should expose pre-college students to computational methods and models. Rather than bemoan the decline of interest in computer science or the decline in funding for research in computer science, we should look to inspire the public’s interest in the intellectual adventure of the field. We’ll thus spread the joy, awe, and power of computer science, aiming to make computational thinking commonplace.

Recebido: 01 out. 2016.
Aprovado: 01 out. 2016.
DOI: http://dx.doi.org/10.3895/rbect.v9n2.4711
Como citar: WING, J. PENSAMENTO COMPUTACIONAL – Um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. **Revista Brasileira de Ensino de Ciência e Tecnologia**, v. 9, n. 2, 2016. Disponível em: <https://periodicos.utfpr.edu.br/rbect/article/view/4711>. Acesso em: xxx.
Correspondência:
Jeannettte Wing
One Microsoft Way, Microsoft Research, Redmond, WA 98052