

بيتر جيہ دنيڻج وماتي تيدري

التفكير الحوسبي

ترجمة هبة عبد العزيز غانم



سلسلة المعارف الأساسية

التفكير الحوسبي

تأليف

بيتر جيه دنينج وماتي تيدري

ترجمة

هبة عبد العزيز غانم

مراجعة

هبة عبد المولى أحمد



الناشر مؤسسة هنداوي

المشهرة برقم ١٠٥٨٥٩٧٠ بتاريخ ٢٦/١/٢٠١٧

يورك هاوس، شبيت ستريت، وندسور، SL4 1DD، المملكة المتحدة

تليفون: ٨٣٢٥٢٢ ١٧٥٣ (٠) ٤٤ +

البريد الإلكتروني: hindawi@hindawi.org

الموقع الإلكتروني: https://www.hindawi.org

إن مؤسسة هنداوي غير مسئولة عن آراء المؤلف وأفكاره، وإنما يعبر الكتاب عن آراء مؤلفه.

تصميم الغلاف: ولاء الشاهد

الترقيم الدولي: ٩٧٨ ١ ٥٢٧٣ ٣٧٨١ ٧

صدر الكتاب الأصلي باللغة الإنجليزية عام ٢٠١٩.

صدرت هذه الترجمة عن مؤسسة هنداوي عام ٢٠٢٥.

جميع حقوق النشر الخاصة بتصميم هذا الكتاب وتصميم الغلاف محفوظة لمؤسسة هنداوي.
جميع حقوق النشر الخاصة بالترجمة العربية لنص هذا الكتاب محفوظة لمؤسسة هنداوي.
جميع حقوق النشر الخاصة بنص العمل الأصلي محفوظة لمعهد ماساتشوستس للتكنولوجيا
(إم آي تي).

المحتويات

٧	شكر وتقدير
٩	تمهيد السلسلة
١١	تمهيد
١٧	١- ما المقصود بالتفكير الحوسبي؟
٣١	٢- أساليب الحوسبة
٤٣	٣- آلات الحوسبة
٥٩	٤- علوم الكمبيوتر
٧٥	٥- هندسة البرمجيات
٩٧	٦- التصميم لتلبية احتياجات الإنسان
١١١	٧- العلوم الحوسبيّة
١٢٩	٨- تعليم التفكير الحوسبي للجميع
١٤١	٩- مستقبل الحوسبة
١٥٣	خاتمة: الدروس المُستفادة
١٥٧	مسرد المصطلحات
١٦٣	ملاحظات
١٦٧	مراجع وقراءات إضافية

شكر وتقدير

بيتر: أقدم جزيل الشكر إلى دوروثي ديننج، زوجتي، التي استمعت إلى خطاباتي الكثيرة حول الحوسبة لمدة تزيد عن خمسين عامًا، وأرشدتني إلى اتجاهات مثمرة. كما أتوجّه بتقديري العميق إلى صديقي فرناندو فلوريس، لتعليمي كيفية قراءة التاريخ للتعرف على المخاوف الناشئة عنه، ومن ثمّ تمييز المراحل المختلفة للتفكير الحوسبي على مرّ القرون. وأشكر مؤسسي الحوسبة وتعليم الحوسبة الذين قابلتهم من خلال جمعية آلات الحوسبة، ومنهم إكرت وموتشلي وبرليس ونيوويل وسایمون وفورسايت وكونتي وويلكس وهامينج وكنوث ودايكسترا. كما أتوجّه بكلّ العرفان والتقدير إلى أساتذتي في معهد ماساتشوستس للتكنولوجيا الذين حولوني من مجرد مهندس كهرباء إلى أحد أوائل حاملي درجة علوم الكمبيوتر، خاصة فانو وكورباتو ودينيس وسالترز وشير وزاديه. وأشكر أيضًا العديد من زملائي في علوم الكمبيوتر والهندسة على مرّ السنين، الذين لا يتسع المجال لذكرهم جميعًا هنا، والذين انخرطوا معي في محادثات تثقيفية حول الحوسبة.

ماتي: أنا مُمتنٌّ لكلّ الذين ساعدوني في تطوير تفكيري الحوسبي: معلميّ القدامى، ومرشديّ، وزملائي السابقين والحاليين في شتّى مجالات البحث. لقد كنت محظوظًا لكوني اكتسبتُ ثروة من الأفكار المتعلقة بالحوسبة من العمل في الجامعات في ستّ دول في ثلاث قارات. كما أنني أتقدم بالشكر إلى أصدقائي وزملائي من مجال تاريخ وفلسفة علوم الكمبيوتر – وهم كثيرون جدًّا لدرجة أنني لن أستطيع ذكرهم هنا بالاسم. كما أودُّ أن أُوجّه شكرًا خاصًّا إلى كلّ من مارتن بولينك وإدجار دايليت وليزبت دي مول ولوري مالي وجون بايونن وجوسيبي بريميرو وسایمون (وكذلك شركائي في برنامج

ANR ANR-17-CE38-0003-01) على المحادثات الملهمة والتعليقات والتعاون في المواد ذات الصلة المباشرة بهذا الكتاب. وقد حصلت على دعم جزئي لعملي من قبل جمعية الكُتّاب الفنلنديين غير الروائيين.

شكر وتقدير للناشرين السابقين

اقتُبست أجزاء من الفصل الخامس من المصادر التالية:

كتاب «المبادئ الكبرى للحوسبة» من تأليف بيتر دنينج وكريج مارتل (MIT Press, 2015).
وكتاب «المهندس المنسي» من تأليف دنينج (Communications of the ACM 60, 12) [December 2017]: 20–23.
وكتاب «الحوسبة كهندسة» من تأليف ماتي تيدري (Journal of Universal Computer) [December 2013]: 29–31.
(Science 15, 8: 1642–1658).

واقْتُبست أجزاء من الفصل السادس:

من كتاب «جودة البرمجيات» من تأليف بيتر دنينج (Communications of the ACM) [September 2016]: 23–25.
ومن كتاب «التفكير التصميمي» من تأليف بيتر دنينج (Communications of the ACM) [December 2013]: 29–31.
ومن كتاب «المبادئ الكبرى للحوسبة».

كما اقتُبست أجزاء من الفصل السابع من كتاب «التفكير الحوسبي في العلوم» من تأليف بيتر دنينج (American Scientist 105 [January–February 2017]: 13–17).

تمهيد السلسلة

تُقدِّم «سلسلة المعارف الأساسية» التي تنشرها مؤسسة «إم آي تي بريس» كُتُبًا موجزة بلغةٍ جَزلة سهلة الفهم، وشكلٍ أنيق، وحجمٍ صغير يُلائم الجيب، تُناقش الموضوعات التي تُثير الاهتمام في الوقت الحالي. ولما كانت كُتُب هذه السلسلة من تأليف مفكرين بارزين، فإنها تُقدِّم آراء الخبراء بشأن موضوعاتٍ تتنوّع بين المجالات الثقافية والتاريخية، بالإضافة إلى العلمية والتقنية.

في ظلّ ما يَشيع في هذا العصر من إشباعٍ لحظي للمعلومات، أضحي لدى الجميع القدرةُ على الوصول إلى الآراء والأفكار والشروح السطحية بسرعةٍ وسهولة، وأصبح من الصعوبة بمكانٍ أن يحظى المرءُ بالمعرفة الأساسية التي تُيسِّر فهمًا صادقًا للعالم؛ وما تفعله كُتُب هذه السلسلة هو أنها تُحقِّق ذلك الغرض. وكل كتابٍ من هذه الكُتُب المختصرة يُقدِّم للقارئ وسيلةً مُيسّرة للوصول إلى الأفكار المعقّدة، من خلال تبسيط المواد المُختصّة لغير المُختصّين، وشرّح الموضوعات المهمة بأبسط طريقةٍ ممكنة.

بروس تيدور

أستاذ الهندسة البيولوجية وعلوم الكمبيوتر

«معهد ماساتشوستس للتكنولوجيا»

تمهيد

نحن الآن في خضم ثورة هائلة في مجال الكمبيوتر. وقد أدى غزو الحوسبة لمختلف جوانب حياتنا إلى فوائد جمة تشمل البريد الإلكتروني، والإنترنت، وشبكة ويب العالمية، والتجارة الإلكترونية عبر أمازون، وأكاديمية خان، وخدمة «أوبر» لطلب سيارات الأجرة، وخرائط جوجل، وأجهزة الملاحة، والهواتف الذكية، وبرامج الترجمة الآنية، وغيرها من ملايين التطبيقات. وفي الوقت نفسه، أثار ذلك مخاوف كبيرة تشمل احتمال فقدان الوظائف بسبب الأتمتة، وخضوعنا جميعًا للمراقبة، وانهيار البنية التحتية الأساسية، والحرب الإلكترونية، وبيع بياناتنا الشخصية على نطاق واسع، وغزو الإعلانات، وفقدان الخصوصية، والاستقطاب السياسي، وانعدام الأدب والافتقار إلى حُسن الإصغاء، وتفاقم مشكلة تفاوت الدخل.

يُعاني الكثير من الناس من صعوبة في استيعاب كل هذا. فهل يمكنهم جني ثمار هذا التطوُّر دون تكبُّد تكاليفه الباهظة؟ وهل يمكنهم عيش حياة هادئة إذا هددت الحوسبة فجأة بإحالة كل ما تعلموه طوال حياتهم إلى شيءٍ عتيق قديم الطراز؟ وما الذي يجب أن يتعلمه أطفالهم عن الحوسبة لتمكينهم من التقدم والنجاح في العالم الجديد؟ يُعد التفكير الحوسبي مصطلحًا جديدًا، دخل الخطاب العام حديثًا، مع محاولات الناس المُستمِنة للإجابة عن هذه الأسئلة. إنه يمنحنا بصيص أمل في أن نتمكن من التفكير بوضوح في قوى الحوسبة الجماعية وأخطارها، وأن نتعلَّم تصميم أجهزة الكمبيوتر والبرامج والشبكات لزيادة الفوائد وتقليل المخاطر. يشعر الآباء بالفعل بالدهشة من سهولة تعامل أطفالهم مع العالم الرقمي. فهل التفكير الحوسبي هو السبيل إلى منح أطفالنا تعليمًا مناسبًا في هذا العالم؟

لقد أعدنا هذا الكتاب ليكون حوارًا مفيدًا لمساعدتك على فهم ماهية التفكير الحوسبي حتى تستطيع الإجابة عن هذه الأسئلة بنفسك.

إنَّ أول ما يجب عليك فهمه هو أنَّ جزءًا كبيرًا من الخطاب اليومي يتشكل بفعل الانتشار الواسع لأجهزة الكمبيوتر. وهذا ليس جديدًا؛ فقد تشكلت طرق تفكير أسلافنا بفعل تقنيات الثورات السابقة. ففي عصر الصناعة، على سبيل المثال، كثيرًا ما كان الناس يستخدمون تعبيرات مثل:

«انفجر غضبه»

«أعمل بكامل طاقتي»

«بيئة كثيرة الضغوط»

«كان عليَّ أن أنفَس عن غضبي.»

واليوم نسمع تعبيرات مثل:

«برمجي حمضي النووي للقيام بذلك بهذه الطريقة.»

«قوانيننا عبارة عن خوارزميات لإدارة مجتمعنا»

«دماغي هو الجهاز المادي وعقلي هو البرنامج»

«تعطل دماغي، وأحتاج إلى إعادة تشغيله.»

وكما هو الحال في عصر الصناعة، تكشف التعبيرات الجديدة في عصر الكمبيوتر عن ثقافتنا الشعبية أكثر مما تكشف عن التكنولوجيا نفسها.

مثل الإله يانوس لدى الرومان، للتفكير الحوسبي وجهان؛ أحدهما: ينظر إلى الماضي ويفسّر كل ما حدث، والآخر: ينظر إلى المستقبل ويُخبرنا بما يمكن تصميمه. ونستدعي الوجهين عندما نريد من أجهزة الكمبيوتر أداء مهامَّ من أجلنا. من الوجه الذي ينظر إلى الماضي، نحتاج إلى فهم آلية عمل أجهزة الكمبيوتر، وكيفية التحكم فيها بواسطة الخوارزميات، وكيف يُمكننا التعبير عن الخوارزميات بإحدى اللغات البرمجية، وكيف يُمكننا دمج العديد من وحدات البرامج في نظم عاملة. أما من الوجه الذي ينظر إلى المستقبل، فنحن نحتاج إلى الإدراك لفهم السياق الذي يعمل فيه مُستخدمو برنامجنا. نريد أن يكون برنامجنا قيمًا لهم ولا يُسبب لهم أو لبيئتهم أي ضرر. ومن ثَمَّ، يُرشدنا التفكير الحوسبي إلى فهم التكنولوجيا المتاحة لنا وتصميم البرامج لإنجاز مهمةٍ أو حلٍّ مشكلة.

إنَّ التفكير الحوسبي ليس مجرد شيءٍ يجب على المُبرمجين معرفته، بل هو أيضًا أداة تفكير لفهم عالمنا الاجتماعي المُشبع بالتكنولوجيا. فهو يَزِيد من وعينا بكيفية عمل أدواتنا

الرقمية اليومية، ويدعم أخلاقياتنا الإلكترونية، ويُعزز مرونتنا ضد مختلف التهديدات مثل المحاولات القائمة على الخوارزميات لتوجيه سلوكنا، الاجتماعي في نشر المعلومات وتداولها بسرعة البرق، وتحليل تحركاتنا عن طريق جمع كميات هائلة من البيانات على نطاق واسع. علاوة على ذلك، فقد غيّر التفكير الحوسبي بشكل لا رجعة فيه أدوات العلم وأساليبه وأصوله. ومن ثمّ يُمكننا القول إن تعلّم التفكير الحوسبي له العديد من الفوائد التي تتجاوز نطاق البرمجة.

إذا حاولت فهم المقصود بالتفكير الحوسبي من خلال وسائل الإعلام، فسوف تجدها تتحدّث عن حلّ المشكلات باستخدام الخوارزميات، وأيضاً عن القدرة على التفكير بالعديد من مستويات التجريد اللازمة لحل المشكلات. كما ستجد صوراً لأطفال سعداء يستمتعون بالبرمجة وبممارسة الألعاب التي يقلّدون فيها الخوارزميات. في الواقع، تعلّم معلّمونا الكثير عن التفكير الحوسبي من تدريس الحوسبة للأطفال، وطوّروا طرقاً رائعة لتدريس مبادئ الحوسبة الأساسية للمبتدئين. وفي هذا الكتاب، سوف نُطلق على هذا المجال اسم «التفكير الحوسبي للمبتدئين».

لكن رؤى ومناقشات التعليم من مرحلة رياض الأطفال إلى الثانوية بالكاد تخدش سطح التفكير الحوسبي. على المستويات الأكثر تقدماً، يتعلق التفكير الحوسبي بتصميم الأجهزة والشبكات ونُظم التخزين ونظم التشغيل والسحابة الإلكترونية. وقد استُخدمت الممارسات القديمة في الاستعانة بفرق بشرية لإجراء الحسابات الكبيرة، وتنظيم خطوط الإنتاج في التصنيع، وتوجيه المُشترعين، وإرساء قواعد البيروقراطيات. وقد طوّر التفكير الحوسبي أنماطاً تتوافق مع المجالات الرئيسية التي تلعب فيها الحوسبة دوراً حاسماً، مثل الذكاء الاصطناعي وتحليل البيانات الضخمة وهندسة البرمجيات وعلم الكمبيوتر. سنُريك كل هذا من خلال إلقاء الضوء على أنواع التفكير الحوسبي اللازمة للتعامل مع هذه الأبعاد المختلفة للحوسبة. وسوف نحتاج إلى نوعٍ أكثر تقدماً بكثير من التفكير الحوسبي للتعامل مع هذه المجالات. وسوف نُطلق على هذا المجال في الكتاب «التفكير الحوسبي للمحترفين».

في بعض الأحيان، يُصوّر التفكير الحوسبي على أنه نهج عام لحل المشكلات. فنجد وسائل الإعلام الجماهيرية تُخبرنا بأننا إذا أخذنا بعض الدورات التدريبية في البرمجة، فسنتمكّن من حل المشكلات في أي مجال. ليت الأمر كان كذلك! إن قدرتك على حلّ مشكلةٍ لشخص ما تعتمد على فهمك للسياق الذي تُوجد فيه المشكلة. على سبيل المثال: لا يمكنك إنشاء محاكاة للطائرات أثناء الطيران دون فهم ديناميكيات الموائع. ولا يمكنك

برمجة عمليات البحث في قواعد بيانات الجينوم دون فهم بيولوجيا الجينوم وأساليب جمع البيانات. إذن فالتفكير الحوسبي قوي، ولكنه ليس حلاً سحرياً لكل المشاكل. يُبرز لنا التفكير الحوسبي فرقاً جوهرياً في طرق معالجة البشر والآلات للمعلومات. فالآلات تستطيع معالجة المعلومات بإجراء مليارات أو تريليونات العمليات الحسابية في الثانية، في الوقت الذي يُحسّن فيه البشر صنْعاً إذا أُجروا عمليةً حسابية واحدة في الثانية. وتقوم الآلات بالمعالجة دون فهم للبيانات التي تُعالجها، بينما يفهمها البشر ويمكنهم تصحيح الأخطاء الواردة فيها أثناء معالجتها. ويمكن للآلات أن تُحوّل خطأً في الخوارزمية إلى كارثة باهظة التكاليف قبل أن يتمكن أي إنسان من التصرف. وقد درس المفكرون في فلسفة العقل وعلم النفس العصبي وعلوم الإدراك والذكاء الاصطناعي هذه الاختلافات وأظهروا لنا الفرق الشاسع بينهما. وعلى الرغم من أن بعض المهام البشرية مثل البحث والفرز يمكن تسهيلها بتطبيق الخوارزميات عليها، فإن معظم التفكير الحوسبي في الصورة الكبرى يركز على الحوسبة الآلية.

فكّر لحظةً في مسألة السرعة. يمكن لجهاز الكمبيوتر العادي، في ثانية واحدة، إجراء مليار عملية حسابية ورسم صورة مُعقدة على الشاشة. أما الإنسان فسوف يحتاج إلى ١٠٠ عام لإجراء الخطوات نفسها بالسرعة البشرية. ومن الواضح أن البشر يرسمون الصور بشكل أسرع بكثير من ذلك، لكن مُصممي الآلات لم يحاكو هذه القدرة البشرية بعد. لو لم يتلقَّ البشر أي مساعدة من أجهزة الكمبيوتر، لما كانت لدينا رسومات في الوقت الفعلي. وتقريباً كلُّ ما نرى البرامج تقوم به قد أصبح ممكناً بفضل السرعات المذهلة لأجهزة الكمبيوتر. هذه الآلات، وليس البشر الذين يُنفذون الخوارزميات بأنفسهم، هي السبب في ثورة الكمبيوتر. تقوم أجهزة الكمبيوتر بما هو مستحيل بالنسبة إلى الإنسان.

قد يُشعرك هذا بالإثارة، ولكنه يجب أن يُشعرك أيضاً بالخوف. يتم التحكم في الطائرات الحديثة بواسطة شبكات من أجهزة الكمبيوتر تقوم بمليارات العمليات الحسابية في الثانية. ويمكن أن يتسبّب خطأ واحد في إحدى الخوارزميات في أن يُرسل نظام التحكم الطائرة إلى دوامة مُميتة قبل أن يتمكن الطيار البشري من التصرف. فقد أُجهزت بعثات أبولو الأولى وبعثات المريخ الأحداث وفشلت بسبب أخطاء في برامجها. يمكن أن تكون الأخطاء في الخوارزميات قاتلةً ومكلفة. فكيف يُمكننا أن نثق بالخوارزميات التي تُشغل النظم المهمة وأن نعرف بأنها ستعمل بشكل صحيح، وتثمر عن نتائج إيجابية وتقلل احتمالية حدوث أضرار؟ نحن بحاجة إلى تفكير واضح لمساعدتنا في إيجاد طريقنا عبر

هذه المتاهة المعقدة. وهذا يتطلب شكلاً متقدماً من التفكير الحوسبي لا يتم تعلّمه من ألعاب المحاكاة الخاصة بالأطفال. «التفكير الحوسبي للمُحترفين» أمر بالغ الجدية والأهمية. سوف نتناول في هذا الكتاب موضوع التفكير الحوسبي بجميع أنواعه، بداية من التفكير الحوسبي للمبتدئين وحتى المُحترفين، وفي المجالات الفرعية الكبرى مثل هندسة البرمجيات والعلوم الحوسبيّة. فهدفنا هو وصف التفكير الحوسبي بكلّ ما فيه من ثراء واتّساع وعمق. نحن نريد الاحتفاء بعمل المُحترفين الخبراء الذين يتصدّون للتحديات الصعبة لإنجاح النظم المعقدة بأمان وفاعلية، وأنواع التفكير التي يُقدّمونها والتي مكّنتهم من تحقيق سجلّ حافل. ونريد أيضاً الاحتفاء بعمل المُعلمين الخبراء الذين يعملون على تسهيل الخطوات الأولى في التفكير الحوسبي في مرحلة رياض الأطفال وحتى مرحلة التعليم الثانوي، ووضع الأساس لتزويد الجميع بالوسائل التي تُمكنهم من التكيّف مع العالم الرقمي. إن التفكير الحوسبي الأساسي للمبتدئين والتفكير الحوسبي المتقدّم للمُحترفين يعملان معاً لإنتاج نسيجٍ غني من الفكر الحوسبي.

بيتر جيه دنينج

ساليناس، كاليفورنيا، أغسطس ٢٠١٨

ماتي تيدري

يوينسو، فنلندا، أغسطس ٢٠١٨

الفصل الأول

ما المقصود بالتفكير الحوسبي؟

الخوارزمية هي مجموعة من القواعد للحصول على مُخرجات مُعينة من مُدخلات مُعينة. ويجب أن تكون كلُّ خطوة محددةً بدقة بحيث يمكن ترجمتها إلى لغة الكمبيوتر وتنفيذها بواسطة الآلة.

دونالد كنوث (١٩٧٧)

ما هو الكمبيوتر؟ معظم الناس سيُجيبون عن هذا السؤال بأن الكمبيوتر عبارة عن صندوقٍ أسودٍ إلكتروني يقوم بأشياء مُذهلة من خلال جمع البيانات وتخزينها واسترجاعها وتحويلها، جميع أجهزتنا وأدواتنا تقريباً عبارة عن أجهزة كمبيوتر: الهواتف، والأجهزة اللوحية، وأجهزة سطح المكتب، وصفحات الويب، وساعات اليد، وأجهزة الملاحة، والترمومترات، والأجهزة الطبية، وساعات الحائط، وأجهزة التلفزيون، وأجهزة تشغيل أقراص الفيديو الرقمية (دي في دي)، وشبكات الواي فاي. وخدماتنا أيضاً عبارة عن برامج: متاجر الكتب، ومتاجر التجزئة، والبنوك، ووسائل النقل، وخدمة «أوبر»، وحجوزات الفنادق، وخدمة «إير بي إن بي»، وصناعة الأفلام والترفيه، و«دروب بوكس»، والدورات التدريبية عبر الإنترنت، وعمليات البحث في جوجل، وكلها تقريباً تديرها أجهزة كمبيوتر غير مرئية عبر شبكة عالمية غير مرئية تُسمى «السحابة الإلكترونية». لقد أثمرت أجهزة الكمبيوتر عن فوائد هائلة؛ منها الوظائف الجديدة، وإمكانية الوصول إلى المعلومات، والتنمية الاقتصادية، والدفاع الوطني، وتحسين قطاع الصحة، وغيرها الكثير من الفوائد. كما أنها جلبت مخاوف مُثيرة للقلق؛ ومنها فقدان الوظائف، والعملة، وانتهاك الخصوصية، والمراقبة، وغيرها الكثير. يبدو أن كل شيء يمكن رقمته تتم رقمته وأجهزة الكمبيوتر تُخزن هذه المعلومات وتحوّلها في كل مكان. إننا نشهد حقاً ثورة في مجال الكمبيوتر.

كيف نُفكر في هذا كله؟ ما الذي نحتاج إلى فهمه حول أجهزة الكمبيوتر؟ ماذا يجب أن نفعل لجعل الكمبيوتر يعمل في خدمتنا؟ كيف تُشكل أجهزة الكمبيوتر الطريقة التي نرى بها العالم؟ ما الجديد الذي نراه؟ ما دور البرمجة؟ ما الأشياء التي لا تتقنها أجهزة الكمبيوتر؟

(١) قوة الحوسبة وقيمتها

يوفر التفكير الحوسبي بعض الإجابات عن هذه الأسئلة. يركز معظم التفكير الحوسبي تحديدًا على معرفة كيفية جعل الكمبيوتر يؤدي مهمةً بالنيابة عنا؛ أي على كيفية التحكم في جهاز إلكتروني معقد لإنجاز مهمة بشكل يُعتمد عليه دون التسبب في ضرر أو أذى. والخوارزميات هي الخطوات التنفيذية التي تُحدد كيفية قيام الكمبيوتر بالمهمة. وعلى الرغم من أن البشر يمكنهم تنفيذ الخوارزميات، فإنهم لا يستطيعون القيام بذلك بنفس سرعة الآلة؛ إذ يمكن لأجهزة الكمبيوتر الحديثة القيام بتريليون خطوة في الوقت الذي يستغرقه الإنسان للقيام بخطوة واحدة. ويكمن السر ببساطة في أن هناك آلة تُنفذ أعدادًا ضخمة من عمليات الحوسبة البسيطة جدًّا بسرعة هائلة. والبرامج هي الوسيلة لتنفيذ ذلك: الخوارزميات المُشفرة بلغاتٍ مُحددة الغرض تترجم إلى تعليمات آلة تتحكم في الكمبيوتر. لكن التفكير الحوسبي يتجاوز الأتمتة. فقد أضحت المعلومات وعمليات الحوسبة طريقةً لفهم الظواهر الطبيعية والاجتماعية. وكثير من التفكير الحوسبي اليوم موجّه نحو تعلّم كيفية عمل العالم. كذلك ينظر عدد متزايد من علماء الأحياء والفيزياء والكيمياء وعلماء آخرين إلى موضوعاتهم البحثية من خلال عدسة الحوسبة؛ وكذلك يفعل المهنيون في مجال الفنون والعلوم الإنسانية والاجتماعية. وقد أصبحت المحاكاة باستخدام أجهزة الكمبيوتر تُمكننا اليوم من إجراء تجارب افتراضية كانت مستحيلةً في الماضي. كذلك يقدم «التفسير المعلوماتي» للعالم أدوات مفاهيمية وتجريبية لا يُقدمها أي نهج آخر.

كما ينصحن التفكير الحوسبي بشأن المهام التي لا تستطيع أجهزة الكمبيوتر القيام بها في الوقت المنظور. أو لن نستطيع القيام بها على الإطلاق؛ إذ إن بعض المهام مُستحيلة بالنسبة إلى أجهزة الكمبيوتر. فالعديد من المشكلات الاجتماعية والسياسية والاقتصادية خارج نطاق أجهزة الكمبيوتر. ومن خلال فهم حدود الحوسبة، يُمكننا تجنب الوقوع في فخ اللجوء إلى تكنولوجيا الحوسبة لحلّ هذه المشكلات.

من الواضح أن تصميم برنامج أو آلة للقيام بالكثير في وقتٍ قصير جداً مهمة شديدة الصعوبة والتعقيد تتطلب طريقة تفكير خاصة بها إذا أردنا أن نكون واثقين من أداء الآلة للمهمة المطلوبة دون خطأ. وفي الواقع، لقد اتضح أن فهم المُستخدمين، وتصميم النظم خصوصاً لهم، من التحديات الكبرى للحوسبة الحديثة. وهكذا يكون التصميم هو أحد الاهتمامات الرئيسية للتفكير الحوسبي.

(٢) تعريف التفكير الحوسبي

أصبح التفكير الحوسبي من المصطلحات الرائجة التي تحمل تعريفات متعددة. وقد لخصنا هذه التعريفات المتعددة في تعريف واحد وهو الذي استخدمناه طوال الكتاب:

التفكير الحوسبي هو الممارسات والمهارات العقلية من أجل:

- تصميم عمليات الحوسبة التي تجعل أجهزة الكمبيوتر تقوم بالمهام من أجلنا.
- شرح العالم وتفسيره على أنه مجموعة مُعقدة من العمليات المعلوماتية.

يعكس الجانب التصميمي المنظور الهندسي للحوسبة، وتُبنى فيه أساليب وآلات لمساعدة الآخرين. أما جانب التفسير فيعكس المنظور العلمي للحوسبة، حيث يسعى الناس إلى فهم كيفية عمل الحوسبة وكيف تتجلى في العالم. ويُبرز التصميم الانغماس في المجتمع الذي تتم مساعدته، بينما يُبرز التفسير المراقبة غير المُتحيزة للعالم من الخارج. من حيث المبدأ، من الممكن تصميم عمليات الحوسبة دون تفسيرها، أو تفسيرها دون تصميمها. أما في الواقع، فهذان الجانبان يسيران جنباً إلى جنب.

عمليات الحوسبة هي سلسلة مُعقدة من العمليات الحسابية الرقمية والمعالجات الرمزية. ومن أمثلة العمليات الحسابية الرقمية العمليات الأساسية من جمع وطرح وضرب وقسمة، وأيضاً الدوال المثلثية الأساسية (الجيب وجيب التمام والظل). ومن أمثلة المعالجات الرمزية المقارنة المنطقية للأرقام أو الرموز، واتخاذ قرارات بشأن التعليمات التي يجب تنفيذها بعد ذلك، أو استبدال سلسلة من الأحرف والأرقام بأخرى. ويمكن إجراء عمليات حوسبة مُذهلة عند ترتيب تريليونات من هذه العمليات البسيطة على النحو الصحيح؛ على سبيل المثال: التنبؤ بطقس الغد، أو تحديد مكان الحفر بحثاً عن النفط، أو تصميم أجنحة طائرة ذات رفع كافٍ للطيران، أو إيجاد الأماكن المادية التي من المرجح أن يزورها شخص ما، أو استدعاء سيارة أجرة، أو استنتاج الشخصين المتوافقين اللذين سينجح زواجهما.

أجهزة الكمبيوتر هي الوكيل الذي ينفذ العمليات المُضمَّنة في عمليات الحوسبة. فهي تتبع برامج التعليمات لإجراء العمليات الحسابية والمنطقية. ويمكن أن تكون أجهزة الكمبيوتر بشرًا أو آلات. فالبشر يستطيعون اتباع البرامج، لكنهم ليسوا بأي حالٍ من الأحوال في سرعة الآلات، كما أنهم عُرضة للخطأ بدرجة أكبر بكثيرٍ من الآلات. وكذلك فإن الآلات تستطيع أداء عمليات الحوسبة التي تتجاوز قدرات الإنسان بكثير.

نستخدم كلمة «مهمة» للإشارة إلى أي عملٍ يعتبره شخصٌ ما ذا قيمة. ويتطلَّع الكثير من الناس اليوم إلى أجهزة الكمبيوتر (أو بالأحرى إلى عمليات الحوسبة التي تؤدِّيها أجهزة الكمبيوتر) لإنجاز المهام. فهم يسعون إلى أتمتة المهام التي لا يمكن القيام بها دون مساعدة الآلة. وقد أصبحت أجهزة الكمبيوتر في الوقت الحالي قادرةً على إنجاز بعض المهام الروتينية بكفاءة عالية للغاية، لدرجة أن فقدان الوظائف بسبب الأتمتة أصبح من المخاوف الاجتماعية التي لا يُستهان بها.

جدير بالذكر أن هناك فرقًا بين «إنجاز المهمة» وبين الأتمتة. يمكن أتمتة المهام الروتينية المحددة بدقة، أما المهام غير المحددة بدقة مثل «التعامل مع مصدر القلق» فلا يُمكن أتمتتها. ويمكن أن يساعد التفكير الحوسبي في المهام التي لا يمكن أتمتتها. وسوف نناقش في الفصل الذي يتناول التصميم نوعَ التفكير الحوسبي الذي يقوم بهذا. من الواضح أن هناك مهارةً تفكيرٍ خاصةً مطلوبة لتصميم برامج وآلاتٍ قادرة على إجراء عمليات حوسبة هائلة، وفهم العمليات المعلوماتية الطبيعية من خلال الحوسبة. هذه المهارة — التفكير الحوسبي — ليست مجرد مجموعة من المفاهيم للبرمجة. وإنما يشمل التفكير الحوسبي طُرُق التفكير والتدريب التي تُشخِّذ وتُصقِّل بالممارسة. إن التفكير الحوسبي عبارة عن مجموعة غنية جدًا من المهارات، وسوف نُلخص في نهاية هذا الفصل الأبعاد الستة للتفكير الحوسبي التي ستُقابلها في هذا الكتاب: الآلات، والأساليب، وتعليم الحوسبة، وهندسة البرمجيات، والتصميم، والعلوم الحوسبيَّة.

(٣) الأمنيات غير الواقعية

في ظلِّ حماسنا للتفكير الحوسبي، يجدر بنا أن نحذَر من الوقوع في براثن التمنيِّ. وربما كانت أولى الأمنيات وأكثرها شيوعًا هي أننا نستطيع أن نجعل أجهزة الكمبيوتر تقوم بأيِّ عملٍ يمكننا تصوُّره. وهذه الأمنية لا يمكن تحقيقها؛ لأنه يوجد العديد من المهام يستحيل لأجهزة الكمبيوتر القيام بها. على سبيل المثال: لا تُوجد خوارزمية يمكنها فحص

مثال ثالث للأمنيات غير الواقعية هو الاعتقاد بأن أجهزة الكمبيوتر ليست ضرورية للتفكير الحوسبي؛ وأننا يمكن أن نُفكر في كيفية حلّ المشكلات باستخدام الخوارزميات دون الاهتمام بأجهزة الكمبيوتر التي تُشغل الخوارزميات. ولكن هذا ليس صحيحًا. عندما لا يكون لدى الكمبيوتر ذاكرة كافية لحفظ جميع البيانات، ستبحث عن طرق لتقسيم مشكلتك إلى مجموعات فرعية بما يتناسب مع المساحة المتوفرة. وعندما لا يكون لدى معالجٍ واحدٍ قوةً معالجةً كافية، ستبحث عن كمبيوتر يحتوي على معالجات متعددة متوازية، وخوارزميات تقسّم عمليات الحوسبة بينها. وعندما يكون الكمبيوتر بطيئًا جدًا، ستبحث داخله لمعرفة المكوّن الذي يُسبب البطء، وستلجأ إما إلى ترقية هذا المكوّن أو العثور على خوارزمية جديدة لا تستخدم هذا المكوّن. وحتى إذا كان الكمبيوتر الخاص بك يحتوي على ذاكرة كافية وقوة معالجة كافية ولا تُوجد به مكوّنات بطيئة، فإن هناك أشياء أخرى يمكن أن تعوق حل المشكلات، ولا سيما سرعة الساعة الداخلية، التي تُحدد سرعة الجهاز لأداء خطوات عمليات الحوسبة بطريقة منظمة ويمكن التنبؤ بها. لكن بعض الأجهزة الجديدة، ولا سيما أجهزة الكمبيوتر الكمية والشبكات العصبية، ليس لديها ساعات داخلية، فكيف نُفكر في برمجتها؟

مثال رابع للأمنيات غير الواقعية هو الاعتقاد بأن الكمبيوتر ذكي. إذا كنت غير دقيق في ترجمة الخطوات البشرية إلى خطوات برمجية، فإن عملية الحوسبة سوف تحتوي على أخطاء يمكن أن تُسبب كوارث. إن أجهزة الكمبيوتر غبية بشكل لا يُصدّق. إنها تؤدي خطوات آلية بسرعة كبيرة بلا تفكير، ولكن ليس لديها أي فهم لما تعنيه هذه الخطوات والأخطاء الوحيدة التي يُمكنها تصحيحها هي تلك التي تتوقعها أنت، ومن ثمّ توفر لها خوارزميات تصحيحية. إذن فأنت مصدر الذكاء، وما يفعله الكمبيوتر هو أنه يُعزّز ذكاءك فحسب، لكنه لا يملك أي ذكاء خاصّ به.

نصيحتنا لك هي أن تدنو من التفكير الحوسبي بتواضع. واعلم أنه مهارة مكتسبة. فأدغمنا بطبيعتها لا تفكر بشكلٍ حوسبي. وتذكّر دائمًا قدرات أجهزة الكمبيوتر والخوارزميات على إنجاز المهام، والحاجة إلى تعلّم شيءٍ ما حول مجال التطبيق الذي تريد تصميمه، واعتماد الحوسبة على أجهزة الكمبيوتر، وافتقار الأجهزة تمامًا إلى الذكاء.

(٤) ظهور التفكير الحوسبي على مدار الألفية

قد يبدو أن التفكير الحوسبي هو نتاج عصر الكمبيوتر الإلكتروني الذي بدأ في الأربعينيات من القرن الماضي. حسنًا، هذا ليس صحيحًا على الإطلاق. قبل عصر الكمبيوتر الحديث،

كانت هناك مهنة يمتحنها خبراء مدرّبون رياضياً على إجراء العمليات الحسابية المعقدة وهم في هيئة فِرَق. وكان يُطلَق عليهم: «الحاسب». ولم يكن هؤلاء هم أول من فعلوا ذلك بأي حالٍ من الأحوال؛ إذ يعود تاريخ مصطلح «الحاسب» (أو الكمبيوتر كما يطلق عليه الآن)، بمعنى «مَن يحسُب»، إلى أوائل القرن السابع عشر. وكانت أول آلات الحوسبة الإلكترونية تُسمَّى الحواسِب الآلية لتمييزها عن الحواسِب البشرية. وكان الحواسِب من البشر، ولا سيما قادة فرق الحواسِب البشرية، ينخرطون بوضوح في التفكير الحوسبي. لذلك، كان العديد من جوانب التفكير الحوسبي موجوداً قبل الحواسِب الإلكترونية، أو ما نُطلق عليه أجهزة الكمبيوتر الإلكترونية. وقبلها بكثير.

سُجِّلَت الأشكال البدائية من التفكير الحوسبي، مثل أساليب الحساب، بداية من الفترة ما بين عامي ١٨٠٠ و ١٦٠٠ قبل الميلاد بين البابليين، الذين كتبوا خطوات عامة لحل المسائل الرياضية. تتميز هذه الخطوات التي تتبع القواعد بميزات تُطلق عليها اليوم، من منظورنا الحالي، أشكالاً من التفكير الحوسبي. وبالمثل، فإن المهندسين المصريين الذين بنوا الأهرام بدءاً من حوالي عام ٢٧٠٠ قبل الميلاد كانوا يعرفون الكثير عن الهندسة، وكانوا قادرين على حساب أبعاد الحجارة وزواياها لكل جزءٍ من الهرم، وقوة الحبال والرافعات والبكرات المُستخدمة لتحريك الحجارة إلى مكانها. إذن فالحوسبة ممارسة بشرية قديمة. على مرّ الزمن منذ العصور القديمة، سعى علماء الرياضيات إلى صياغة خطوات تنفيذية لعمليات حسابية تزداد تعقيداً شيئاً فشيئاً، وتتجاوز حساب المعاملات التجارية وهندسة الإنشاءات إلى حساب المثلثات والتوقعات الفلكية والملاحاة الفلكية وحل المعادلات الجبرية، وصولاً في النهاية إلى الحوسبة باستخدام حساب التفاضل والتكامل لنيوتن ولايبنتس. ومن خلال صياغة الخطوات التنفيذية للحوسبة، أتاح علماء الرياضيات خبرتهم لغير الخبراء الذين كان عليهم ببساطة اتّباع تعليمات إجراء العمليات الحسابية البسيطة بالترتيب الصحيح. تُسمَّى فئة خاصة من هذه التعليمات اليوم «خوارزمية»؛ وهو مفهوم أساسي في الحوسبة الحديثة. يرجع الأصل في مصطلح «خوارزمية» إلى عالم الرياضيات الفارسي محمد بن موسى الخوارزمي الذي ناقش، حوالي عام ٨٠٠ الميلادي، كيفية صياغة الخطوات الرياضية وأعطى أمثلةً مثل إيجاد القاسم المشترك الأكبر لمجموعة من الأعداد.

لدينا، نحن البشر، ميلٌ إلى أتمتة الإجراءات الروتينية. لذلك طبقنا هذا على خطوات الحوسبة؛ إذ سعى المخترعون إلى ابتكار آلاتٍ تقوم بأتمتة عمليات الحوسبة بهدف تحقيق سرعة أكبر وأخطاء أقل. واتضح أن إنشاء آلات لتنفيذ هذه الخطوات كان أصعب من تحديد

الخطوات نفسها. صمّم باسكال آلة حسابية في القرن السابع عشر لتقوم بالجمع والطرح، وصنعها بنفسه. ولكن لم تكن الآلة تستطيع الضرب إلا في حالة الاستعانة بمُشغل بشري يفهم الجمع المُتكرر. ولم تستطع القسمة أيضًا. كذلك اخترع نابير اللوغاريتم، الذي أصبح أساسًا للمسطرة اللوغاريتمية المنزلقة؛ وهي أداة للمساعدة في الحساب ظلت تُستخدم حتى النصف الثاني من القرن العشرين. ولم تستطع هذه الأداة الجمع أو الطرح. وفي عام ١٨١٩، صمم بابيدج آلة من التروس والأعمدة والعجلات يُمكنها حساب جداول الأرقام الحسابية مثل اللوغاريتمات. وفي تعداد الولايات المتحدة لعام ١٨٩٠، قامت آلات البطاقات المثقبة التي ابتكرها هوليريث بجدولة كميات كبيرة من البيانات، وأصبحت شركة «آي بي إم»، بعد تأسيسها في عام ١٩٢٤، ثرية من جرّاء بيع هذه الآلات. في عشرينيات القرن الماضي، صمّم المهندسون أجهزة الكمبيوتر التناظرية لحساب الدوال المُستمرة عن طريق محاكاتها بالدوائر والتروس. ومن الواضح أن مُصممي أجهزة الكمبيوتر التناظرية والرقمية كانوا يستخدمون التفكير الحوسبي. ولكن حتى فكرة الكمبيوتر التناظري قديمة، تعود إلى الساعة الفلكية اليونانية، وهي جهاز ميكانيكي كان يُستخدم لحساب مواقع الكواكب في عام ١٠٠ قبل الميلاد.

لطالما تطلّبت الحوسبة عبر العصور تفكيرًا حوسبيًا لتصميم خطوات حوسبة وآلات لأتمتتها. لم يكن الدافع وراء السعي الطويل إلى ابتكار آلات الحوسبة هو إجراء عملية الحوسبة بسرعة أكبر فحسب، بل أيضًا القضاء على الأخطاء البشرية، والتي كانت شائعة عندما كان البشر، الذين يشعرون بالملل سريعًا أو يتشتتون بسهولة، يقومون بالعديد من العمليات الحسابية المكررة. واعتقد المُصمّمون أن الآلات المؤتمتة ستتغلب على مصادر الخطأ في العمليات الحسابية. لكننا اليوم نعرف الحقيقة: في حين أن الآلات قضت على بعض أنواع الأخطاء، فقد ظهر نطاق واسع من الأخطاء الجديدة. وأصبحت آلات الحوسبة معقدة للغاية، لدرجة أننا لا نعرف إلى أي مدى يُمكننا الوثوق بها.

ازدادت أصول التفكير الحوسبي تعقيدًا على مرّ تلك القرون العديدة. ولكن مُصممي الكمبيوتر والمبرمجين لم يجدوا حافزًا لتطوير التفكير الحوسبي كمطلبٍ مهني إلا عندما أصبح الكمبيوتر الإلكتروني صناعة في خمسينيات القرن الماضي. وانجذب هؤلاء المحترفون إلى البرمجيات لأنهم يستطيعون بسهولة تغيير وظيفة الآلة عن طريق إعادة برمجة البرنامج. سعت صناعة الحوسبة الناشئة إلى المبرمجين والمهندسين المُدرّبين على التفكير الحوسبي والممارسة. وبحث مسئولو التعليم في كيفية التدريس لهؤلاء المبرمجين

والمهندسين. وورث مجال علوم الكمبيوتر، الذي ظهر بحلول عام ١٩٦٠، مسئولية تعريف التفكير الحوسبي وتدريبه.

نُقدّم في هذا الكتاب العديد من الأمثلة من تاريخ الحوسبة لتوضيح الاحتياجات التي لبّأها التفكير الحوسبي، والإمكانيات الجديدة للأمور التي أتاحها التفكير الحوسبي، والتغيّرات العقلية الهائلة التي تسبّب فيها التفكير الحوسبي فيما يتعلق بكيفية رؤيتنا للآتمتة والعلم والعالم. وعلى الرغم من أن التفكير الحوسبي، بوصفه طريقة تفكير، موجودٌ منذ آلاف السنين، فإن مصطلح «التفكير الحوسبي» جديد نسبيًا؛ إذ إن أول ظهور نعرفه له يعود إلى عام ١٩٨٠.

(٥) ظهور حركة تعليم التفكير الحوسبي في المدارس

تأسّس أول قسم لعلوم الكمبيوتر في جامعة بيردو في عام ١٩٦٢. وتطور علم الكمبيوتر الأكاديمي تدريجيًا رغم مواجهته العديد من التحديات. فقد كانت العديد من الجامعات متشككة في البداية في ما إذا كان المجال الجديد جديًا حقًا أو أكاديميًا بما فيه الكفاية؛ حيث بدا وكأنه فرع من فروع الهندسة الكهربائية أو الرياضيات التطبيقية. كما أن الكثير من أقسام علوم الكمبيوتر لم تُفتتح إلا بعد معارك سياسية حامية في جامعاتها. وعلى الرغم من هذه الصعوبات السياسية، فقد شهدت نموًا مطردًا. فبحلول عام ١٩٨٠، كان هناك حوالي ١٢٠ قسمًا لعلوم الكمبيوتر في الولايات المتحدة وحدها. واليوم يوجد في جميع الجامعات الكبرى والعديد من الجامعات الأصغر قسم واحد على الأقل لعلوم الكمبيوتر.

خلال السنوات الأربعين الأولى، تركّزت معظم اهتمامات ممارسي الحوسبة في جعل هذه التكنولوجيا تُؤتي ثمارها. وكان كل ما نعتبره اليوم تقنيةً أساسية يجب اختراعه وتصميمه واختباره وإعادة اختباره؛ خذ — على سبيل المثال — لغات البرمجة، ونظم التشغيل، والشبكات، والرسومات، وقواعد البيانات، والروبوتات، والذكاء الاصطناعي. وأصبح تصميم تقنيات حوسبة موثوقٍ بها هدفًا أساسيًا في مجال التفكير الحوسبي خلال هذه الفترة.

في الثمانينيات من القرن العشرين، شهدت الحوسبة انفتاحًا كبيرًا؛ حيث اتّجه العلماء في جميع المجالات إلى إدخال الحوسبة والتفكير الحوسبي في علومهم الأساسية. في البداية، استخدموا الحوسبة لمحاكاة النماذج النظرية الموجودة أو لجدولة البيانات التي يحصلون عليها من التجارب وتحليلها رقميًا. ولكنهم سرعان ما اكتشفوا أن التفكير القائم على

الحوسبة قد فتحت باباً لطريقة جديدة تماماً لتنظيم البحث العلمي، وأدى التفكير الحوسبي إلى الحصول على جوائز نوبل في اكتشافات كانت تُحير العلماء من قبل. وأعلن الكثيرون أن الحوسبة هي الركيزة الثالثة التي يُبنى عليها العلم، إلى جانب النظرية والتجربة. واتسع مجال التفكير الحوسبي ليشمل تصميم عمليات حوسبة في جميع مجالات العلوم، خاصة للرد على أسئلة «التحديات الكبرى» التي يطرحها الرُّوَاد في مختلف المجالات. وأعلنت كل مجالات العلوم في النهاية أنها تتضمن فرعاً حوسبياً، مثل الفيزياء الحوسبية، وعلم المعلومات الحيوية (أو الأحياء الحوسبية)، والكيمياء الحوسبية، والعلوم الإنسانية الرقمية، وعلم الاجتماع الحوسبي.

كان لدى علماء الكمبيوتر ردود فعل مختلطة على هذه التطورات. تذكراً للندوب التي لحقت بهم في معركة إنشاء أقسام علوم الكمبيوتر، كان البعض حساساً للغاية لصورة علوم الكمبيوتر في الأذهان وأرادوا التحكم فيها. واعتبر البعض أن حركة علوم الحوسبة ما هي إلا وسيلة يستولي بها العلماء على علوم الكمبيوتر بعد أن كانوا يُعربون في السابق عن تشكُّكهم فيها. ونتيجة لذلك، كان لدى علماء الكمبيوتر دافع قوي لمساعدة الجمهور على فهم تكنولوجيا الحوسبة ونظرية الحوسبة. وتعاون معلِّمو علوم الكمبيوتر مع المُعلِّمين من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي لوضع دورات محو الأمية في الكمبيوتر، ولكنها لم تكن شائعة جداً. في عام ٢٠٠٠ تقريباً، اقترح بعض رجال التعليم نهجاً أكثر تطوراً أطلقوا عليه: «الإلمام باستخدام تكنولوجيا المعلومات»، واختار مدرسو المدارس الثانوية كتاباً مدرسياً مشهوراً في هذا المجال. وحتى مع نجاح هذا النهج، لم يعتد دورات الكمبيوتر سوى عدد قليل من المدارس الثانوية. وواصل معلِّمو علوم الكمبيوتر البحث عن طرق لاختراق نظم المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي وتعريف جميع الطلاب بالحوسبة.

شهد عام ٢٠٠٦ نقطة تحوُّل عندما أعادت جانيت وينج، التي كانت تعمل آنذاك مساعدة مدير في المؤسسة الوطنية للعلوم بالولايات المتحدة، صياغة النهج من «الإلمام باستخدام تكنولوجيا المعلومات» إلى التفكير الحوسبي. وأشارت إلى أن التفكير الحوسبي هو أسلوب تفكيرٍ يحتاج الجميع إلى تعلُّمه في عصر الحوسبة. وفي المؤسسة الوطنية للعلوم، حشدت موارد كبيرة لتدريب المُعلِّمين، وترقية اختبار المستوى المُتقدم، وتصميم دورات جديدة في «مبادئ علوم الكمبيوتر» للكليات، وتعريف التفكير الحوسبي لقطاع التعليم من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي، وإصدار توصيات بمنهج دراسي

للمدارس من مرحلة رياض الأطفال إلى نهاية المرحلة الثانوية. وحققت هذه الحركة التي تحمل شعار «علوم الكمبيوتر للجميع» انتشاراً أكبر بكثير للحوسبة في المدارس من مرحلة رياض الأطفال إلى نهاية التعليم الثانوي مقارنة بأي من سابقتها.

جدير بالملاحظة أن تعريفات التفكير الحوسبي التي ظهرت بعد حركة التفكير الحوسبي لعام ٢٠٠٦ قد نُشرت للجمهور. ولكن العديد من التعريفات العامة ضيّقة الأفق جداً مقارنة بمفاهيم التفكير الحوسبي التي تطوّرت على مدار القرون السابقة من الحوسبة، لا سيما في ضوء تفسيرها من قِبَل صنّاع السياسات. وتُقدم وسائل الإعلام الرئيسية أحياناً وجهة نظر خاطئة عن نطاق الحوسبة وتأثيرها. ومن ثم فقد أدّت إلى أن يُقدّم الأشخاص غير المُلمّين بالحوسبة ادّعاءات مُبالغاً فيها حول قوة التفكير الحوسبي، مما سيؤدّي إلى تضليل الطلاب وغيرهم وتقديم وعودٍ حول الكمبيوتر لا يمكنهم الوفاء بها.

(٦) أهدافنا في هذا الكتاب

نهدفُ في هذا الكتاب إلى عرض الصورة الكاملة للتفكير الحوسبي ومبادئه فيما يخصّ الحوسبة، وتبديد المفاهيم الخاطئة حول نقاط قوة الحوسبة ونقاط ضعفها. تطوّر التفكير الحوسبي من أصوله القديمة التي تعود إلى أكثر من ٤٥٠٠ عام إلى حالته الحالية الشديدة التقدّم والمهنية. ولم يكن الدافع وراء السعي الطويل نحو ابتكار آلات الحوسبة عبر العصور هو تسريع عمليات الحوسبة فحسب، بل أيضاً القضاء على الأخطاء البشرية، التي كانت شائعةً عندما كان البشر السريعو الملل أو التشتت يقومون بالعديد من العمليات الحسابية المُكررة. وقد تطوّرت مهارة تفكير خاصة لتحقيق هذا الهدف.

أدى تطوير التفكير الحوسبي إلى ستة أبعاد مهمة تُميّز التفكير الحوسبي حالياً:

الأساليب: طوّر علماء الرياضيات والمهندسون أساليب للحوسبة والتفكير يمكن لغير الخبراء استخدامها ببساطة باتباع التعليمات.

الآلات: بحث المخترعون عن آلات لأتمتة إجراءات الحوسبة بهدف زيادة سرعة العمليات الحسابية والحد من الأخطاء البشرية في تنفيذ عمليات الحوسبة. وقد أدى ذلك في النهاية إلى اختراع الكمبيوتر الإلكتروني الرقمي الذي يستغل حركة الإلكترونات في الدوائر الكهربائية في إجراء عمليات الحوسبة.

تعليم الحوسبة: صاغ علماء الكمبيوتر في الجامعات علم الكمبيوتر لدراسة وترميز الحوسبة وطرق تفكيرها وممارستها للمؤسسات والشركات والعلوم والهندسة.

هندسة البرمجيات: وضع مطورو البرامج هندسة البرمجيات للتغلب على المشاكل المنتشرة المتعلقة بالأخطاء وعدم الموثوقية في البرمجيات، خاصة النظم البرمجية الكبيرة مثل نظم التشغيل والتطبيقات الرئيسية.

التصميم: يستجيب المصممون إلى اهتمامات مجتمعات المستخدمين وشواغلهم وممارساتهم وتاريخهم.

العلوم الحوسبية: أنشأ العلماء العلوم الحوسبية لإدخال الحوسبة في العلوم، ليس فقط لدعم جانبي النظرية والتجربة التقليديين، ولكن أيضاً لتقديم طرق جديدة ثورية لتفسير العمليات الطبيعية وإجراء عمليات التحقق العلمي.

هذه الأبعاد الستة أشبه بنوافذ مختلفة ننظر من خلالها إلى التفكير الحوسبي. وتُقدم كل نافذة منها زاوية رؤية معينة. وقد تكون بعض جوانب التفكير الحوسبي مرئية من نافذتين، ولكن كلاً منهما تُسلط الضوء على شيء مختلف.

في الفصول الستة التالية، سنفحص التفكير الحوسبي فيما يتعلق بكل بُعد من الأبعاد المذكورة أعلاه. ثم نُكمل بفصل شبه ختامي عن التفكير الحوسبي في التعليم العام الحديث، وفصل ختامي عن مستقبل التفكير الحوسبي.

الفصل الثاني: التفكير الحوسبي فيما يتعلق بالإجراءات الخوارزمية لأتمتة العمليات

الفصل الثالث: التفكير الحوسبي فيما يتعلق بأجهزة الحوسبة

الفصل الرابع: التفكير الحوسبي فيما يتعلق بنظرية الحوسبة والتخصص الأكاديمي

الفصل الخامس: التفكير الحوسبي فيما يتعلق بإنشاء نظم البرمجيات الكبيرة

الفصل السادس: التفكير الحوسبي فيما يتعلق بالتصميم لتلبية احتياجات الإنسان

الفصل السابع: التفكير الحوسبي وارتباطه بجميع العلوم

الفصل الثامن: تدريس التفكير الحوسبي للجميع

الفصل التاسع: مستقبل التفكير الحوسبي

ما المقصود بالتفكير الحوسبي؟

ونحن نسرد هذه الأبعاد لكي نُريك قوة التفكير الحوسبي والطرق التي قد يُساعدك بها في عملك مع أجهزة الكمبيوتر والحوسبة.

تطوّر التفكير الحوسبي من أصوله القديمة التي تعود إلى أكثر من ٤٥٠٠ عام إلى حالته الحالية الشديدة التقدّم والمهنية. ولم يكن الدافع وراء السعي الطويل نحو ابتكار آلات الحوسبة عبر العصور هو تسريع عمليات الحوسبة فحسب، بل أيضاً القضاء على الأخطاء البشرية.

الفصل الثاني

أساليب الحوسبة

إذا نشأت خلافات، فإن الجدل بين فيلسوفين لن يزيد عن الجدل بين اثنين من المحاسبين. ذلك لأنه يكفي عندئذ أن يأخذ كل منهما قلمه، ويجلس إلى لوح الكتابة، ثم يقول للآخر (في حضور شاهدٍ من أصدقائهما، إذا أحببنا ذلك): دعنا نحسب.

لايبنتس، في ترجمة راسل (١٩٣٧)

عندما كان بيتر في العاشرة من عمره، أخبره معلم الرياضيات ذو العينين اللامعتين أنه يستطيع قراءة أفكار الآخرين. سأل بيتر: «كيف يمكنك ذلك؟» فقال المعلم: «تعال، سأريك. فكّر في رقم. اضربه في ٢. أضف ٨. اقسم على ٢. اطرح الرقم الأصلي. الآن ركز بشدة في الإجابة. ها أنا ذا أراها. الإجابة هي ٤.» كان بيتر مندهشاً جداً وأصرّ على أن يُريه المعلم كيف يفعل ذلك. فقال المعلم: «حسنًا، إنها مجرد رياضيات. لنفترض أن س هو رقمك. ثم طلبت منك حساب: $(٢س + ٨) \div ٢ - س = ٤$. لقد طرَحَ رقمك الأولي. والإجابة دائماً نصف الرقم الذي أخبرتك أن تُضيفه.» حظي بيتر بالعديد من اللحظات الرائعة وهو يقرأ أفكار عائلته وأصدقائه بهذا الأسلوب. كما أنه أصبح مهووساً بالرياضيات والحوسبة.

هذا الأسلوب واحد من أساليب رياضية عديدة جرى تناقلها عبر العديد من الأجيال. مثل هذه الأساليب تكمن وراء مجموعة من «الحيل الأوتوماتيكية» التي يستخدمها السحرة، حيث يقود الساحر واحدًا من الجمهور عبر سلسلة من الخطوات إلى إجابة معروفة لدى الساحر، ويعتقد الجمهور أنها سرية. تكمن المهارات العقلية اللازمة لتحقيق هذه الحيلة في الخطوات الرياضية المعروفة للساحر ولكنها ليست معروفة للجمهور. وهي تنجح

مع أي ساحرٍ يتبع التعليمات، حتى لو لم تكن لدى الساحر فكرة عن سبب نجاح هذه التعليمات.

لقد جرى توريث العديد من الأساليب الأخرى ذات الأغراض الأكثر جديةً عبر العصور. ويرجع أحد أقدم الأساليب، الذي يُدرّس للعديد من تلاميذ المدارس اليوم، إلى عالم الرياضيات اليوناني إقليدس حوالي عام ٣٠٠ قبل الميلاد. قدّم إقليدس أسلوباً لإيجاد القاسم المشترك الأكبر (ق. م. أ.) لعددين، وهو أكبر عدد صحيح يقبل القسمة على كلا العددين. وجد إقليدس قاعدة اختزال ذكية من خلال ملاحظة أن القاسم المشترك الأكبر لعددين يقبل القسمة على الفرق بينهما. استخدم الرقم الأكبر بدلاً من الفرق بينهما، وكرّر ذلك حتى أصبحا متساويين. على سبيل المثال: ق. م. أ. (٤٨، ١٨) = ق. م. أ. (٣٠، ١٨) = ق. م. أ. (١٢، ١٨) = ق. م. أ. (٦، ١٢) = ق. م. أ. (٦، ٦) = ٦. استخدم هذا الأسلوب لاختزال الكسور. واليوم هو من بين الأساليب الأساسية الكامنة في التشفير.

من الأساليب الأخرى المشهورة التي تعود إلى اليونانيين أسلوب «غربال إراتوستينس»، المُستخدم لإيجاد جميع الأعداد الأولية حتى حدٍّ معين. يبدأ هذا الأسلوب بقائمة بجميع الأعداد الصحيحة من ٢ إلى الحد الأقصى. ثم يُستبعد كل مضاعفات العدد ٢، ثم كل مضاعفات العدد ٣، ثم العدد ٥، وهكذا. وبعد كل جولةٍ من الاستبعاد، سيظهر عدد أولي جديد، وفي الجولة التالية يُستبعد كل مضاعفات هذا العدد الأولي الجديد. وهذا أسلوب فعّال جداً لإيجاد الأعداد الأولية الصغيرة، وقد استخدم لإيجاد الأعداد الأولية الكبيرة للمفاتيح في نظم التشفير الحديثة.

كان اليونانيون مهتمين أيضاً بحساب مساحات الأشكال. وقاموا بذلك من خلال إيجاد طرقٍ لملء الأشكال بقوالب من الأشكال الأساسية البسيطة مثل المربعات أو المثلثات، ثم تقليل أبعاد القوالب واحداً تلو الآخر حتى تملأ الشكل الأصلي بالكامل وتُغطيه تقريباً. كان هذا الأسلوب، الذي سجّل لأول مرة في الفترة ما بين ٤٠٠ و ٣٥٠ قبل الميلاد، هو بداية ظهور أساليب أفضل قدّمت في حساب التفاضل والتكامل الحديث بعد ألفي عام.

استخدم العديد من علماء الرياضيات هذه الأساليب لإنشاء متسلسلات لا نهائية من حدود بسيطة والتي تتقارب حتى نهاية مُعينة. تمتلئ كتب الرياضيات بجداول المتسلسلات؛ حيث استعان بها علماء الرياضيات لاستخدام الصيغ المغلقة بدلاً من المتسلسلات الطويلة. من أمثلة ذلك المتسلسلة $1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$ ، والتي تقدم طريقةً لحساب قيمة π ، وتزداد الدقة كلما أدرجنا المزيد من الحدود.

أَتَقَنَّ حساب التفاضل والتكامل، الذي قَدَّمه نيوتن ولايبنتس كلُّ على حدة حوالي عام ١٦٨٠، فكرة تقريب الأجسام والمنحنيات باستخدام العمليات الحسابية الخاصة بالمتسلسلات اللانهائية. كانت الفكرة هي تمثيل الأشكال الهندسية والمنحنيات المستمرة بمكونات صغيرة جداً مُتصلة اتصالاً مباشراً فيما بينها، على سبيل المثال: كان الشكل يُملأ بمربعات صغيرة أو يُمثَّل المنحنى كمتوالية من القطع المُستقيمة الصغيرة التي ترتبط بالقطع المُستقيمة المجاورة لها وكأن بينها قوى تجاذب. ومن ثم، كان من الممكن إيجاد مقدار أكبر مثل المساحة أو الطول من خلال جمع هذه المكوّنات الصغيرة. عند السماح لحجم هذه المكوّنات أن يقترب من الصفر، فإن القيم الناتجة من هذه المجاميع اللانهائية تكون فعلية وليست تقريبية. تم تمثيل قاعدة الاتصال المباشر كمُشتق والمجموع كتكامل. بدافع من حساب التفاضل والتكامل، أوجد علماء الرياضيات قيمة الدوال بتقسيم الزمان والمكان إلى تزايدت صغيرة تم إحصاؤها كـ «شبكة»، وحسبوا قيمة الدالة بشكلٍ تكراري عند كل نقطة من نقاط الشبكة. وأفادت هذه الطريقة الفيزياء أيما إفادة، حيث تم التعبير عن العديد من النماذج الرياضية للظواهر الفيزيائية كمعادلات تفاضلية يمكن حسابها على شبكات منتهية.

من الأساليب الرياضية المشهورة الأخرى أسلوب الحذف لجاوس لحل نظم المعادلات الخطية. عمل جاوس بهذا الأسلوب في منتصف القرن التاسع عشر، على الرغم من أن علماء الرياضيات الصينيين عرفوه في عام ١٧٩ قبل الميلاد. تُستخدم أشكال فعّالة جداً من هذه الخوارزمية في شاشات الرسومات الحديثة لعرض الأجسام الثلاثية الأبعاد في الوقت الفعلي.

توضح هذه الأمثلة القليلة الفوائد الجمة التي تعود علينا من الكنز الهائل من الأساليب التي انتقلت إلينا على مرّ قرون عديدة. يُمكننا أن نستنتج من فحص هذه الأساليب أن العديد منها كان متطوراً للغاية. وكان هدفها هو معرفة ما يجب القيام به في عمليات الحوسبة المُعقدة، كسلسلة من الخطوات. في البداية، كانت هذه الخطوات يتم تنفيذها من قِبَل علماء الرياضيات، ولكن مع إجراء التحسينات الكافية، أصبح من الممكن استخدام هذه الأساليب من قِبَل أي شخص يُمكنه اتباع التعليمات. ثمة نقطة مُهمة ولكنها صعبة، وهي أن خطوات الأسلوب يجب أن تكون خالية من الغموض تماماً. كلما قلَّ الغموض، زادت موثوقية الطريقة بالنسبة إلى غير الخبراء. لذلك كان يتم تقليل الغموض عن طريق استخدام سلاسل دقيقة من العمليات الحسابية والمنطقية بدلاً من الخطوات المُبهمة.

بدءاً من حوالي عام ١٦٥٠، بدأ بعض علماء الرياضيات في البحث عن آلات لتنفيذ العمليات الأساسية في الأساليب الشائعة. كانت بعض الأساليب مُعقدة للغاية بحيث يصعب تذكرها. وبعضها كان لا بد من تكراره عدة مرات، ومن ثم كان من الصعب على البشر الذين يتشتت انتباههم بسهولة إكماله دون أخطاء. ولذا فإن الآلات ستساعد في إجراء الحسابات بسرعة أكبر وبأخطاء أقل.

لإنشاء الآلات، كان على علماء الرياضيات والمُخترعين ابتكار أساليب، مثل وضع العجلات والتروس، لتمثيل الأعداد بمقادير فعلية. كما كان عليهم ابتكار تمثيلات للخطوات المنطقية مثل القفزة الشرطية أو الحلقة. واليوم، تُعد تمثيلات البيانات وخطوات المنطق عناصر أساسية مُهمة في التفكير الحوسبي. وسوف نوضح هذه الجوانب بمزيد من التفصيل فيما تبقى من هذا الفصل.

(١) السعي إلى القضاء على الحدس

تطوّرت الأساليب الحسابية عبر تاريخ الرياضيات بهدف مساعدة البنّائين والمهندسين والتجار والعلماء في حساب الأعداد. اخترع التجار القدماء نُظم الأعداد ونظم المحاسبة وأدوات مثل العدّاد لمتابعة أعمالهم التجارية. واخترع المهندسون القدماء طرقاً لبناء أسلحة لفترات الحرب ومنشآت مدنية لفترات السلام. وسعى الجميع إلى أساليب موثوق بها للتعامل مع العمليات الحسابية التي تتضمن مقادير كبيرة من الأعداد بحيث تؤدي الأعمال المطلوب منها ويمكن الاعتماد عليها.

وقد انتقلت هذه الأساليب من خلال التدريب المهني، ولم ينجح في استخدامها غالباً سوى الخبراء. طوّر الخبراء قواعد قائمة على التجربة العملية وخوارزميات تجريبية وتخمينات من وحي تجربتهم، وغيرها ليتمكنوا من حلّ المشكلات التي يعجز قليلو الخبرة عن حلّها. ويصف المصطلح الحديث «الحدس» عمل الخبير في صياغة حلّ بسرعة، بناءً على خبرة واسعة في مواقف مُماثلة. ويمكن الحدس الخبراء من العثور على جوهر المشكلة الأساسي، وتخطّي الخطوات غير الضرورية لحلّها، والتبديل بين طرق الحل. والحدس هو مظهر من مظاهر الخبرة ويسمح بالوصول إلى نتائج جديدة.

قد يبدو مُتناقضاً إذن أن الكثير من علماء الرياضيات والمنطق على مرّ العصور قد استهدفوا القضاء على الحدس في كلّ من الاستدلال المنطقي وعمليات الحساب الروتينية. كان المطلوب أن تكون مهام الحساب الروتينية بسيطة و«ميكانيكية» قدر الإمكان من

أجل الحصول دائماً على النتائج نفسها بغض النظر عمن قام بالعملية الحسابية. ولذا سعى علماء الرياضيات عبر التاريخ إلى وضع خبرتهم في إجراءات مُتدرّجة الخطوات بحيث يمكن لأي شخص اتباعها بالقليل من التدريب. ولم يكن القضاء على الحدس في المهام الروتينية يعني القضاء على الخبراء، بل كان يعني بالأحرى جعل خبرتهم متاحةً لعددٍ كبير من الأشخاص غير الخبراء.

وقد ورثنا الأفكار الحديثة حول تمثيل المعلومات الرمزية، ومعالجة الرموز، والخطوات الواضحة في الحوسبة، والحساب الأساسي، والخوارزميات، ومزامنة عمليات الحوسبة، والتحقّق المنهجي من الأخطاء من تلك القرون العديدة. من خلال إظهار كيف كانت أتمتة العمليات الحسابية سمةً رئيسية في العديد من تطورات طرق الحوسبة، فإن هدفنا هو الكشف عن كمّ مهارات التفكير الحوسبي التي كانت جزءاً لا يتجزأ من العديد من أنواع التفكير الأخرى التي سبقت ظهور الحوسبة الحديثة. والواقع أن العديد من سمات التفكير الحوسبي وممارساته تدعم تصميم عمليات الحوسبة في العديد من المجالات، وليس فقط في مجال علوم الكمبيوتر.

(٢) التمثيلات العددية والأساليب العددية

يعتمد التفكير الحوسبي، مثل الكثير من العلوم الحديثة، على عملية تمثيل الأشياء في العالم بأعداد أو رموز أخرى. والتمثيل هو ترتيب الرموز بتنسيقٍ مُتفق عليه؛ بحيث يُشير كل تمثيل إلى شيءٍ ما. وكثيراً ما نستخدم الأعداد لتمثيل مقادير كمية مثل بيانات التعداد السكاني، ودفاتر الحسابات التجارية، أو القياسات الهندسية. ولكن الأعداد ليست النوع الوحيد من التمثيل. فالعلماء والمهندسون يُمثلون الظواهر بمعادلات، مثل استخدام المصفوفات الجبرية الخطية لتمثيل دوران الأجسام أو استخدام المعادلات التفاضلية لتمثيل حركة الكواكب. كما يمثلون الأجسام بأدوات ميكانيكية مثل نماذج المباني أو أنفاق الريح أو نماذج محاكاة حركة الكواكب. ويعتمد القياس والتجريب وإمكانية تكرار النتائج — وهي قوام التفكير العلمي — على الأعداد والمعادلات والنماذج. في عصر الحوسبة، أصبحت التمثيلات بالرموز غير العددية شائعة؛ مثل برامج لغة جافا، أو الخرائط النقطية للصور، أو المسارات الصوتية للمقطوعات الموسيقية. واليوم نستخدم مصطلح «الرقمنة» للإشارة إلى عملية تشفير أي معلوماتٍ تقريباً في شكل تمثيلات بيانات يمكن معالجتها بواسطة أجهزة الكمبيوتر.

قد يبدو متناقضاً إذن أن الكثير من علماء الرياضيات والمنطق على مرّ العصور قد استهدفوا القضاء على الحدس في الاستدلال المنطقي وعمليات الحساب الروتينية. ولم يكن القضاء على الحدس في المهام الروتينية يعني القضاء على الخبراء، بل كان يعني بالأحرى جعل خبرتهم متاحةً لعددٍ كبير من الأشخاص غير الخبراء.

لم يعوّل بعض المتشككين على هذه الأساليب الحوسبيّة لأنّ الحسابات العددية كانت معرضة طوال الوقت لحدوث أخطاء صغيرة في دقّة معاملات عملية الحوسبة ومتغيراتها. وأدى هذا بمُصممي الأساليب إلى وضع قيودٍ للحد من تراكم الأخطاء. ولا تزال أجهزة الحوسبة الحديثة اليوم تواجه المشكلات نفسها؛ لأنّ لديها دقة محدودة (مثل ٣٢ أو ٦٤ بت) ويمكن أن تتراكم أخطاء التقريب في الخوارزميات الرديئة التصميم. كان حساب التفاضل والتكامل يُعد بمثابة انطلاقة؛ لأنه أتاح للمُصممين طرقاً منهجية للحدّ من الأخطاء في حسابات الفروق المُنتهية التي يُجرونها.

(٣) تقسيم مهام الحوسبة

خلال الفترة التي سبقت الحرب العالمية الثانية، طوّر الجيش الأمريكي مدفعية أكثر تطوراً قادرة على إطلاق القذائف لمسافة عدة أميال. واحتاج رجال المدفعية إلى معرفة كيفية توجيه مدفيعتهم في ضوء المسافة، وفرق الارتفاع، والرياح. ولذا كلف الجيش فرقاً من الحواسيب البشرية لوضع جداول إطلاق النار لهذه المدافع. وكان على رجال المدفعية ببساطة البحث عن الزاوية والاتجاه الصحيحين لتوجيه مدافعهم، في ضوء قياساتهم للمسافة والارتفاع والرياح.

كان أحد أشهر هذه الفرق يتألف من نساء يعملن في قاعدة «أبردين» في ولاية ميريلاند حوالي عام ١٩٤٠. وقد نظمن أنفسهنّ في خطوط تجميع، حيث تقوم كل واحدةٍ منهنّ بمرحلة مختلفة من الحوسبة، حتى قمن بتجميع جداول إطلاق النار وتحويلها إلى لغة تفهمها الآلة. واستخدمن في ذلك الآلات الحاسبة التي تقوم بالعمليات الحسابية الأساسية (الجمع والطرح والضرب والقسمة). واتّبعن برامج (أي مجموعات من الإجراءات) وضعها المديرون لتقسيم العمل والتحكّم في انتقال الحسابات الوسيطة من حاسبة بشرية إلى أخرى. ولكونهنّ عاملات رياضياتٍ مدربات، تمكنّ من اكتشاف الأخطاء في عمليات الحوسبة الخاصة بهنّ، ومن ثمّ حافظن على خلوّ جداول إطلاق النار من الأخطاء.

يتبع التفكير الحوسبي اليوم نمطاً مُشابهاً يرجع إلى هذا النهج المُتبع قديماً:

- تقسيم عمليات الحوسبة بأكملها إلى مراحل يمكن أن تقوم بها أجهزة كمبيوتر منفصلة ومتصلة ببعضها.
- تنظيم أجهزة الكمبيوتر لتحسين أنماط التواصل والمراسلة بينها؛ على سبيل المثال: في خط تجميع، أو في نسقٍ موازٍ ضخم يعتمد على التوزيع الموسَّع والربط.
- تضمين أدوات التحقق من الأخطاء في أساليبهم بحيث يمكن للمستلمين التحقق من صحة مدخلاتهم ومخرجاتهم.

يعرف مصممو البرامج اليوم هذه المبادئ بالأسماء التالية: الحوسبة الموزعة، والتوازي، والتحقق من الأخطاء. ولكن هذه الممارسات لم تُوضَّع في الأصل للآلات، بل وُضعت للحواسِب البشرية.

أراد الجيش الأمريكي إجراء هذه الحسابات على نطاقٍ أكبر بكثيرٍ وأسرع بكثيرٍ مما تستطيع الفرق البشرية في «أبردين» القيام به، لذلك كُلف مشروع الكمبيوتر والمكامل الرقمي الإلكتروني (إنياك) في جامعة بنسلفانيا بالقيام بذلك. وواجه مصممو الكمبيوتر والمكامل الرقمي الإلكتروني تحديات هائلة، مثل تعلم كيفية بناء دوائر إلكترونية موثوقة لتنفيذ نفس عمليات الحوسبة بشكلٍ أسرع بكثيرٍ، وتعلُّم كيفية تصميم برامج التحكم والخوارزميات لمنع تراكم الأخطاء في عمليات الحوسبة. وتحوَّل أسلوب تقسيم المهام إلى خطوات واضحة تُنقل البيانات بينها، من كونه مبدأً إدارياً في «أبردين» إلى مبدأ تصميمي في أجهزة الكمبيوتر (أو الحواسِب الآلية).

ازداد القلق بشأن الأخطاء مع زيادة حجم الآلات وتعقيدها. وما زلنا حتى الآن نُعلِّم الطلاب في علوم الكمبيوتر هذه الحكمة القديمة: يمكن أن تحدث الأخطاء في أي مرحلة من مراحل عملية الحوسبة، بما في ذلك وصف المشكلة، وإعداد الصِّغ، وتعيين الثوابت، وإرسال البيانات، وتسجيل البيانات واسترجاعها، وتنفيذ الخطوات المُوضَّحة، وعرض النتائج.

(٤) قواعد التفكير المنطقي

إن تصميم عمليات الحوسبة في خطوات واضحة ليس كافياً للوثوق في أن عمليات الحوسبة خالية من الأخطاء. إذ يجب ربط الخطوات معاً وفق خطة مُعينة. وفي أي مرحلة من

عملية الحوسبة يجب أن تخبرنا الخطة بشكل واضح وصريح عن الخطوة التالية. ويُعد تحديد الخطوة التالية ممارسةً من ممارسات المنطق.

كان هناك فرع قديم جداً من الرياضيات والفلسفة مُهتَمٌ بالمنطق. فهل يُمكننا توفير الوسائل لتطوير سلاسل طويلة من التفكير المنطقي لحل المشكلات والتحقق من صحة سلسلة معينة من التفكير المنطقي؟ وعلى غرار نظرائهم في علم الحساب، سعى علماء المنطق إلى طرُق لتوحيد التفكير المنطقي وأتمتته. وسعى فلاسفة مثل رينيه ديكارت وجوتفريد لايبنتس في القرن السابع عشر إلى لغةٍ تصوغ الاستدلال البشري بشكل كامل وتقلل سوء الفهم. وكان هدفهم إرساء طريقةٍ قياسية للتعبير عن المفاهيم وقواعد الاستدلال لتحديد صحة العبارات أو خطئها بشكل قاطع. وَفَقاً لرؤيتهم، فإن «لغة التفكير» هذه من شأنها أن تُنهي الخلافات في جميع المجالات؛ لأننا يمكن أن نحسم جميع النقاشات بالمنطق البحت.

كان التقدم نحو هذا الحلم بطيئاً. ولكن حدثت طفرة في القرن التاسع عشر. رأى جورج بول (١٨١٥-١٨٦٤) كيف أن نظم الرموز الرياضية الجيدة الصياغة كانت قادرة على تقديم نتائج للمشكلات على نحو شبه آلي بمجرد تعيين القيم الصحيحة في الصيغة، وافترض بذلك.¹ وفي كتابه «قوانين التفكير» (١٨٥٤) قدّم «الجبر الفكري» الموازي لجبر الأعداد. وتضمّن منطقته متغيراتٍ يمكن إعطاؤها قيمة true أو false. واستطاع صياغة تعبيراتٍ منطقية، وهي صيغ من المتغيرات المرتبطة عن طريق عوامل تشغيل مثل and و or و not. وبعد ما يقرب من تسعة عقود (١٩٣٧)، أظهر كلود شانون كيف تمكّن جبر جورج بول من وصف دالة دوائر المُرحّلات في نظم التليفون والدوائر الكهربائية الأخرى. وتم تحسين الجبر البوليني لتصميم الدوائر الإلكترونية في خمسينيات القرن العشرين، حيث قدم وسيلةً لإيجاد أصغر دائرة لدالةٍ منطقيةٍ معينة ووسيلة لتصميم الدائرة لتجنّب حالات التعارض، وهي المخرجات الغامضة الناجمة عن تغير الإشارات بسرعات مختلفة في أجزاء مختلفة من الدائرة. وأصبح الجبر البوليني ركيزة أساسية في تصميم دوائر الكمبيوتر. على الرغم من مزايا الجبر البوليني، فإنه كان لديه بعض المساوئ الخطيرة. فالجُمْل التي تشير إلى المجموعات، مثل «كل من لديه شعر رَمادي»، على الرغم من كونها مفهومة تماماً باللغة الطبيعية، لا يمكن التعبير عنها في المنطق البوليني. إذ لم يكن هناك طريقة لإنشاء قائمة بالكيانات التي تنطبق عليها صيغة ما؛ فلم يكن هناك معنى واضح لمفاهيم «كل» و«أحدهم». ولم تكن هناك قواعد للمُحددات الكميّة المهمة «كل» و«بعض».

قدم جوتلوب فريجه (١٨٤٨-١٩٢٥) نظامًا جديدًا للاستدلال المنطقي، «لغة التفكير الخالص» (١٨٧٩)، والذي يُسمَّى حاليًا المنطق الإسنادي. وتضمَّن مُحددات كميَّة جديدة لـ «كل» و«بعض» ليسد ثغرات المنطق البولياني. وقَدَّم نظام فريجه أيضًا قواعد آلية لمعالجة الرموز يمكن اتِّباعها دون اللجوء إلى الحدس البشري. يُشبه المنطق الإسنادي لفريجه لغة البرمجة من حيث إنه يوفر لغةً شكلية اصطناعية تُقدِّم قواعد واضحة وحاسمة وآلية لمعالجة الرموز.

في أوائل القرن العشرين، بدا أن حلم الوصول إلى لغة شكلية للتفكير على وشك التحقق. وأدَّى اندماج الرياضيات والمنطق إلى ظهور العمل الرائد لراسل ووايتهد «مبادئ الرياضيات» (١٩١٠) وظهور التجريبية المنطقية في العلوم. ولكن كان هناك عنصر لا يزال مفقودًا لتحقيق هذا الحلم؛ أسلوب لتحديد ما إذا كانت عبارة في المنطق الإسنادي صحيحة أم غير صحيحة بشكلٍ قاطع. أصبح السؤال حول وجود مثل هذا الأسلوب معروفًا باسم «مشكلة القرار». وبحلول عشرينيات القرن العشرين، اعتُبر هذا أحد التحدّيات الرئيسية في الرياضيات. واعتقد معظم علماء المنطق الرياضي أن هذا الأسلوب موجود، ولكن لم يتمكَّن أحد من العثور عليه.

(٥) ميكنة الحوسبة

في عام ١٩٣٥، تعرَّف طالب رياضيات شاب في كامبريدج على مشكلة القرار للمرة الأولى. وفُتِن الشاب بالكلمات التي استخدمها المحاضر لطحها: هل كانت هناك عملية ميكانيكية للجزم على نحو قاطع، وفي عددٍ محدود من الخطوات، بصحة فرضيةٍ ما في المنطق الإسنادي من خطئها؟ قرَّر هذا الطالب، آلان تورينج (١٩١٢-١٩٥٤)، تطوير نموذج ميكانيكي تمامًا للحوسبة حتى يتمكن من التحقيق في مشكلة القرار.

بدأ تورينج بفكرة أنه عند إحصاء الأعداد، يكتب الحاسب البشري سلسلةً من الرموز على الورق. مثل الورقة بتسلسلٍ خطِّي من المربعات يحتوي كلُّ منها على رمزٍ واحد. وأثناء الإحصاء ينقل الشخص اهتمامه من المربع الحالي إلى أحد أقرب المربعات المجاورة، مع إمكانية تغيير الرمز في المربع. وافترض أن عقل الشخص الذي يقوم بالعملية الحسابية كان في حالةٍ من عددٍ محدود من الحالات، وأن كل واحدة من هذه التحركات الأساسية على الورق نتجت عن الانتقال من الحالة الحالية إلى حالة تالية محددة. وتستمر هذه العملية حتى إتمام العملية الحسابية. أخذ تورينج هذه الإجراءات الأساسية — عند وجودك في

حالة معينة، انتقل يسارًا أو يمينًا مربعًا واحدًا، وقرأ الرمز في المربع الحالي، ثم غير الرمز في المربع الحالي، وانتقل إلى الحالة التالية — واعتبرها التقنية الأساسية لإجراء عملية الحوسبة.

من الواضح أن الآلة يُمكنها القيام بهذه الخطوات وتتبع الحالات. لاحظ أن هذه الآلة حاكت خطوات إحصاء الأعداد أو إيجاد قيمة الدوال المنطقية. بعد توضيح كيفية عمل مثل هذه الآلة، أظهر تورينج أنه توجد آلة واحدة يُمكنها محاكاة جميع الآلات الأخرى، مما يعني أن نموذج الآلة هو طريقة عامة لتمثيل جميع العمليات الحسابية والبراهين. ثم أثبت أنه لا توجد آلة يُمكنها حل مشكلة القرار؛ لأن وجود آلة يُمكنها القيام بذلك يؤدي في حد ذاته إلى مفارقة منطقية. وفي نهاية الأمر، أدى هذا العمل الرائع إلى ذبوع صيته وشهرته بـ «آلة تورينج» وأثارها على الحوسبة.

بعد بضع سنوات، قدم الكمبيوتر الرقمي الإلكتروني الوسيلة اللازمة لأتمتة العملية الحسابية والبرهان؛ محققًا أخيرًا، وإلى حد ما على الأقل، حلم ميكنة العمليات الحسابية والمنطقية. وكانت الأتمتة هي أساس جميع هذه التطورات. للتأكيد على ذلك، أطلق تورينج على آلاته اسم «الأجهزة الآلية». وبالمثل، فإن المهندسين الذين صمّموا أجهزة الكمبيوتر الإلكترونية الأولى في الأربعينيات من القرن العشرين، مثل UNIVAC و BINAC، أطلقوا عليها «الكمبيوتر الآلي». خلال الثمانينيات، كان علم الكمبيوتر نفسه يوصف بأنه علم الأتمتة. يتطلّب الجانب الأساسي من الأتمتة أن تقوم الآلة بالعمل دون تدخل بشري. والكمبيوتر الآلي هو الشكل النهائي لتحقيق الحلم القديم في إتاحة إجراء العمليات الحسابية للجمهور دون الحاجة إلى أن يكونوا خبراء في إجراء العمليات الحسابية.

جانب آخر مهم من الأتمتة هو إدراك أن أجهزة الكمبيوتر الآلية لا يُمكنها أداء بعض المهام الحيوية. بيّن تورينج هذا عندما أثبت أنه لا يمكن أن يُوجد جهاز آلي لحل مشكلة القرار. ويُبين منطق نفسه أنه لا يمكن لأي جهاز آلي حل مشاكل ذات أهمية عملية، مثل تحديد ما إذا كان برنامج كمبيوتر معين سيتوقف أو سيدخل في حلقة تكرارية لا نهائية، أو ما إذا كان برنامج معين يحتوي على فيروس كمبيوتر. ولهذا السبب، يهتم جزء كبير من التفكير الحوسبي بكيفية تقديم حلول جزئية للمشكلات التي لا يمكن حلها بواسطة أجهزة الكمبيوتر الآلية.

لم نكن لنفهم إمكانيات الكمبيوتر الآلي وحدوده دون دمج العمليات الحسابية والمنطقية. ومن غير المستغرب أن يعتبر الكثير من الناس التفكير المنطقي جزءًا لا يتجزأ من التفكير الحوسبي.

(٦) رؤى التفكير الحوسبي تأتي من العديد من المجالات

يجب أن يكون واضحاً من هذا النقاش حول أصول التفكير الحوسبي أن التفكير الحوسبي لا يتعلق بكيفية تفكير علماء الكمبيوتر. فعلم الكمبيوتر الحديث هو آخر ١٪ من الجدول الزمني التاريخي للتفكير الحوسبي. ورث علماء الكمبيوتر التفكير الحوسبي من صف طویل من علماء الرياضيات والفلاسفة الطبيعيين والعلماء والمهندسين المهتمين جميعاً بإجراء عمليات حسابية كبيرة واستنتاجات معقدة بدون أخطاء، وطوّروا منه. ولذا، فالتفكير الحوسبي هو سمة في العديد من المجالات، وليس في مجال الحوسبة فقط.

أراد علماء المنطق إنشاء نظم شكلية يمكن للمرء أن يبدأ فيها من المقدمات، ومن خلال اتباع سلاسل من البدائل داخل نظام شكلي من القواعد، سيصل دائماً إلى نفس الاستنتاجات. كانت رؤى بول وشانون المنطقية — أن عدداً قليلاً من العمليات المنطقية يمكن أن يُعبر عن قيم صدق كل المنطق المقترح، وكذلك التصميم المنطقي للدوائر الرقمية — مدفوعة بسعي قديم لاستبعاد كل العاطفة البشرية والحكم من منطق الاستدلال المنطقي. تُعد هذه الرؤى اليوم من بين المبادئ الأولى للحوسبة. قدمت رؤية فريجه المنطقية — ممثلة في المنطق الإسنادي — نظاماً أكثر فاعلية للاستدلال، له العديد من أوجه التشابه مع لغات البرمجة الحديثة. ورؤية تورينج للسلمات الأساسية للمعالجة الآلية — أن خمسة إجراءات وعدداً محدوداً من الحالات يكفيان لأي عملية حوسبة — جاءت من المنطق الرياضي.

نشأت رؤى أساسية أخرى للتفكير الحوسبي من العلوم والهندسة. من أهمها إدراك أن معظم عمليات الحوسبة في العلوم والتكنولوجيا تتطلب عمليات حسابية طويلة جداً تتجاوز قدرات أي فريق من البشر. إن مُصممي الأساليب الحوسبية لحل المشكلات العملية مهووسون بالتحكم في الأخطاء والحد منها من خلال جعل خطوات الحوسبة بسيطة وواضحة، وجعل منطقها الترابطي لا يمكن دحضه.

يُعد الكمبيوتر الحالي هو الأداة التي سعى إليها الكثيرون عبر العصور لأتمتة العمليات الحسابية وتحريرها من ضعف البشر والحاجة إلى تدخّلهم وأحكامهم. يجسد الباحثون والمحترفون الجدد في مجال الحوسبة هذا التاريخ الطويل ويتفوّقون في أتمتة عمليات الحوسبة باستخدام أفضل الأساليب المتاحة. ومع ذلك، كما سنرى في الفصل التالي، كان من الصعب جداً تحقيق الرغبة في بناء نظم حقيقية لعمليات الحوسبة الضخمة الخالية من الأخطاء.

التفكير الحوسبي

يعد الكمبيوتر الحالي هو الأداة التي سعى إليها الكثيرون عبر العصور لأتمتة العمليات الحسابية وتحريرها من ضعف البشر والحاجة إلى تدخلهم وأحكامهم.

الفصل الثالث

آلات الحوسبة

كان العمل الشاق والرتابة المُضنية من التَّكرار المستمر لعملياتٍ حسابية متماثلة، هو ما أثار الرغبة في البداية في وجود آلة لتكون بمثابة بديلٍ لإحدى أبسط العمليات الذهنية التي يُجريها البشر، ومن ثَمَّ طُرحت الفكرة لابتكار مثل هذه الآلة، بمساعدة الجاذبية أو أي قوة مُحركة أخرى.

تشارلز بابيدج (٣ يوليو ١٨٢٢، رسالة إلى همفري ديفي)

الآن سنُلقي نظرة على تطور آلات الحوسبة وبعُد التفكير الحوسبي اللازم لتصميمها وفهمها. كان الدافع الفعلي الأساسي لبناء آلات الحوسبة هو دائماً تسريع عمليات الحساب والتخلُّص من الأخطاء المتأصِّلة في الحواسِب البشرية.

لطالما كان الناس مفتونين بفكرة إنشاء أجهزة لأتمتة جوانب من السلوك البشري أو التفكير البشري. على مدى آلاف السنين، أنشأ الحرفيون آلاتٍ ذاتية التشغيل لأغراضٍ فنية وترفيهية، مثل الحيوانات ذاتية الحركة وصناديق الموسيقى والبيانوهات الذاتية التشغيل وتمائيل البشر التي تُحاكي سلوك الناس. وقد أثار «التركي الآلي»، الذي يُطلق عليه أحياناً لاعب الشطرنج الآلي، الجدل في عام ١٧٧٠ إذ بدا أنه يحوِّل لعب الشطرنج إلى عملية آلية مُمكنة، في حين أنه كان يُعتبر آنذاك مهارة عقلية رفيعة المستوى. ولكن اتضح لاحقاً أنه خدعة مُعقدة. لكنه أثار فضول المُخترعين الذين تساءلوا عما إذا كان بإمكانهم حقاً بناء آلة تلعب الشطرنج. واعتقد بعض الفلاسفة أن الآلات الذاتية التشغيل المُستخدمة في إجراء العمليات الحسابية، وهي أيضاً مهارة عقلية عُليا لدى البشر، قد تكون أكثر جدوى؛ لأن قواعد الحساب الأساسية كانت أوضح وأبسط بكثيرٍ من قواعد الشطرنج واستراتيجياته.

(١) بزوغ نجم آلات الحوسبة

عندما يتمكن الخبراء من تشفير ما يعرفونه عن الحساب والمنطق في شكل خطوات إجرائية، تُصبح معرفتهم مفيدة للعديد من غير الخبراء، الذين يمكنهم الحصول على النتائج دون خطأ عن طريق اتباع التوجيهات ببساطة. ولكن بغض النظر عن مدى دقة الإجراء، فإن المُشغّلين البشر عرضة للخطأ. فهم ينسون، ولا يفهمون كل العمليات الحسابية فهماً تاماً، ويسهل تشتيتهم، ويشعرون بالملل بسرعة عند إجراء عملية حسابية روتينية طويلة. وبغض النظر عن مدى بساطة الخطوات ووضوحها، يرتكب الحواسِب من البشر أخطاء. بل يرتكبون الكثير من الأخطاء. فقد وجدت إحدى الدراسات التي أجريت على ٤٠ مجلداً من الجداول الرياضية القديمة ٣٧٠٠ خطأ، ووجدت دراسة أخرى ٤٠ خطأ في صفحة واحدة.

لهذا السبب، سعى المخترعون عبر العصور إلى آلات حوسبة وأدوات حسابية مساعدة تتيح للبشر إجراء عمليات حسابية أطول بأخطاء أقل. وقد حدثت هذه العملية ببطء. اخترعت المسطرة المنزلقة حوالي عام ١٦٢٠. وعن طريق تحريك العصي المميزة بمقاييس لوغاريتمية بعضها فوق بعض، كانت تنفذ أسلوب الضرب القائم على جمع اللوغاريتمات. ولكن المسطرة المنزلقة لم تكن تستطيع الجمع أو الطرح. وفي عام ١٦٤٢ صمم بليز باسكال آلة حاسبة يُمكنها الجمع والطرح، ولكن لا يمكنها الضرب أو القسمة. وفشلت محاولات الآخرين لتوسيع إمكانيات الآلة الحاسبة التي صمّمها باسكال بحيث تسمح بعملية الضرب.

وجدت المسطرة المنزلقة مكانها بين المهندسين، بينما استخدم الآلة الحاسبة الرياضية علماء الرياضيات والمحاسبون. وعلى مدار القرون التالية أُدخلت تحسينات تدريجياً على هذه الآلات. فبحلول الثلاثينيات من القرن العشرين، كانت شركة «كوفل أند إسر» هي المورد الرئيسي للمساطر المنزلقة اللوغاريتمية المثلثية، وكانت شركة «مرشانت» هي المورد الرئيسي للآلات الحاسبة الميكانيكية التي تقوم بجميع العمليات الحسابية الأربع. اجتاح طوفان التغيير الذي أطلقته ثورة الكمبيوتر الإلكتروني في خمسينيات القرن العشرين العديد من شركات المساطر المنزلقة والآلات الحاسبة الميكانيكية. وبدأت شركات جديدة مثل «هوليت - باكارد» و«تكساس إنسترومنتس» في إنتاج آلات حاسبة مكتبية إلكترونية بالكامل يمكنها أداء جميع وظائف المسطرة المنزلقة والعمليات الحسابية. وجاءت الضربة

القاضية في عام ١٩٧٢ عندما ظهرت الآلة الحاسبة المحمولة القابلة للبرمجة «إتش بي-٣٥»، والتي حلّت محلّ المسطرة المنزقة لدى المهندسين.

رغم شعبيتهما، كان للمسطرة المنزقة والآلة الحاسبة عيبان خطيران. أولهما: أنهما لم يتمكّنا من إجراء سلاسل طويلة من العمليات الحسابية؛ ولذا كان على المشغلين البشريين القيام بذلك. وثانيهما: أنه يمكن استخدام هاتين الأدوات لغرض واحد فقط. تغلب الكمبيوتر الرقمي الإلكتروني على هذين العيبين بفكرة جذرية: البرنامج المخزّن داخلياً في ذاكرة الآلة. يمكن للبرنامج إجراء عمليات حسابية طويلة ويمكن تعديله بسهولة لتغيير استعمال الآلة الأساسية.

تعود أصول فكرة البرامج إلى ما قبل عصر الحوسبة الإلكترونية. ففي أوائل القرن الثامن عشر، جرّب النساجون الفرنسيون آلاتٍ يمكنها نسج أنماط مُعقدة باستخدام النول الآلي. كانت إحدى أكثر هذه الآلات شهرة هي نول جاكارد، والذي كان التحكم فيه من خلال سلاسل طويلة من البطاقات المثقبة؛ إذ تدخل إبرة خطافية في الثقب الموجود في البطاقة، فيُرفَع خيط يُصبح جزءاً من خطٍّ واحد من النسيج. أحدث نول جاكارد الآلي ثورة في صناعة النسيج. وكانت بطاقات جاكارد شكلاً من أشكال البرامج الخارجية القابلة للتغيير التي تتحكّم في تشغيل النول.

أعجبت فكرة التحكم في الآلات بالبطاقات المثقبة هيرمان هوليريث، الذي صمّم آلةً لجدولة البيانات المُستخلصة من تعداد سكان الولايات المتحدة لعام ١٨٩٠. وسجّل بيانات كل مواطن على هيئة نمط من الثقوب المثقبة في بطاقة، حيث يُمثّل هذا النمط خصائص مثل الجنس والعنوان والأصل العرقي. وكانت آلة الجدولة تختار البطاقات التي تنطبق عليها خصائص مُعينة وتُحصّر الإحصاءات للمجموعة المختارة من المواطنين. وبالإستعانة بآلة هوليريث، أكمل مكتب الإحصاء تحليله لـ ٦٣ مليون سجلٍّ في عامٍ واحد، وكان هذا أسرع وأرخص من أي تعدادٍ سكاني تم في السنوات السابقة. وفي السنوات التالية، استُخدِمت التقنية نفسها لمهام معالجة العديد من البيانات؛ مثل تتبّع الحالة الصحية لعشرات الآلاف من الجنود، والإحصاءات الزراعية، وقوائم الشحن بالسكك الحديدية، وما إلى ذلك.

قبل التطرُّق إلى ما بعد آلات الجدولة، نوّد أن نعود ٥٠ عاماً إلى الوراء لنُلقي نظرةً على الإنجاز المُهم الذي حققه تشارلز بابيدج وآدا لافليس: الكمبيوتر المتعدّد الأغراض.

(٢) آلات بابيدج

صمم تشارلز بابيدج آلتين حاسبتين مهمتين خلال حياته المهنية الطويلة. أولاهما: «محرك الفرق» (حوالي عام ١٨٢٠) ووظيفته أتمتة حساب الجداول الرياضية مثل جداول اللوغاريتمات أو جيب الزاوية. والثانية: «المحرك التحليلي» (حوالي عام ١٨٤٠)، وهو عبارة عن كمبيوتر متعدد الأغراض يُمكنه تنفيذ أي دالة حسابية.

في زمن بابيدج أعد الخبراء كتبًا تحتوي على جداول للدوال المهمة مثل لوغاريتمات جميع الأرقام المكوّنة من ستة أرقام. وشاع استخدام هذه الجداول للعمليات الحسابية الرياضية؛ على سبيل المثال: يمكن للمرء ضرب عددين عن طريق البحث وجمع لوغاريتماتهما. تم حساب هذه الجداول يدويًا باستخدام صيغ الفرق التي تحسب كل سطرٍ من الجدول من السطر السابق له. وعرف بابيدج أن هذه الكتب المحسوبة يدويًا تحتوي على العديد من الأخطاء، وأن هذه الأخطاء أدت أحيانًا إلى عواقب وخيمة. على سبيل المثال: أوضح بابيدج أن الأخطاء في جداول الملاحة التي تستخدمها البحرية البريطانية تسببت في حوادث غرق السفن. وأراد القضاء على الأخطاء بإحلال آلات محل البشر الذين يتعبون ويشعرون بالملل ويتشتت انتباههم. فابتكر آلة أطلق عليها اسم «محرك الفرق» لحساب جداول الأعداد وطباعتها. وأُعجبت الحكومة البريطانية بهذه الآلة ومنحته المال لتطويرها.

قضى بابيدج معظم العشرين عامًا التالية في محاولة بناء آله. وكان هذا تحدّيًا أكبر بكثير مما كان يتصور؛ إذ لم تكن أساليب الهندسة الميكانيكية في ذلك الوقت قادرة على إنتاج آلاف التروس والرافعات بالدقة المطلوبة لتجنب التخطّي أو التعطّل. وفي ثلاثينيات القرن التاسع عشر ابتكر تصميمًا جديدًا يُسمّى «المحرك التحليلي»، والذي سيحتاج إلى قطع غيار أقلّ ويكون أكثر قوة؛ إذ سيكون قادرًا على حساب أي دالة رياضية. ولكن بحلول ذلك الوقت، كانت الحكومة قد فقدت الثقة فيه بسبب فشله في تسليم «محرك الفرق» ورفضت دعم مشروع «المحرك التحليلي». وتابّع ذلك المشروع بتمويل محدود حتى وفاته عام ١٨٧١، ولكنه لم يكمله أبدًا. وظلّت أفكاره الرائدة في حالة سكونٍ على مدى الـ ٨٠ عامًا التالية.

كان «المحرك التحليلي» يأخذ تعليماته من البطاقات المثقبة، وهي فكرة مُستوحاة من نول جاكارد. واحتوت البطاقات المثقبة على برنامج يوجّه الآلة لحساب دالة رياضية تلقائيًا. وكانت الآلة قادرة على تحديد ما يجب فعله بناءً على النتائج السابقة (الاختيار)

وتكرار أجزاء من البرنامج (التكرار). وكانت تحتوي على وحدات مُنفصلة لدوالٍ منفصلة تخصُّ الآلة، وهي: الإدخال والمعالجة والذاكرة والإخراج. وتتشكل تعليمات الآلة من البرامج المصغرة.

تعاون بابيدج مع عالمة الرياضيات الإنجليزية الموهوبة آدا لافليس، التي صممت خوارزميات للمحرك التحليلي. كان أحد برامجها يحسب سلسلةً من الأعداد الكسرية تُسمَّى أعداد برنولي. وغالبًا ما يُنظر إلى بابيدج ولافليس على أنهما أول المبرمجين. والأكثر من ذلك، رأت لافليس آلة بابيدج أكثر من مجرد آلة لحساب الأعداد؛ إذ كانت بالنسبة إليها معالجًا لأي معلوماتٍ يمكن تشفيرها في صورة رموز. وأطلقت على دراسة هذه البرامج اسم «علم العمليات». رأت أن آلات الحوسبة يمكن أن تحسب ليس فقط الأرقام، ولكن أيضًا الرموز التي يمكن أن تُمثل أي شيءٍ في العالم، متوقعةً بسابقٍ مائة عامٍ مبدأً أساسيًا من مبادئ عصر الكمبيوتر الحديث. رأت لافليس الكمبيوتر كآلة معلومات.

كانت رؤية كلٍّ من بابيدج ولافليس مبتكرة. قدمت تصميماتهما العديد من الأفكار التي تُعد اليوم ميزات تُميز التفكير الحوسبي عن أنواع التفكير الأخرى. بالإضافة إلى تمثيل البرامج في وسيط خارجي قابل للتغيير، جسّد المحرك التحليلي العديد من جوانب أجهزة الكمبيوتر الحديثة: التمثيل الرقمي للبيانات، والبرمجة، والخوارزميات القابلة للتنفيذ الآلي، وهياكل التحكم لاختيار الحالات ونظم التكرار، ووحدة الحساب والمنطق، والبرمجة الدقيقة لتحليل تعليمات الآلة إلى عمليات بوابة منطقية منخفضة المستوى. ومن المفارقات أن بعض الأفكار الجوهرية لعصر الكمبيوتر قد وُلدت في عصر البخار.

(٣) الكمبيوتر ذو البرنامج المخزن

لم يكن من الممكن تنفيذ التصميمات المنطقية لجهاز الكمبيوتر الذي ابتكره بابيدج بالاستعانة بتكنولوجيا ذلك العصر، ولكن بعد عدة عقود، فتح عصر الإلكترونيات الناشئ إمكانيات جديدة. كانت الفترة من أواخر الثلاثينيات من القرن العشرين فترة تجريب مكثف لبناء الآلات الحاسوبية. فقد بنى كونراد تسوزه جهاز كمبيوتر في ألمانيا عام ١٩٣٨، لكن الحكومة الألمانية لم تأخذه على محمل الجد ولم يكن له تأثير يُذكر. وتعاون هوارد آيكن مع شركة «آي بي إم» برعاية البحرية الأمريكية، وبنى «مارك الأول» في هارفارد عام ١٩٤٤. وكان كمبيوتر إلكتروميكانيكيًا يتجاوز العالم الميكانيكي الذي تحكمه قوانين

نيوتن للحركة والعالم الفائق السرعة الذي تحكمه قوانين ماكسويل للكهرومغناطيسية. وكانت برامجه وبيانات الإدخال الخاصة به تخرَّن خارجياً على أشرطة ورقية مثقبة.

في كلية مور للهندسة الكهربائية بجامعة بنسلفانيا، صمَّم جون موتشلي وبريسبر إكرت — بدعم من الجيش الأمريكي — جهاز كمبيوتر ربما يُعد أشهر أجهزة الكمبيوتر الإلكترونية الأولى. بدأ تشغيل المكامل الرقمي الإلكتروني (إنياك) الخاص بهما في عام ١٩٤٥ واستُخدم لحساب جداول إطلاق المدفعية واستكشاف جدوى السلاح النووي الحراري. وحصل المكامل الرقمي الإلكتروني على برنامجه من لوحة توصيلٍ بأسلاكٍ خارجية، كانت برمجتها مُملة. وكان المكامل خير دليل على جدوى مفهوم الحساب الإلكتروني بالكامل؛ فقد نجح وكان سريعاً ومُهدِّ لظهور آلاتٍ أفضل بعد ذلك بفترة قصيرة. أسس مهندسوه شركة «يونيفاك»، وهي أول شركة تجارية تُقدم جهاز كمبيوتر إلكترونياً.

في عام ١٩٤٥ اجتمع فريق إنياك، وانضم إليهم جون فون نيومان، لتصميم آلة أفضل بالاستعانة بخبرتهم. بصرف النظر عن صعوبة برمجة إنياك، كانت ذاكرته محدودة، واستخدم الآلاف من أنابيب التفريغ (١٨٠٠٠ أنبوب) التي تآكلت تدريجياً. قسَّم الفريق الآلة — في تصميمها الجديد — إلى ثلاثة نُظُم فرعية رئيسية: وحدة المعالجة المركزية لأداء العمليات الحسابية والمنطقية، والذاكرة للتخزين، ووحدة الإدخال والإخراج للتواصل مع العالم الخارجي. ولزيادة سرعة الكمبيوتر، صمَّموا وحدة معالجة مركزية تأخذ تعليماتها من الذاكرة، وليس من البطاقات أو الأشرطة المثقبة الخارجية، ومن ثَم بدأت فكرة «الكمبيوتر ذو البرنامج المخزن». والغريب أن هذه الطريقة في تنظيم الآلة أصبحت تُعرَف الآن باسم «بنية فون نيومان»؛ حيث دوَّن فون نيومان الملاحظات في اجتماعاتهم ووَزَّعها. وزعم أنه مُدوَّن الملاحظات وليس المُصمَّم. ظهرت بنية فون نيومان في جميع أجهزة الكمبيوتر التجارية تقريباً من ذلك الوقت وحتى الوقت الحاضر. وأصبحت فكرة أن وحدة المعالجة المركزية تتتبع تسلسل تعليماتٍ بين التعليمات المُخزنة في الذاكرة مبدأً رئيسياً من مبادئ التفكير الحوسبي.

(٤) التفكير الحوسبي والآلات

دعنا نتناول الآن المبادئ المُختلفة للتفكير الحوسبي التي قدَّمتها لنا هذه الآلات المُبكرة ونُظَّم تشغيلها.

التمثيلات الرقمية باستخدام الإشارات والرموز الثنائية

لكي تُصبح البيانات قابلةً للمعالجة، يجب تمثيلها كإشارات في الآلة أو كاهتزازات قابلة للقياس في بنية وسائط التخزين. لا تُوجد معلومات دون تمثيل. إذ يجب تمثيل العمليات الحسابية مثل الجمع والطرح كقواعد لنقل الإشارات. وكانت إحدى الطرق المبكرة لتمثيل رقم عشري عبارة عن حلقة من ١٠ أنابيب مفرغة ثنائية القطب تُحاكي عجلة ذات ١٠ مواضع. كان هذا المخطط أكثر تكلفةً بكثير من التمثيل الثنائي المكوّن من ٤ أنابيب لنفس الرقم. رُفِضَت المقترحات لتمثيل الأرقام العشرية بـ ١٠ فولتات مُنفصلة بسبب تعقيد الدوائر. وسرعان ما استقرّ المهندسون على استخدام الرموز الثنائية لتمثيل الأعداد؛ لأن العمليات الحسابية ذات الرموز الثنائية تستخدم مكونات أقل بكثير من العمليات الحسابية ذات الرموز العشرية، ولأن الدوائر التي تُميز بين قيمتين فولتيتين كانت أكثر موثوقيةً بكثير من الدوائر التي تُميز بين أكثر من قيمتين. علاوة على ذلك، يمكن بسهولة إنشاء وحدة التخزين من التكنولوجيا المتاحة ذات الحالة الثنائية مثل خطوط التأخير الصوتي أو النوى المغناطيسية أو الدوائر الإلكترونية القلّابة ذات الوضعيتين أو بُقَع الفوسفور على شاشة أنبوب أشعة الكاثود. وقد أدّى القرار بالتخلي عن العمليات الحسابية العشرية واستخدام التشفير الثنائي في كلِّ شيءٍ في الكمبيوتر إلى دوائر ووسائط تخزين بسيطة للغاية وأكثر موثوقيةً بكثير. وأصبح مُصطلح «بت» مُستخدمًا بشكلٍ قياسي كاختصارٍ لمصطلح «رقم ثنائي». واليوم لا يمكن لأحدٍ التفكير في أجهزة الكمبيوتر المعاصرة دون التفكير في التمثيلات الثنائية.

من المهم أن نضع في اعتبارنا أن الكمبيوتر لا يُعالج الأعداد والرموز داخليًا. ودوائر الكمبيوتر لا تتعامل إلا مع الفولتية والتيارات والمفاتيح والمواد القابلة للتشكيل. وأنماط الأصفار والآحاد هي تجريدات اخترعها المصمّمون لوصف ما تفعله دوائر الكمبيوتر. ونظرًا إلى أن الرموز الثنائية ليست كلها وصفًا صحيحًا للدائرة أو الرمز أو العدد، فقد اخترع المصمّمون قواعد تركيب تُفرّق بين الرموز الصالحة وغير الصالحة. وعلى الرغم من أن الآلة لا تستطيع فهم معنى الأنماط، فإنها يمكن أن تُميز بين الأنماط المسموح بها والأنماط الأخرى عن طريق تطبيق قواعد التركيب اللغوي.

لن نكون مُبالغين إذا شدّدنا على أهمية المكونات المادية في أجهزة الكمبيوتر — مثل الإشارات في الدوائر الإلكترونية أو الرُّقَع المُغْنَطَة على أقراص التخزين — لأنه من دون هذه المكونات المادية لا يُمكننا بناء كمبيوتر. وعلى الرغم من أن برامج الكمبيوتر

تبدو وكأنها أفكار مجردة، فإنها لا يمكن أن تعمل من دون الآلات التي تُسخر الظواهر الفيزيائية لتمثيل الأعداد الثنائية. لهذا السبب، من الآمن القول إن كل مجموعة بيانات وكل برنامج وكل تصميم لدائرة منطقية هو «ترتيب استراتيجي لمكوناتها».

الجبر البوليني وتصميم الدوائر

بسبب رؤية كلود شانون بأن منطق جورج بول يصف بدقة دوائر التبديل الإلكترونية، لا يمكننا اليوم التفكير في أجهزة الكمبيوتر دون التفكير في الجبر البوليني. يساعدنا الجبر البوليني في فهم كيفية تنفيذ المكونات المادية لتعليمات الآلة التي يُصدرها برنامج التحويل البرمجي. لكن الجبر البوليني هو شكل من أشكال التجريد. وفي بعض الأحيان يُخفي حالات التعارض المادية الناجمة عن الإشارات المتعددة التي تتبع مسارات مختلفة إلى نفس المخرج؛ ويمكن أن تتسبب حالات التعارض في أخطاء عن طريق جعل الدوائر تنحرف عن صيغها البولينية. وهذا يُربك المبرمجين الذين هم على دراية فقط بالتجريدات وليس الدوائر الإلكترونية، ولهذا السبب لا يمكن العثور على الأخطاء عن طريق دراسة برامجهم.

دورة وحدة المعالجة المركزية المقيسة الوقت لخطوات الحوسبة الأساسية

تتكوّن البنية المادية لأجهزة الكمبيوتر من السجّلات التي تُخزن أنماط البت، ودوائر المنطق التي تحسب دوال البيانات في السجّلات. يستغرق الأمر وقتاً من دوائر المنطق لنقل الإشارات من سجّلات الإدخال الخاصة بها إلى سجّلات الإخراج الخاصة بها. وفي حال توفير مُدخلات جديدة قبل استقرار الدوائر، من المحتمل أن تُفسّر المخرجات بشكل خاطئ من قبل الدوائر اللاحقة. وقد حلّ المهندسون هذه المشكلة بإضافة ساعات إلى أجهزة الكمبيوتر. ومع كل دقة ساعة تُخزن مخرجات دائرة منطقية في سجّلاتها. وتكون الفترة الفاصلة بين الدقات طويلة بما يكفي لضمان استقرار الدائرة تماماً قبل تخزين مخرجاتها. ولا يمكن لأجهزة الكمبيوتر ذات بنية فون نيومان أن تعمل دون ساعة. وتُصنّف أجهزة الكمبيوتر اليوم وفقاً لسرعات الساعات الخاصة بها؛ على سبيل المثال: فإن «معالج ٣,٨ جيجاهرتز» هو المعالج الذي يصل عدد دقات ساعته إلى ٣,٨ مليار مرة في الثانية.

يمنح وجود الساعات ترجمة مادية دقيقة لـ «الخطوات الخوارزمية» في المجال الرقمي. ويجب إكمال كل خطوة خوارزمية قبل الإقدام على الخطوة التالية. ويدعم الجهاز هذا عن طريق ضمان الانتهاء من كل تعليمة بشكل صحيح قبل إجراء التعليمة التالية وهكذا. (هناك أنواع قليلة من أجهزة الكمبيوتر التي لا تستخدم الساعات، ولكن لن نناقشها هنا.) وتُعد الساعات ضرورية لدعم مفهومنا لخطوات الحوسبة وضمان أداء الكمبيوتر لها بشكل موثوق.

تدفق التحكم

منذ زمن بابيدج ولافليس، أدرك المبرمجون أن الآلة يجب أن تكون قادرة على تحديد التعليمات التالية. فالتعليمات لا تتبع دائماً تسلسلاً خطياً. في بنية فون نيومان، يُخزّن عنوان التعليمة التالية في سجلّ وحدة المعالجة المركزية الذي يُسمّى عداد البرنامج والذي يتم تحديثه بعد كل تعليمة. ويكون الوضع الافتراضي هو تنفيذ التعليمة التالية في التسلسل (يتم ضبط عداد البرنامج على العداد +1). ومن أحد الانحرافات الشائعة عن التسلسل الخطّي الانتقال إلى تعليمة أخرى في موقع ذاكرة مختلف، على سبيل المثال X . يتم التحكم في قرار الانتقال بواسطة الشرط C (مثل «هل A يساوي B ؟») ويتم تنفيذ الانتقال من أحد أجزاء البرنامج إلى جزء آخر عن طريق تعليمة تقول «إذا تحقق C ، فاضبط عداد البرنامج على X ». هذا الأسلوب في التحكم في عداد البرنامج بحيث ينتقل تنفيذ البرنامج إلى جزء آخر من التعليمات البرمجية يظهر في التفكير الحوسبي في صورة تعليمة if-then-else في لغات البرمجة.

الجُمْل التكرارية: برامج صغيرة لإجراء عمليات حوسبة كبيرة

لو اقتصر شكل برامجنا على أشجار اتخاذ القرار التي تتكوّن من تسلسلات من التعليمات، تُحدّد كلّاً منها جملة if-then-else، فلن يُمكنها أبداً إنشاء عمليات حوسبة أطول من عدد التعليمات الموجودة بالبرنامج. تسمح الجملة التكرارية بتصميم عمليات حوسبة أطول بكثير من حجم البرنامج نفسه. والجملة التكرارية عبارة عن تسلسل من التعليمات تتكرّر مراراً حتى يتحقق شرط التوقف. يُمكن تنفيذ الجملة التكرارية باستخدام جملة if-then-else التي تعود إلى بداية الجملة التكرارية عندما لا يتحقّق شرط التوقف. ومن

أخطاء البرمجة الشائعة إعداد شرط توقّف غير صحيح لا يُنهي الجملة التكرارية. ويُسمى هذا السلوك «الجملة التكرارية اللانهائية».

أثبت آلان تورينج أنه لا تُوجد خوارزمية لفحص برنامج بهدف تحديد ما إذا كانت أيُّ من جُمَله التكرارية لا نهائية. وهذا يجعل عملية اكتشاف الأخطاء وإصلاحها مشكلة صعبة لا يمكن أتمتتها. ويقضي المبرمجون قدرًا كبيرًا من الوقت في البحث عن الأخطاء في برامجهم.

تُصمّم بعض البرامج لتستمر في التكرار إلى الأبد. وهذا شائع جدًا في عمليات تقديم الخدمات على الإنترنت. تظل عملية تقديم الخدمة في موضعها الأصلي في انتظار أي طلب وارد، ثم تنفّذ التعليمات البرمجية لتلبية الطلب وتعود إلى موضعها الأصلي مرة أخرى. ومع أن هذا يُسهّل تصميم عمليات تقديم الخدمات، فإنه لا يُزيل التحدي المتمثل في إثبات أن عملية تقديم الخدمة تعود دائمًا إلى موضعها الأصلي.

تمييز العنوان والمحتوى

أدخلت الآلات الحاسوبية ذات البرامج المخزنة تمييزًا بين العنوان (الاسم) والقيمة (المرتبطة بالاسم). في البرنامج، يمثل المتغير X موقع ذاكرة يحمل قيمة ما. في الجبر الكلاسيكي X هي قيمة غير معروفة. في البرنامج، تعني العبارة $X = 3$ «تخزين القيمة ٣ في موقع الذاكرة المُسمّى X ». قارن هذا بمعنى $X = 3$ في الجبر الكلاسيكي، وهي تعني «المجهول X له قيمة ٣». في البرنامج، $X = 3$ أمر؛ أما في الجبر، فهي حقيقة. هذا التمييز هو جزء من تفكيرنا الحوسبي. وغالبًا ما يضع المبرمجون المُبتدئون الذين لا يُميزون بينهما برامج لا تعمل.

البرامج الفرعية

بحلول أواخر أربعينيات القرن العشرين، أدرك مُصمّمو أجهزة الكمبيوتر أن الممارسة الشائعة للمُبرمجين ستكون كتابة تعليمات برمجية للدوال القياسية التي يمكن استدعاؤها من أي موضع في برامجهم. على سبيل المثال: يمكن للمبرمج الخبير كتابة تعليمية برمجية لدالة الفرز SORT، يمكن لأي شخص آخر استخدامها لترتيب قائمة من الأعداد ترتيبًا تصاعديًا. لتمكين الاستدعاء الفعّال لِثل هذه البرامج الفرعية، ضُمّن المُصمّمون نوعًا

جديدًا من تعليمات التفرع في أجهزتهم. ستتذكّر التعليمات CALL X القيمة الحالية لعداد البرنامج ثم تضبط PC على X، ومن ثمّ تنقل التحكم إلى البرنامج الفرعي المخزّن في الذاكرة في الموقع X. عند إتمام التعليمات، سيُنَفَّذ البرنامج الفرعي التعليمات RETURN التي تستعيد قيمة عداد البرنامج المحفوظة، مما يسمح للبرنامج الأصلي باستئناف التشغيل من موضع الاستدعاء.

أصبحت فكرة البرامج الفرعية مبدأً أساسيًا في التفكير الحوسبي. وقد قدّم لنا مُصممو الأجهزة تطبيقاتٍ فعالة لها. تظهر البرامج الفرعية في لغات البرمجة على شكل «روتينات فرعية»، و«دوال»، و«إجراءات». ومن المُسلّم به اليوم أن تكون البرامج مقسّمة إلى وحدات تُنفَّذ كبرامج فرعية.

الآلات ذات الأغراض العامة

في عام ١٩٣٦، قدّم آلان تورينج فكرة الآلة ذات الأغراض العامة؛ وهي عبارة عن كمبيوتر يُمكنه محاكاة أي كمبيوتر آخر، شريطة أن يكون لديه البرنامج الخاص بالكمبيوتر الآخر. ولم تكن الآلة ذات الأغراض العامة مُعقدة في حدّ ذاتها. كانت هذه الفكرة مُضمّنة في تصاميم الآلات التي تعود إلى المحرك التحليلي لبابيدج، حيث يصنع المُصممون آلة واحدة أساسية يُمكنها تشغيل العديد من البرامج. وهذه الآلة الأساسية مثالٌ على الآلة ذات الأغراض العامة. واليوم يُعدّ هذا من المُسلّمات؛ إذ يفترض مصممو البرامج أن برامج التحويل البرمجي ونظم التشغيل ستجعل برامجهم تعمل على أي آلةٍ أساسية عامة الأغراض.

يخلط الناس أحيانًا بين فكرة الآلة ذات الأغراض العامة والكمبيوتر ذي البرنامج المخزّن. وفي الحقيقة هما ليسا مُتماثلين. كان المحرك التحليلي لبابيدج آلة ذات أغراض عامة، وكانت برامجها عبارة عن مجموعاتٍ خارجية من البطاقات المثقبة. وكان المكامل الرقمي الإلكتروني آلة ذات أغراض عامة، وكانت برامجها عبارة عن لوحات توصيل خارجية. بعد عام ١٩٤٥، أصبحت أجهزة الكمبيوتر آلاتٍ ذات أغراض عامة تُخزن برامجها في الذاكرة الداخلية.

يُتيح الكمبيوتر ذو البرنامج المخزّن إمكانيةً تبديل تفسير مجموعة من وحدات البت في الذاكرة ما بين بيانات وتعليمات. فمن المُمكن أن تكون الأنماط نفسها في ذاكرة الكمبيوتر وحدات بت تُمثّل أشياء (بيانات)، وكذلك وحدات بت تقوم بأشياء (تعليمات).

ينشئ برنامج التحويل البرمجي، على سبيل المثال، التعليمات البرمجية للآلة على هيئة بيانات مخرَجة؛ ويمكن لوحدة المعالجة المركزية تفسير هذه البيانات على الفور كتعليمات قابلة للتنفيذ. وقد سمحت بعض الآلات المبكرة للبرامج بتعديل تعليماتها البرمجية لتحقيق مزيد من الفاعلية. ولكن معظم نظم التشغيل قد حالت دون ذلك من خلال تعيين التعليمات البرمجية للقراءة فقط؛ وهذا يسمح بمشاركة التعليمات البرمجية ولكن لا يسمح بتغييرها. إن الفكرة القديمة للبرامج الذاتية التعديل ما زالت موجودة حتى الآن؛ حيث تعدّل البرامج الضارة اليوم تعليماتها البرمجية باستمرار لتجنّب اكتشافها بواسطة برامج مكافحة الفيروسات.

تحمل الأخطاء وحماية البيانات

كثيراً ما تتعرض الدوائر المنطقية للأخطاء الناجمة عن أسبابٍ مادية مختلفة. على سبيل المثال: قد لا يمكن التنبؤ بحالة المُكوّن إذا وصلت إشارات مُتعارضة في الوقت نفسه، أو إذا كانت الساعة سريعة جداً بحيث لا تسمح لبعض المُكوّنات بالاستقرار في حالاتٍ جديدة، أو إذا كانت المكونات تتدهور وتتعطّل بمرور الوقت. ويقضي مهندسو الدوائر الكثير من الوقت في إعداد تحمل الأخطاء. وقد أحسنوا عملهم بشكلٍ عام؛ لأن الأجهزة موثوقٌ بها بما فيه الكفاية بحيث لا يقلق المُستخدمون من حدوث أخطاء في الأجهزة.

في الخمسينيات من القرن العشرين، بدأ مهندسو التصميم في التفكير في أجهزة الكمبيوتر المتعددة الوصول التي سيتم مشاركتها داخل مجتمع المُستخدمين. وبالمثل، توسع التفكير الحوسبي من عمليات الحوسبة للمُستخدم الواحد إلى عمليات الحوسبة المتعددة المُستخدمين. وكان على النظم المتعددة المُستخدمين ضمان عدم قدرة أي مُستخدم على الوصول إلى بيانات الآخر دون إذن صريح. ويُتيح هذا الإعداد فائدة كبيرة تتمثّل في السماح للمُستخدمين بمشاركة البرامج والبيانات، وخفض التكلفة للمُستخدم الواحد عن طريق توزيعها على العديد من المُستخدمين. وقد حقق مصمّمون نظم التشغيل الأولى ذلك عن طريق عزل كلّ برنامج قيد التنفيذ في منطقة خاصة من الذاكرة مُحدّدة بعنوان أساسي وطول مُعين. وكانت أعداد الأساس والطول توضع في سجلّ في وحدة المعالجة المركزية بحيث تقتصر جميع عمليات الوصول إلى الذاكرة من وحدة المعالجة المركزية على المنطقة المُحدّدة من الذاكرة. وهذه الفكرة الخاصة بتقسيم الذاكرة وإعداد الأجهزة، بحيث يستحيل على وحدة المعالجة المركزية الوصول إلى خارج ذاكرتها الخاصة، كانت

بالغة الأهمية لحماية البيانات. ولم يؤد ذلك فقط إلى حماية برامج المُستخدم بعضها من بعض، بل أيضًا حماية المُستخدمين من البرامج غير الموثوق بها، والتي يمكن حصرها في منطقة الذاكرة الخاصة بها.

يُدرِك مستخدمو الأجهزة والشبكات اليوم أنهم يُشاركون أجهزتهم وشبكاتهم مع العديد من الأشخاص. ويفترضون أن نظم التشغيل والشبكات تفرض مبدأ العزل من خلال الاحتفاظ بالبرامج قيد التنفيذ في مناطق ذاكرة خاصة. وعندما يقومون بتنزيل برنامج جديد لا يثقون به، فإنهم يتوقعون أن يعزل نظام التشغيل البرنامج الجديد في منطقة ذاكرة يُطلق عليها «بيئة الاختبار المعزولة».

على الرغم من أن عزل البرامج بواسطة نظم التشغيل موجود في تفكيرنا الحوسبي منذ فترة طويلة، فإن العديد من شرائح الكمبيوتر المُصمَّمة في ثمانينيات القرن العشرين قد أسقطت من حساباتها التحقق من حدود الذاكرة في سبيل زيادة السرعة. وبأسف العديد من خبراء الأمان الآن على هذا. وقد تُطبق الأجيال الجديدة من الأجهزة مرةً أخرى عمليات التحقق من الأمان التي يتوقع المستخدمون أن تكون موجودة في أجهزتهم بسبب خبرتهم في التفكير الحوسبي.

(٥) ما بعد بنية فون نيومان

من أكثر تعريفات التفكير الحوسبي الجديدة شيوعًا أنه يعني «صياغة المشكلات بحيث يمكن التعبير عن حلولها على شكل خطوات حوسبيّة تنفذها آلة». ويرتبط هذا التعريف ارتباطًا وثيقًا بإطار عمل بنية فون نيومان. في الواقع، يُعدّ التعريف تعميمًا لآلية عمل وحدة المُعالجة المركزية في آلة فون نيومان.

بعد نصف قرن، أوشكت بنية فون نيومان على نهايتها. وهناك سببان رئيسيان لذلك. أحدهما هو أن تقنية الشرائح التي تقوم عليها بنية فون نيومان، والتي تضاعف عدد مكوناتها كل عامين وفقًا لقانون مور، لم يُعد بإمكانها استيعاب التقليل المستمر في حجم المكونات. ستُصبح المكونات قريبًا صغيرة جدًا بحيث لا يمكن أن تتضمن عددًا كافيًا من الذرات يسمح لها بالعمل بكفاءة. وقد حفزت النهاية الوشيكة لقانون مور إجراء أبحاث مكثفة حول بنى بديلة.

السبب الآخر هو أن الفصل بين المعالج والذاكرة في بنية فون نيومان يخلق حركة بياناتٍ ضخمة بين المعالج والذاكرة. ومن التقنيات التي ابتُكرت لتقليل هذا الاختناق

الذاكرة المؤقتة، التي تحتفظ بالبيانات في وحدة المعالجة المركزية بدلاً من إعادتها إلى الذاكرة. وثمة تقنية أخرى تدمج المعالج والذاكرة في مصفوفة خلوية لتوزيع حمل البيانات على العديد من القنوات الأصغر بين المعالج والذاكرة. وتقنية ثالثة هي الشرائح ذات الأغراض الخاصة؛ وهي عبارة عن شرائح تؤدي عملاً محدداً بكفاءة منقطعة النظير ولكنها في حد ذاتها ليست أجهزة كمبيوتر ذات أغراض عامة. ومن أمثلتها وحدات معالجة الرسومات التي تدخل الآن في كل كمبيوتر مزود بشاشة عرض رسومية. وتخضع حالياً المعالجات ذات الأغراض الخاصة نفسها لبحث مكثف.

حظيت فئتان جديدتان من بنية الكمبيوتر باهتمام خاص. وكلتاهما تُمثلان تحدياً مُحتملاً للتفكير الحوسبي الحالي. إحداها هي الشبكات العصبية، التي كانت القوة الدافعة وراء التطورات الحديثة في مجال الذكاء الاصطناعي. تُعَيِّن الشبكة العصبية أنماط بت كبيرة (على سبيل المثال: وحدات البت لصورة فوتوغرافية) إلى أنماط بت أخرى (على سبيل المثال: الوجوه المميزة في الصورة). وتنتقل إشارات المُدخلات عبر طبقات متعددة حيث تدمج وفقاً للأوزان المُعينة لها. وتُدرب خوارزمية خارجية الشبكة من خلال إتاحة عدد كبير من أزواج المُدخلات والمُخرجات وتعيين الأوزان الداخلية بحيث تُعَيِّن الشبكة كل مُدخل بشكل صحيح إلى المُخرج المقابل له. ويُعتبر تدريب الشبكة عمليةً حوسبيةً معقدة، حيث تستغرق مدةً تتراوح ما بين عدة ساعات وعدة أيام. وتكون الشبكة المدربة سريعة جداً، حيث تُنتج مُخرجها فور تقديم المُدخل تقريباً. وقد نجحت شرائح معالجة الرسومات في تحقيق استجابة سريعة من الشبكات العصبية المدربة. وعلى الرغم من أن الآلات القادرة فقط على مطابقة الأنماط والتعرُّف عليها ليست أجهزة كمبيوتر ذات أغراض عامة، فإنها حققت تقدماً مذهلاً في أتمتة بعض المهام المعرفية البشرية، مثل التعرُّف على الوجوه. ومع ذلك، لا توجد آلية للتحقق من أن الشبكة العصبية ستقدم المُخرج المناسب عند تقديم مُدخل غير موجود في مجموعة البيانات التي تدربت عليها. كما أن من المُزعج جداً لطريقة تفكيرنا الحوسبية أن نعجز عن «تفسير» كيفية توليد شبكة حوسبية لنتائجها.

أدى اختراع الكمبيوتر الإلكتروني بالكامل ذي البرامج المُخزنة إلى تغيير مفهوم الحوسبة نفسه، وخلق عالماً جديداً من المفاهيم الحوسبية التي كان لها عدد قليل من النظائر أو النماذج الأولية السابقة عليها. وسرعان ما انحرفت المفاهيم والممارسات والمهارات الخاصة بتصميم البرامج وأجهزة الكمبيوتر عن الرياضيات والمنطق. وكان هذا تغييراً جذرياً.

تستخدم بنية الكمبيوتر الأخرى التي تحظى باهتمام خاص تأثيرات ميكانيكا الكم لمعالجة البيانات. وتمثل هذه الآلات الكمّية وحدات البت بدوران الإلكترون والاتصالات بالتأثيرات الكمّية مثل التشابك. ويمكن لأجهزة الكمبيوتر الكمّية إجراء بعض عمليات الحوسبة بشكل أسرع بكثير من أجهزة فون نيومان.

من أمثلة هذه العمليات تحليل عدد مركب كبير إلى العددين الأوليين المكوّنين له. لطالما كانت صعوبة التحليل في بنى فون نيومان هي المبدأ وراء أمان نظام التشفير «آر إس إيه»، والذي يُعد حاليًا أكثر نظم التشفير الواسعة الانتشار أمانًا. وتُهدد أجهزة الكمبيوتر الكمّية باختراق أمانه. ونظرًا إلى أن طريقة عملها مختلفة تمامًا عن أجهزة فون نيومان، يجد معظم الأشخاص المُختصّين في علوم الكمبيوتر وليس الفيزياء صعوبة كبيرة في فهم طريقة عمل هذه الآلات أو كيفية برمجتها.

يوضّح هذان المثالان كيف أن لكل نوع من الآلات أسلوب تفكير حوسبي مُرتبطًا به، وهو جيد جدًا في أنواع مُعينة من المشكلات. والشخص الذي يتمتّع بمعرفة مُتقدمة في التفكير الحوسبي سيكون على دراية بهذه البنى، وكجزء من عملية التصميم، سيختار أفضل بنية لحل المشكلة. في الوقت نفسه، يمكن لأنواع مُعينة من الآلات أن تُسبب نوعًا من «الغشاوة»؛ على سبيل المثال: يفكر المُصمّمون المدربون على بنية فون نيومان الأساسية بطريقة تعتمد على التعليمات ويجدون صعوبة في فهم كيفية عمل الكمبيوتر الكمّي.

حتى أربعينيات القرن العشرين، كانت الحوسبة تُعد مهمةً عقلية لا يستطيع القيام بها إلا البشر، وفرعًا من فروع الرياضيات والمنطق. وأدّى اختراع الكمبيوتر الإلكتروني بالكامل ذي البرامج المُخزّنة إلى تغيير مفهوم الحوسبة نفسه، وخلق عالمًا جديدًا من المفاهيم الحوسبيّة التي كان لها عدد قليل من النظائر أو النماذج الأولية السابقة عليها. وسرعان ما انحرفت المفاهيم والممارسات والمهارات الخاصة بتصميم البرامج وأجهزة الكمبيوتر عن الرياضيات والمنطق. وكان هذا تغييرًا جذريًا.

وحتى أربعينيات القرن العشرين، كان التفكير الحوسبي مدمجًا في المعرفة الضمنية والممارسات المتطورة للعديد من المجالات المختلفة، التي من بينها الرياضيات والمنطق والهندسة والعلوم الطبيعية. وبعد الأربعينيات، بدأ التفكير الحوسبي يُصبح محور المهنة الجديدة التي صممت الآلات والأجهزة المعلوماتية للقيام بأعمال لم يتصوّر البشر أبدًا إمكانية القيام بها.

الفصل الرابع

علوم الكمبيوتر

يُعد السؤال «ما الذي يمكن أتمتته؟» أحد أكثر الأسئلة الفلسفية والعملية إلهاً في الحضارة المعاصرة.

جورج فورسايت (١٩٦٩)

في الخمسينيات من القرن العشرين، بدأ الأكاديميون يدعون إلى إنشاء برامج لتدريس علوم الكمبيوتر في الجامعات لتلبية الحاجة المتزايدة إلى تعلّم التقنيات الجديدة. ومنذ ذلك الحين، صُقلت العديد من مبادئ التفكير الحوسبي في أقسام علوم الكمبيوتر وطوّرت. ننتقل الآن إلى كيفية تطوير التفكير الحوسبي في الجامعات.

قبل أن نبدأ، نودُّ أن نُسلط الضوء على بعض الجوانب الرئيسية للبيئة الأكاديمية التي تطوّر فيها التفكير الحوسبي. أولاً وقبل كل شيء، الحوسبة هي مجال تقني يجمع بين الهندسة والعلوم والرياضيات. وقد أتى معظم طلاب الحوسبة إلى الجامعة لتعلّم مهنة تصميم البرامج والأجهزة، وليس للحصول على تعليم عام. كما يأتي أصحاب الشركات إلى الجامعة لتوظيف الخريجين في شركاتهم. لذلك، فإن التفكير الحوسبي الذي تطوّر جنباً إلى جنب مع الحوسبة الأكاديمية كان دائماً مرتبطاً ارتباطاً قوياً بالتصميم ومتأثراً بشدة بما يقول أصحاب الشركات أنهم يحتاجونه.

ولكن هذا ليس كل شيء. تُنظّم الجامعات في مجموعة من الأقسام حسب التخصص ومجموعة من المعاهد والمراكز المتعددة التخصصات. وتدافع الأقسام باستماتة عن هويتها وميزانياتها ومساحتها. ولأن ميزانياتها تعتمد على التحاق الطلاب بها، فإنها تكافح من أجل الحفاظ على عدد المُلتحقين بها. ولأن عدد الطلاب المُلتحقين يعتمد على سمعة القسم والسمعة تعتمد على الإنتاجية البحثية، فإن الأقسام تدافع عن نطاقات بحثها.

جانب مُهم آخر من جوانب تشكيل البيئة الأكاديمية هو السعي وراء توافق الآراء في جميع القرارات. فالجميع يُريدون أن يُدلووا بدلوهم، سواء كان ذلك عند توظيف شخصٍ جديد، أو منح رتبةٍ علمية، أو تحديد الدورات التي ستُقدمها، أو الموافقة على الدورات المتداخلة المُحتملة التي تقترحها الأقسام الأخرى، أو الموافقة على إنشاء برامج أو أقسام جديدة.

هذا هو الجوُّ الذي تشكلت فيه أقسام علوم الكمبيوتر الجديدة وتشكل فيه التفكير الحوسبي الأكاديمي. شعر المؤسسون بالقلق بشأن المناهج الدراسية ومُتطلبات الصناعة في سياق مجموعةٍ من الأقسام الساعية إلى توافق الآراء والمدافعة عن امتيازاتها بقوة، والمُهتمة دائماً بصورتها العامة وهويتها.

اقترح المؤسسون أقساماً جديدة مُنفصلة عن الأقسام الرئيسية الموجودة. وكانت الأقسام الرئيسية في الغالب تُعارض هذا الانفصال؛ لأنها ستفقد الطلاب والميزانية والهوية. وواجه المؤسسون الكثير من المعارضة من الأقسام الأخرى التي لم تُعدّ تكنولوجيا الكمبيوتر علماً منفصلاً عن العلوم أو الهندسة، ولم ترَ أنه سيقدم منظوراً فكرياً فريداً. وكان الوصول إلى توافق آراء تؤيد تشكيل قسمٍ جديد يُمثل تحدياً. ومن ثَمَّ أمضى المؤسسون الكثير من الوقت في مناقشة جوهر الحوسبة، ولماذا كان مختلفاً وجديداً، وكيف سيُفيد المجالات الأخرى. وقَدِّموا حجةً وجيهة ونجحوا في مهمَّتهم. وببطءٍ تزايد عدد أقسام علوم الكمبيوتر، من قسمٍ واحد في عام ١٩٦٢ إلى حوالي ١٢٠ قسمًا في عام ١٩٨٠ في الولايات المتحدة وحدها. وفي أواخر التسعينيات، انطلقت علوم الكمبيوتر عندما أدرك الناس أخيراً أن ثورة الحوسبة حقيقية. واليوم، يُوجد في كل الجامعات تقريباً قسم لعلوم الكمبيوتر.

تُوجد أقسام علوم الكمبيوتر في كليات العلوم والهندسة وحتى التجارة. فلماذا هذه الأماكن المُتعددة؟ الإجابة تعكس تلك المعارك السياسية المبكرة؛ حيث أنشئت الأقسام الجديدة في الكليات الأكثر ترحاباً. ونظراً إلى أن معظم الأقسام كانت في كليات العلوم والهندسة، فبحلول الثمانينيات من القرن العشرين، كان علماء الكمبيوتر يُطلقون على مجالهم «علوم الكمبيوتر والهندسة». وتم تبسيط هذا الاسم الطويل في التسعينيات حتى أصبح مُصطلح «الحوسبة» اختصاراً شائعاً لعلوم الكمبيوتر والهندسة، بينما انتشر استخدام مصطلح «علم المعلومات» في أوروبا. في التسعينيات من القرن العشرين، ذهبت بعض الجامعات إلى أبعد من ذلك وأنشأت كليات مُنفصلة للحوسبة، وما زال هذا مُستمراً إلى اليوم. فيا له من تحول!

في بداية الأمر، تأسست جمعيتان أكاديميتان للكمبيوتر: أولاهما: جمعية الكمبيوتر التابعة إلى معهد مهندسي الكهرباء والإلكترونيات، وتأسست في عام ١٩٤٦؛ والثانية: جمعية آلات الحوسبة، وتأسست في عام ١٩٤٧. وبسبب اجتهداهما في تطوير وتعزيز توصيات المناهج الدراسية، كانت هناك سلسلة من اللقطات للمناهج الدراسية للحوسبة على فترات مُنتظمة: ١٩٦٨، و١٩٧٨، و١٩٨٩، و١٩٩١، و٢٠٠١، و٢٠١٣. تُظهر هذه اللقطات كيف أدّى تنسيق الجهود التي بذلها رواد الحوسبة في خلق هوية فريدة لعلوم الكمبيوتر إلى إدراكهم للتفكير الحوسبي كجانبٍ مُميز من البداية. وبالتدقيق، يُمكننا تمييز أربعة عصور تصف كيف فُكّرت الجامعات في الحوسبة وكيف تغيّرت هذه الآراء حول التفكير الحوسبي:

- الظواهر المحيطة بأجهزة الكمبيوتر (خمسنيات وسبعينيات القرن العشرين).
- البرمجة فنًا وعلمًا (سبعينيات القرن العشرين).
- الحوسبة باعتبارها عملية أتمتة (ثمانينيات القرن العشرين).
- الحوسبة بوصفها عملياتٍ معلوماتية شاملة (تسعينيات القرن العشرين وحتى الوقت الحاضر).

وستتناول هذه العصور بالمناقشة في الأقسام التالية. تأثرت هذه المراحل الأربع لتطوُّر التفكير الحوسبي في الجامعات بشدة بالمقاومة الأولية التي واجهتها علوم الكمبيوتر من قِبل المجالات الأخرى؛ فقد بذل علماء الكمبيوتر الأكاديميون الكثير من الجُهد في توضيح مجالهم وتبريره. ولكن علوم الكمبيوتر لم تكن دائمًا المُستقبلية لسهام المقاومة. هناك حالتان مُهمّتان كانت فيهما علوم الكمبيوتر هي المُوجّهة لهذه السهام. الأولى: كانت حركة علوم الحوسبة في ثمانينيات القرن العشرين، والتي رفضها العديد من علماء الكمبيوتر. كان رد الفعل الشائع لإعلان من قسم الفيزياء أو الأحياء بأنهم ينشئون فرعًا لعلوم الحوسبة هو صرخة احتجاج على أن تلك الأقسام تتعدّى على أراضي الحوسبة. فقد اعتقد بعض علماء الكمبيوتر أن الفيزياء والبيولوجيا، بعد أن أدركا أهمية الحوسبة، يُحاولان الاستيلاء على المجال الذي عارضوه بشدة من قبل. وفي النهاية، تغلّب علماء الكمبيوتر على هذا الأمر وأصبحوا الآن يتعاونون مع علوم الحوسبة. وسوف نتحدّث عن العلوم الحوسبيّة في الفصل السابع.

حدث الشيء نفسه مع هندسة البرمجيات. فقد كان هناك تفرقة واضحة بين علوم الكمبيوتر وهندسة البرمجيات في الماضي؛ إذ إن أقسام علوم الكمبيوتر كانت تُركز على

الأبحاث النظرية وكثيراً ما كانت تقاوم المناهج العملية القائمة على المشاريع والتي يُفضلها مهندسو البرمجيات. وقد أدى هذا إلى جدل مُستمر بشأن ما إذا كانت هندسة البرمجيات يجب أن تكون مجالاً مُنفصلاً عن علوم الكمبيوتر، أم جزءاً منها. وسوف نتحدث عن هذا الموضوع في الفصل الخامس.

(١) الظواهر المُحيطة بأجهزة الكمبيوتر

سرعان ما أدرك مُطوِّرو أجهزة الكمبيوتر الآلية المبكرة أن الآلات الجديدة تتطلب طريقة تفكير وتصميم مُختلفة عن أي شيء موجود بالفعل في العلوم أو الهندسة. وأنشأت جمعية آلات الحوسبة ومعهد مهندسي الكهرباء والإلكترونيات دوريات علمية لهذا المجال الناشئ في بداية الخمسينيات من القرن العشرين. وكانت كلية مور، موطن مشروع المُكامل الرقمي الإلكتروني، من أولى الجهات التي بدأت تعليم الحوسبة في عام ١٩٤٦ بدورة مكثفة مدتها شهران حول «نظرية تصميم أجهزة الكمبيوتر الرقمية الإلكترونية وتقنياتها». وفي الخمسينيات من القرن العشرين، منحت كلية مور درجة علمية متعددة التخصصات في الحوسبة تضمنت التحليل العددي، والبرمجة، وتصميم لغة البرمجة. وبدأت كليات أخرى برامجها الخاصة.

لم تُثمر هذه الجهود المبكرة لإرساء الحوسبة كتخصص أكاديمي مستقل إلا عن نتائج بطيئة. ولم يكن ما يعوق تقدُّمها هو فقط التردد الحذر للتأكد من أن أجهزة الكمبيوتر جاءت لتبقى؛ بل كان شكاً عميقاً فيما إذا كانت الحوسبة ذات جوهر أكاديمي يتجاوز الرياضيات والهندسة الكهربائية والفيزياء. وكان الذين هم خارج مجال الحوسبة في خمسينيات القرن العشرين يرون أنه غابة فوضوية كثيفة من الحيل التكنولوجية الخاصة لا يمكن اختراقها. علاوة على ذلك، كانت وجهات النظر المُختلفة للتفكير في الحوسبة مُنقسمة: فكان أولئك الذين صمَّموا آلات الحوسبة غير مُدركين في الغالب للتطورات المُهمّة في نظرية الحوسبة؛ مثل تورينج فيما يختص بالأعداد القابلة للحوسبة، وتشيرش فيما يختص بحساب لامدا، وبوست فيما يختص بمعالجة سلاسل الأحرف، وكلين فيما يختص بالتعبيرات العادية، ورايين وسكوت فيما يخص الآلات غير المُحدَّدة، وتشومسكي فيما يخص العلاقة بين القواعد اللغوية وأنواع الآلات الذاتية التشغيل.¹

واجه الأكاديميون الذين اقترحوا أقساماً أو برامج كاملة لعلوم الكمبيوتر في الجامعات البحثية مقاومة قوية. فلم يُصدق العديد من المُنتقدين قيمة الطرق الجديدة للحوسبة؛

ومن الاعتراضات الشائعة: الافتقارُ إلى المحتوى الفكري الفريد والافتقار إلى الأساس النظري الكافي. وجادل الأصوليون بأن أجهزة الكمبيوتر كانت مصنوعات بشرية وليست أحداثاً طبيعية، ومن ثم لا يمكن اعتبار دراستها ضمن العلوم الطبيعية الغراء. علاوة على ذلك، شكَّ الكثيرون في استمرار الحوسبة. ولم يتمكن أحد من تأسيس قسم علوم الكمبيوتر إلا بعد أن توصَّلت العديد من الأقسام إلى توافق في الآراء.

بدأ هذا التيار في التغيُّر عام ١٩٦٢، عندما أنشأت جامعة بورديو أول قسم لعلوم الكمبيوتر وتبعتها ستانفورد بعد ذلك بوقت قصير. وعلى مدار العقدين التاليين، زاد عدد الأقسام، ببطء ولكن بثبات، إلى أكثر من مائة في الولايات المتحدة الأمريكية وحدها. ومع ذلك، واصل العديد من الأكاديميين التشكيك فيما إذا كانت علوم الكمبيوتر مجالاً علمياً أو هندسياً مستقلاً.

حدث تحول رئيسي في السؤال حول أحقية الحوسبة بالاستقلال عام ١٩٦٧، عندما نشر ثلاثة من علماء الكمبيوتر المعروفين — ألن نيُوويل، وألان برليس، وهيربرت سايمون — رسالة شهيرة في مجلة «ساينس» تتناول هذا السؤال. كتبوا: «أينما وُجدت الظواهر، يمكن أن يكون هناك علم لوصف تلك الظواهر وتفسيرها. ومن ثمَّ، فعلم النبات هو دراسة النباتات، وعلم الحيوان هو دراسة الحيوانات، وعلم الفلك هو دراسة النجوم، ... وهكذا. الظواهر تولد العلوم ... تُوجد أجهزة كمبيوتر. ومن ثمَّ، وجدت علوم الكمبيوتر لدراسة أجهزة الكمبيوتر. فالظواهر المحيطة بأجهزة الكمبيوتر مُتنوعة ومعقدة وغنية»² من هذا الأساس، سرعان ما رفضوا ستة اعتراضات، بما في ذلك الاعتراض على أن أجهزة الكمبيوتر من صنع الإنسان، وبناءً عليه فليست موضوعات تستحق أن ينشأ لها علم مستقل. واعترض هيربرت سايمون، الحائز على جائزة نوبل في الاقتصاد، بشدة على فكرة أنه لا يمكن أن يكون هناك علم يُحيط بالأشياء التي صنعها الإنسان، لدرجة أنه كتب كتاباً أضحى من الكتب الكلاسيكية حالياً بعنوان «علوم الاصطناعي» يدحض هذه الفكرة.³ وأعطى مثلاً على نظم المشاركة الزمنية (أجهزة الكمبيوتر التي تسمح بالعديد من المُستخدمين المتزامنين):

لم يكن من الممكن أن يهتدي التطوير المبكر لنظم المشاركة الزمنية بالنظريات؛ حيث لم تكن ثمة نظريات موجودة، وكانت معظم التوقُّعات حول سلوك نظم المشاركة الزمنية غير دقيقة بالمرة. لم يكن من الممكن وضع نظرية لنظم المشاركة الزمنية دون إنشاء هذه النظم بالفعل، وبعد إنشائها، أدى البحث التجريبي حول سلوكها إلى وضع

أساس نظري ثري. بعبارة أخرى، لا يمكن للتفكير الحوسبي أن يتعامل مع المشكلات من اتجاه واحد فقط؛ ومن هنا تطوّرت الجوانب الهندسية والجوانب العلمية الرياضية للحوسبة بطريقة تآزرية لإنتاج علم لم يكن علماً طبيعياً بحثاً.

سرعان ما اكتسبت فكرة الحوسبة، بوصفها دراسة للظواهر المحيطة بأجهزة الكمبيوتر، قبولاً، وبحلول نهاية الستينيات من القرن العشرين، اعتُبرت تعريفاً للحوسبة. وبدأت رؤية فريدة للمجال تتشكّل حول هذه الفكرة. واستُخدم مصطلح «التفكير الخوارزمي» لوصف الجانب الأكثر وضوحاً لهذا النوع الجديد من التفكير. كانت الأهداف الفريدة للمجال، والمشكلات المعتادة فيه، وأساليب حل هذه المشكلات، وأنواع الحلول هي أساس التفكير الحوسبي.

وسّع رواد الحوسبة التفكير الحوسبي إلى أبعد مما ورثوه من التاريخ الطويل للحوسبة. وركّزوا على مبادئ إنشاء البرامج وآلات الحوسبة ونظم التشغيل. ووضعوا عدداً كبيراً من مفاهيم الحوسبة التي تُعد اليوم من المُسلّمات، بما في ذلك المتغيرات المُسمّاة، وهياكل التحكم، وهياكل البيانات، وأنواع البيانات، ولغات البرمجة الشكلية، والروتينات الفرعية، وبرامج التحويل البرمجي، وبروتوكولات الإدخال والإخراج، وقنوات التعليمات، ونظم المقاطعة، وعمليات الحوسبة، والتسلسل الهرمي للذاكرة، والذاكرة المؤقتة، والذاكرة الافتراضية، والأجهزة الطرفية، والواجهات.

وكانت أساليب البرمجة وبنى نظم الكمبيوتر هي القوى الدافعة الرئيسية في تطوير التفكير الحوسبي. وبحلول عام ١٩٧٠، اتفق معظم علماء الكمبيوتر على أن طرق الحوسبة المميزة في التفكير والممارسة — التي يُطلق عليها اليوم التفكير الحوسبي — تشمل كل المعارف والمهارات المتعلقة بأجهزة الكمبيوتر.

انقسم التفكير الحوسبي في وقتٍ مبكرٍ إلى جانبين: الأجهزة والبرامج. وتابع مهندسو الكمبيوتر جانب الأجهزة في كلية الهندسة؛ بينما تابع مُصمّمو البرامج وواضعو نظريات الحوسبة الجانب الخاص بالبرامج في كلية العلوم.

(٢) البرمجة فناً وعلماً

كانت الستينيات من القرن العشرين فترة نُضج للحوسبة، أنتجت ثراءً كبيراً في الطرُق التي فكّر بها علماء الكمبيوتر في عملهم ومجالهم. فقد ظهر فرع نظم التشغيل في أوائل الستينيات من القرن العشرين لإتاحة الحوسبة التفاعلية الرخيصة في مجتمعات

المستخدمين الكبيرة؛ ومن ثم اكتسب التفكير الحوسبي توجهًا متكاملًا يقوم على المنظومة الكاملة. وظهر فرع هندسة البرمجيات في أواخر الستينيات من القرن العشرين بسبب القلق من أن النماذج الحالية للبرمجة غير قادرة على تطوير برامج إنتاج موثوق بها يُمكن الاعتماد عليها؛ ومن ثم اكتسب التفكير الحوسبي توجهًا هندسيًا. وظهر فرع الشبكات في عام ١٩٦٧ عندما بدأ مشروع شبكة وكالة مشاريع الأبحاث المتقدمة (أربانت)، وعندها اكتسب التفكير الحوسبي توجهًا شبكيًا.

مع وجود أساس تكنولوجي قوي وموثوق به، تحول اهتمام المجال إلى البرامج والبرمجة. وظهرت العديد من لغات البرمجة جنبًا إلى جنب مع طرق البرمجة القياسية. ونشأ اهتمام كبير بالتحقق الشكلي من البرامج، سعيًا وراء طريقة قائمة على النظرية لإثبات أن البرامج موثوق بها وصحيحة. كما نشأ اهتمام مُماثل بالتعقيد الحوسبي، سعيًا وراء طرق تحليلية لتقييم مقدار العمل الحوسبي الذي تتطلبه الخوارزميات المختلفة. البرامج هي تعبيرات عن الخوارزميات بلغة شكلية تتحكم في أفعال الآلة عند ترجمتها إلى شكل قابل للتنفيذ بواسطة الآلة. والبرامج ضرورية لمُعظم عمليات الحوسبة؛ إذ يستخدم معظم المحترفين والباحثين في الحوسبة البرامج بطريقة أو بأخرى. في أجهزة الكمبيوتر الأولى ذات البرامج المُخزنة التي كانت موجودة في الأربعينيات من القرن العشرين، كانت البرمجة تُنفَّذ بلُغات التجميع التي كانت تُحوَّل الأكواد القصيرة المختصرة للتعليمات سطرًا تلو الآخر إلى تعليمات آلة يستطيع الكمبيوتر تشغيلها. على سبيل المثال: سيضع الأمر $ADD\ R1, R2, R3$ مجموع السجلين $R1$ و $R2$ في السجل $R3$. وكان هذا الأمر يُحوَّل إلى تعليمات آلة عن طريق استخدام ADD و $R1$ و $R2$ و $R3$ بدلًا من الرموز الثنائية. وكانت كتابة البرامج بلغة التجميع أمرًا مُملًا للغاية ومُعَرِّضًا للأخطاء.

اخترعت لغات البرمجة لتوفير تعبيرات دقيقة عالية المستوى لما يريد المبرمج، والتي يمكن ترجمتها بشكل لا التباس فيه بواسطة برنامج التحويل البرمجي إلى تعليمات آلة. وقد أدى ذلك إلى تبسيط وظيفة البرمجة تبسيطًا هائلًا، مما جعلها أكثر إنتاجية وأقل عرضة للأخطاء. وقد أدخلت لغات البرمجة الأولى مجموعة كبيرة من مفاهيم التفكير الحوسبي الجديدة التي لم تكن لها نظائر في المجالات الفكرية الأخرى، أو كانت لها نظائر قليلة.

استهدفت معظم لغات البرمجة المساعدة في أتمتة الوظائف المهمة مثل تحليل البيانات العلمية وتقييم النماذج الرياضية (فورتران في عام ١٩٥٧)، وإجراء الاستنتاجات المنطقية

(ليسب في عام ١٩٥٨)، أو تتبع عمليات الجرد في الشركات والحفاظ على قواعد بيانات العملاء (كوبول في عام ١٩٥٩). وهدفت لغات قليلة إلى السماح للأشخاص بتحديد المواصفات الدقيقة للخوارزميات التي يمكن دمجها في لغاتٍ أخرى. وطُورت لغة ألجول (١٩٥٨) من هذا المنظور.

أصبحت فكرة أن اللغات توفر متطلبات طرق مُعينة للتفكير في المشكلات تُسمَّى «أنماط البرمجة». على سبيل المثال: البرمجة الأمرية ترى البرامج كسلسلة من الوحدات النمطية (تُسمى «الإجراءات») التي تُوجَّه أوامر إلى الآلة عن طريق تعليماتها. وتنتمي لغات البرمجة «فورتران» و«كوبول» و«ألجول» جميعها إلى هذه الفئة. أما البرمجة الموجهة نحو الكائنات، فهي تتعامل مع البرامج كمجموعاتٍ من الوحدات شُبه المُستقلة ذاتياً، أو «الكائنات» التي تتفاعل بعضها مع بعض ومع العالم الخارجي عن طريق تبادل الرسائل. وتنتمي اللغات اللاحقة مثل «سمولتوك» و«جافا» إلى هذه الفئة. وتعاملت البرمجة الوظيفية مع البرامج كمجموعاتٍ من الدوال الرياضية التي تولّد بيانات الإخراج من بيانات الإدخال. ومن أمثلتها لغة «ليسب».

كان يُنظر إلى أنماط البرمجة في السبعينيات من القرن العشرين على أنها أنماط مختلفة من التفكير الخوارزمي. وكانت جميعها تهدف إلى برامج تحتوي على تعبيرات واضحة يستطيع البشر قراءتها وتنفيذها بشكلٍ صحيح وفَعَال عند تجميعها وتنفيذها. وقد قام كلٌّ من دونالد كنوث في كتابيه الشهيرين «فن برمجة الكمبيوتر» و«البرمجة الحرفية»، وإدسجر دايكسترا في عمله عن البرمجة الهيكلية، بتجسيد الفكرة القائلة بأن الحوسبة تدور حول الخوارزميات بهذا المعنى. وبحلول ثمانينيات القرن العشرين، أوضح معظم علماء الكمبيوتر أن التفكير الحوسبي هو مجموعة من المهارات والمعرفة المتعلقة بالخوارزميات وتطوير البرامج.

ولكن الأمور أصبحت صعبةً عندما اضطرَّ أنصار التفكير الخوارزمي إلى وصف المقصود بالتفكير الخوارزمي وأوجّه اختلافه عن أنواع التفكير الأخرى. قارن كنوث أنماط التفكير في كتب الرياضيات وكتب الحوسبة، وحدّد الأنماط النموذجية في كليهما.⁴ وخلص إلى أن التفكير الخوارزمي يختلف عن التفكير الرياضي في العديد من الجوانب: من خلال الطرق التي يختزل بها المشكلات المُعقدة إلى مشكلات بسيطة مترابطة، ويركز على هياكل المعلومات، ويهتم بكيفية تأثير الإجراءات على حالات البيانات، ويصوغ تمثيلاتٍ رمزية للواقع. في دراساته الخاصة، فرّق دايكسترا بين علماء الكمبيوتر والرياضيين من خلال

قدرتهم على التعبير عن الخوارزميات بلغاتٍ طبيعية وكذلك شكلية، وصياغة الرموز التي تبسط عمليات الحوسبة، والتغلب على التعقيد، والانتقال بين مستويات التجريد، وابتكار المفاهيم والكائنات والرموز والنظريات عند الضرورة.⁵

لم تُعد الأوصاف الحالية لأدوات التفكير الحوسبي العقلية موجهةً نحو الرياضيات مثلما كانت الكثير من الأوصاف القديمة للتفكير الخوارزمي. على مرّ الزمن، رأى الكثيرون أن البرمجة والتفكير الخوارزمي مهمّان مثل القراءة والكتابة والحساب — وهي الأسس الثلاثة التقليدية للتعليم — ولكن اقترح إضافتهما كأساسين جديدين إلى هذه القائمة لم يُقبل بعد. ولدى رواد الحوسبة تاريخ طويل من الخلاف حول هذه النقطة. اعتبر بعض رواد الحوسبة أن طريقة التفكير الحوسبي أداة عامة للجميع، مثلها مثل الرياضيات واللغة.⁶ واعتبر البعض الآخر أن التفكير الخوارزمي هو قدرة فطرية نادرة إلى حدٍّ ما؛ يتمتع بها شخص واحد تقريباً من كل خمسين شخصاً.⁷ ويدعم الرأي الأول عددٌ أكبر من المُختصين في التعليم لأنه يتبنّى فكرة أن الجميع يمكنهم تعلّم التفكير الحوسبي؛ فالتفكير الحوسبي هو مهارة يجب تعلّمها وليس ملكة يولد بها المرء.⁸

أنتجت رؤية البرمجة والخوارزميات للحوسبة إضافاتٍ جديدة إلى مجموعة أدوات التفكير الحوسبي. وقدّم الجانب الهندسي والتقني برامج التحويل البرمجي (لتحويل البرامج التي يستطيع البشر قراءتها إلى تعليمات آلة قابلة للتنفيذ)، وأساليب التحليل (لتقسيم عبارات لغة البرمجة إلى عناصرها الأساسية)، وتحسين التعليمات البرمجية، ونظم التشغيل، وأساليب الاختبار وتصحيح الأخطاء التجريبية (للعثور على الأخطاء في البرامج). بينما قدّم الجانب الرياضي العلمي مجموعة من أساليب تحليل الخوارزميات مثل الترميز O لتقدير كفاءة الخوارزميات، ونماذج مختلفة للحوسبة، وإثباتات لصحة البرامج. وبحلول أواخر السبعينيات من القرن العشرين، كان من الواضح أن الحوسبة تحرّكت على مسارٍ فكري مع مفاهيم ومخاوف ومهارات تختلف تمامًا عن التخصصات الأكاديمية الأخرى.

(٣) الحوسبة كأتمتة

إن رؤية الحوسبة على أنها دراسة الخوارزميات وتصميمها كانت تُعد رؤية ضيقة للغاية، على الرغم من كل ما تنطوي عليه من ثراء. وبحلول أواخر السبعينيات من القرن العشرين، كان هناك العديد من الأسئلة الأخرى التي تحتاج إلى إجابات. كيف نُصمّم

لغة برمجة جديدة؟ كيف تزيد من إنتاجية المبرمج؟ كيف تُصمم نظام تشغيل آمنًا؟ كيف تصمم أجهزةً ونظمًا برمجية قادرة على تحمّل الأخطاء؟ كيف تنقل البيانات بشكل موثوق به عبر شبكة من حُرَم البيانات؟ كيف تحمي النظم من سرقة البيانات من قبل المُتسلّلين أو البرامج الضارة؟ كيف تجد نقاط الاختناق في نظام الكمبيوتر أو في الشبكة؟ كيف تجد وقت استجابة النظام؟ كيف تجعل النظام يؤدّي عملاً سبق إنجازَه بواسطة مُشغّلين بشريّين؟

ركزت دراسة الخوارزميات على خوارزميات فردية ولكنها نادراً ما ركّزت على تفاعلاتها مع البشر أو تأثيرات عمليات الحوسبة الخاصة بها على المُستخدِمين الآخرين للنظم والشبكات. ومن ثم لم يكن من الممكن أن تقدّم إجاباتٍ كاملةً عن هذه الأسئلة. نشأت فكرة أن العامل المشترك في كل هذه الأسئلة، وأن جوهر التفكير الحوسبي، هو أن الحوسبة تسمح بالأتمة في العديد من المجالات. والأتمة، بشكل عام، تعني أحد شيئين: إما التحكم في العمليات بطرق ميكانيكية مع الاستعانة بالبشر في أضيق الحدود، أو تنفيذ العملية بواسطة آلة. وأراد الكثيرون العودة إلى مفهوم الستينيات بأن الأتمة هي الغاية النهائية وراء أجهزة الكمبيوتر وأنها من أكثر المسائل إثارة للاهتمام في العصر الحديث. وكان من الواضح أن الأتمة هي العامل المشترك بين جميع علوم الكمبيوتر، وأن التفكير الحوسبي يهدف إلى جعل الأتمة فعّالة.

في عام ١٩٧٨، أطلقت المؤسسة الوطنية للعلوم بالولايات المتحدة مشروعاً شاملاً لتحديد ما هو ضروري في الحوسبة. وأُطلق على هذا المشروع اسم «دراسة أبحاث علوم الكمبيوتر والهندسة». وفي عام ١٩٨٠، أصدرت كتاب «ما الذي يمكن أتمتته؟» وهو مجلّد من ألف صفحة يتناول بالبحث العديد من جوانب الحوسبة وتطبيقاتها من وجهة نظر الأتمة الفعّالة.⁹ وأجابت هذه الدراسة عن العديد من الأسئلة أعلاه، ولسنواتٍ عديدة، قدم تقرير «دراسة أبحاث علوم الكمبيوتر والهندسة» الصورة الأكثر اكتمالاً للحوسبة والتفكير الحوسبي في ذلك العصر. ولا يزال مرجعاً مهمّاً للغاية لأي شخص يريد موجزاً مكتوباً من قبل رؤاد الحوسبة المشهورين حول العديد من الموضوعات والمشكلات والأسئلة الجوهرية في الحوسبة.

حتى منتصف التسعينيات من القرن العشرين، تم تبني فكرة الحوسبة كأتمتة في الكتب والتقارير البحثية ووثائق السياسة المؤثرة باعتبارها «السؤال الأساسي الكامن وراء الحوسبة». وقد تردّد صدّى هذه الفكرة في تاريخ التفكير الحوسبي؛ وكما ناقشنا في

الفصول السابقة، حققت الحوسبة الآلية حُلْم علماء الرياضيات التطبيقية والمهندسين في سرعة الحساب ودقته دون الاعتماد على الحدس البشري والأحكام البشرية. وكان واضعو النظريات، مثل آلان تورينج، مفتونين بفكرة ميكنة الحوسبة. ورأى الممارسون أن البرامج عبارة عن أتمتة للمهام. وبحلول عام ١٩٩٠، أصبح «ما الذي يمكن أتمتته؟» شعارًا رائجًا في شرح الحوسبة للوافدين وموضوعًا أساسيًا من موضوعات التفكير الحوسبي.

وللمفارقة، أدى سؤال «ما الذي يمكن أتمتته؟» إلى تعطيل تفسير الأتمتة؛ لأن الحدود الفاصلة بين ما يمكن أتمتته وما لا يمكن أتمتته غير واضحة. وما كان من المستحيل أتمتته سابقًا ربما يُصبح ممكنًا الآن بفضل خوارزميات جديدة أو أجهزة أسرع. وبحلول السبعينيات من القرن العشرين، طوّر علماء الكمبيوتر نظريةً غنيةً معنيةً بالتعقيد الحوسبي، وصنّفت هذه النظرية المسائل وفقًا لعدد الخطوات الحوسبية التي تحتاجها الخوارزميات لحلّها. على سبيل المثال: البحث عن عنصرٍ مُعين في قائمة غير مرتّبة مكوّنة من عدد N من العناصر يستغرق وقتًا يتناسب مع عدد N من الخطوات. ويُعد فرز قائمة مكونة من عدد N من العناصر فرزًا تصاعديًا مهمةً أكثر تعقيدًا؛ إذ يستغرق الأمر وقتًا يتناسب مع عدد N^2 من الخطوات في بعض الخوارزميات، وعدد $N \log N$ من الخطوات في أفضل الخوارزميات، فيما يُعرف بدرجة تعقيد الخوارزمية. وتستغرق طباعة قائمة بجميع المجموعات الفرعية لعدد N من العناصر وقتًا يتناسب مع 2^N . تُعدّ مسألة البحث ذات «صعوبة خطية»، ومسألة الفرز ذات «صعوبة تربيعية»، ومسألة الطباعة ذات «صعوبة أُسية». فالبحث سريع، والسرد بطيء؛ ومن وجهة نظر واضعي نظرية التعقيد الحوسبي فإنّ الإجراء الأول «سهل» والأخير «صعب».

لفهم مدى اتساع الفارق بين المسائل السهلة والصعبة، تخيل أن لدينا جهاز كمبيوتر يُمكنه تنفيذ مليار (١٠^٩) تعليمة في الثانية. إن البحث في قائمة مكونة من ١٠٠ عنصر يتطلب ١٠٠ تعليمة أو ٠,١ ميكرو ثانية. أما إعداد قائمة بجميع المجموعات الفرعية المكونة من ١٠٠ عنصر، فإنه يستغرق ١٠٠٢ تعليمة، وهي عملية تستغرق حوالي ١٤١٠ أعوام. وهذا أطول بعشرة آلاف مرة من عمر الكون، الذي يبلغ تقريبًا حوالي ١٠١٠ أعوام. وعلى الرغم من أنه يُمكننا كتابة خوارزمية للقيام بذلك، فلا يوجد كمبيوتر يُمكنه أن يكمل المهمة في وقتٍ معقول. بتطبيق ذلك على مسألة الأتمتة، قد تستغرق خوارزمية الأتمتة وقتًا طويلًا جدًّا إلى الحد الذي يجعلها مستحيلة. ليس بالضرورة أن يكون كل شيء لدينا خوارزمية له قابلاً للأتمتة. وبمرور الوقت، تمكّنت أجيال جديدة من الأجهزة الأكثر قوة من أتمتة المهام التي كان من الصعب أتمتتها سابقًا.

إن الخوارزميات التجريبية تجعل مسألة الصعوبة الحوسبية أكثر إثارة للاهتمام. تُطالبنا مشكلة حقبة الظهر الشهيرة بتعبئة مجموعة من العناصر في حقبة ظهر ذات وزن محدود لاختيار العناصر الأعلى قيمة. والخوارزمية المستخدمة في ذلك تُشبه تلك المستخدمة في مسألة سرد العناصر التي تحدّثنا عنها، وسوف تستغرق وقتاً طويلاً على نحو يستحيل معه تنفيذها بالنسبة لمعظم حقائب الظهر. ولكنّ لدينا روتيناً قائماً على التجربة العملية مفاده أن «نُقيّم كل عنصر بنسبة قيمته إلى وزنه، ثم نضع العناصر في الحقبة حسب الترتيب التنازلي للنسب حتى تمتلئ حقبة الظهر». هذا الروتين القائم على التجربة العملية يؤدي إلى حزم حقائب الظهر بطريقة جيدة جداً وبسرعة، ولكن ليس بالضرورة أن يكون ذلك بالطريقة الأفضل. وهذا هو الحال في العديد من المسائل الصعبة. توجد خوارزميات تجريبية سريعة تعمل جيداً ولكنها قد لا تكون الأفضل. ولا يمكننا أتمتتها إلا إذا وجدنا خوارزميةً تجريبيةً جيدة.

الأبحاث المبكرة حول ما لا يمكن فعله في الحوسبة، سواء بسبب استحالة أو استهلاكه الكثير من الوقت، أدّت إلى تشاؤم حول ما إذا كان من الممكن أن تساعد الحوسبة في حلّ معظم المشاكل العملية.¹⁰ ولكن اليومَ تغيرت الأمور إلى الأفضل. يستخدم من يستعينون بالتفكير الحوسبي فهماً متطوراً لتعقيد الحوسبة، والمنطق، وأساليب التحسين لتصميم خوارزميات تجريبية جيدة.

على الرغم من أن جميع أجزاء الحوسبة تلعب دوراً في عملية الأتمتة، فقد برز الذكاء الاصطناعي ك مجالٍ رئيسي في الحوسبة لأتمتة المهام المعرفية لدى البشر وغيرها من الأعمال البشرية. ضمّ صندوق أدوات التفكير الحوسبي أساليب تجريبية للبحث في نطاقات حلول الألعاب، ولاستخلاص النتائج من المعلومات المُعطاة، ولأساليب تعلم الآلة التي تجد حلولاً للمشكلات من خلال وضع تعميماتٍ بناءً على الأمثلة.

(٤) الحوسبة بوصفها عمليات معلوماتية شاملة

كان انتشار الحوسبة في العديد من المجالات في تسعينيات القرن العشرين أحد العوامل الأخرى التي أسهمت في تراجع الاعتقاد السائد في الأوساط الأكاديمية بأن الكثير من المهام يمكن أتمتتها بسهولة بالاستعانة بالتفكير الحوسبي. كان من الواضح أن العلماء الذين يستخدمون الكمبيوتر في إجراء عمليات المحاكاة أو تقييم النماذج الرياضية يستخدمون التفكير الحوسبي، ولكن هدفهم الأساسي لم يكن هو أتمتة المهام البشرية. بل كانوا

يستكشفون الكون من منظور الحوسبة (انظر القسم التالي، «الكون بوصفه كمبيوتر»). دُقَّ المسمار الأخير في نعش الأتمتة عندما بدأ العلماء من مجالات أخرى حوالي عام ٢٠٠٠ يقولون إنهم تعرفوا على عمليات معلوماتية تحدث في الطبيعة دون تدخّل بشري. على سبيل المثال: قال علماء الأحياء إن العملية الطبيعية لنسخ الحمض النووي هي في الأساس عملية حوسبيّة. وتحوّل التركيز من أتمتة هذه العمليات الطبيعية إلى فهمها وتعديلها. والأمر لا يقتصر على علم الأحياء فحسب. فعلماء الإدراك يرون العديد من العمليات التي تتمّ في الدماغ عمليات حوسبيّة. كذلك يرى علماء الكيمياء العديد من العمليات الكيميائية عمليات حوسبيّة، وصمموا موادّ جديدة عن طريق حوسبة التفاعلات التي تؤدي إليها. كما تستخدم شركات الأدوية المحاكاة والبحث بدلاً من التجارب المعملية المملّة لاكتشاف مرُكّبات جديدة لعلاج الأمراض. ويرى علماء الفيزياء أن ميكانيكا الكم هي وسيلة لتفسير جميع الجسيمات والقوى كعمليات معلوماتية. والقائمة تطول. علاوة على ذلك، فإن العديد من الابتكارات الجديدة مثل: المدوّنات، والتعرّف على الصور، والتشفير، وتعلّم الآلة، ومعالجة اللغات الطبيعية، وسلاسل الكتل، هي كلها ابتكارات أصبحت ممكنة بفضل الحوسبة. ولكن لم يكن أيّ مما سبق أتمتة لعملية موجودة؛ بل أنشأ كلّ منها عملية جديدة تماماً.

يا له من تغيير جذري مقارنةً بأيام نيوويل، وبرليس، وسايمون! حينها تعرضت فكرة علوم الكمبيوتر نفسها للهجوم؛ لأنها لا تدرس العمليات والظواهر الطبيعية. أما الآن، فقد أصبحت علوم الكمبيوتر ضرورية لفهم هذه العمليات الطبيعية.

تراجع الاعتقاد السائد بأن علوم الكمبيوتر هي المجال الذي يدرس عملية الأتمتة في مُستهل القرن الجديد. ودُقَّ المسمار الأخير في نعش الأتمتة عندما بدأ العلماء من مجالات أخرى يقولون إنهم تعرفوا على عمليات معلوماتية تحدث في الطبيعة.

(٥) الكون بوصفه جهاز كمبيوتر

يقول بعض الباحثين إننا على شفا فهم جديد للكون، يتمثل في فكرة أن الكون نفسه عبارة عن كمبيوتر عملاق. ووفقاً لهذه النظرية، فكلّ ما نعتقد أننا نراه، وكل ما نُفكر فيه، هو نتيجة لعمليات حوسبيّة داخل هذا الكمبيوتر الكوني. وبدلاً من استخدام الحوسبة

أداة لفهم الطبيعة، فإننا — حسب قولهم — سنقبل في النهاية أن كل شيء في الطبيعة هو نتاج للحوسبة. في هذه الحالة، فإن التفكير الحوسبي ليس مجرد مهارة أخرى يجب تعلّمها، بل هو السلوك الطبيعي للدماغ.

يُحب كَتَّاب سيناريوهات هوليود هذا الخط السري. ولذلك فقد استغلّوه في العديد من أفلام الخيال العلمي الشهيرة القائمة على فكرة أن كل ما نعتقد أننا نراه يُنتج لنا بواسطة محاكاة باستخدام الكمبيوتر، وفي الواقع فإن كل فكرة نُفكر فيها هي وهم أنتجته لنا عملية حوسبة. قد تكون قصة جذّابة، ولكن هناك القليل من الأدلة لدعّمها.

هذا الادعاء هو تعميم لفكرة مألوفة في الذكاء الاصطناعي. يشير «الذكاء الاصطناعي القوي» إلى الاعتقاد بأن الآلات المبرمجة بشكل مناسب يمكن أن تكون ذكية بكل ما تحمل الكلمة من معنى. أما «الذكاء الاصطناعي الضعيف»، فيشير إلى الاعتقاد بأنه من خلال البرمجة الذكية، يُمكن للآلات محاكاة الأنشطة العقلية بشكل جيد بحيث تبدو ذكية دون أن تكون ذكية. على سبيل المثال: تطبيقا المساعد الافتراضي «سيري» و«أليكسا» ينتميان إلى الذكاء الاصطناعي الضعيف؛ لأنهما يقومان بعمل جيد في التعرف على الأوامر الشائعة وتلبيتها دون «فهمها».

استحوذ السعي وراء الذكاء الاصطناعي القوي على أجندة الذكاء الاصطناعي منذ تأسيس مجاله في عام ١٩٥٠ حتى أواخر التسعينيات. ولكن هذا لم يُثمر عن تقدّم يُذكر، ولم تُصمّم أي آلة يمكن اعتبارها ذكية بما يتماثل مع ذكاء البشر. ومن ثمّ أصبح السعي وراء تطبيقات الذكاء الاصطناعي الضعيف المُتخصّصة في الصدارة بدءاً من تسعينيات القرن العشرين، وكان مسؤولاً عن الابتكارات المذهلة في الشبكات العصبية وتحليل البيانات الضخمة.

وعلى غرار تمييز الذكاء الاصطناعي إلى ضعيف وقوي، فإن رؤية الكون باعتباره كمبيوتر «قوياً»، تفترض أن الكون نفسه، بالإضافة إلى كل الكائنات الحية، عبارة عن كمبيوتر رقمي. وكل بُعد من أبعاد الزمكان مُنفصل، وكل حركة للمادة أو الطاقة هي عملية حوسبية. على النقيض من ذلك، فإن رؤية الكون باعتباره كمبيوتر «ضعيفاً» لا تزعم أن العالم يُجري عمليات حوسبة، وإنما ترى أن التفسيرات الحوسبية للعالم مُفيدة جداً في دراسة الظواهر؛ إذ يُمكننا نمذجة العالم ومحاكاته ودراسته باستخدام الحوسبة. تُعد رؤية العالم باعتباره كمبيوتر قوياً رؤية تخيلية للغاية، وعلى الرغم من أن هناك مَنْ يدعمونها بحماس، فإنها تواجه العديد من المشاكل التجريبية والفلسفية. ويمكن

تفهُم ظهور هذه الرؤية على أنه استمرار للسعي المُستمر نحو فهم العالم من خلال أحدث التقنيات المتاحة. على سبيل المثال: في عصر التنوير قُورن العالم بالساعة. وقورن الدماغ بالطاحونة، ونظام التلغراف، والنظم الهيدروليكية، والنظم الكهرومغناطيسية، والكمبيوتر على التوالي. وكانت أحدث مرحلة في هذا التطور هي تفسير العالم لا باعتباره كمبيوتر تقليدياً وإنما باعتباره كمبيوتر كميّاً.

الفصل الخامس

هندسة البرمجيات

هندسة البرمجيات هي الجزء الشديد الصعوبة من علوم الكمبيوتر بالنسبة إلى عالم الكمبيوتر.

فريتز باور (١٩٧١)

في الساعة التاسعة وعشر دقائق مساءً من يوم ٢٢ يوليو ١٩٦٢، تم تجهيز صاروخ «مارينر ١» الذي يبلغ طوله ٣٣ مترًا والذي كان يقف على منصة الإطلاق في كيب كانافيرال. وكان يُوجد فوق الصاروخ إطار من المغنيسيوم سداسي الشكل يحتوي على نظام إلكتروني عالي التقنية لجمع وتحليل وحوسبة وتبادل البيانات العلمية، ونظام تشغيل للحفاظ على جميع العمليات الحيوية على متن الصاروخ. وكان صاروخ «مارينر ١»، المتجه إلى كوكب الزهرة، هو الأول في سلسلة من ١٠ مسابير بين كوكبية تابعة إلى ناسا لإجراء عمليات استكشاف للمريخ وعطارد والزهرة أثناء التحليق بالقرب منها. وكانت هذه أول مرة في التاريخ يتم فيها التحليق بالقرب من كوكب آخر. وتعد هذه اللحظة هي ثمرة جهود آلاف الأشخاص في عمليات التخطيط والحوسبة وتصميم المركبة الفضائية واختبارها ثم بنائها. كان «مارينر ١» يهدف أيضًا إلى إثبات تفوق الولايات المتحدة على الاتحاد السوفييتي في سباق الفضاء. وبعد عشر دقائق والعشرات من عمليات التحقق، أعطى نظام التحكم في الطيران الضوء الأخضر للعد التنازلي للإطلاق النهائي.

بعد ثوانٍ من إطلاق الصاروخ نحو عالم جديد، بدأت معدّات المراقبة في الإشارة إلى وجود مشكلات. فنظام تتبّع بيانات السرعة وإرسالها لم يعمل في الصاروخ بشكل صحيح. وكان من المفترض أن أجهزة الكمبيوتر الخاصة بالتحكم الأرضي تتولّى الأمر؛ وهو أمر طبيعي في العادة؛ لأن هذا هو الغرض من النظم الاحتياطية. ولكن حدث خطأ

ما أثناء تطوير نظام الكمبيوتر الذي استغرق وقتًا طويلًا، وأغفل شخصٌ ما علامة ترقيم صغيرة في البرنامج، مما جعل الكمبيوتر يبنّي قراراته على البيانات الأولية بدلاً من البيانات التي تمّت معالجتها على مدار فترةٍ زمنيةٍ مُعينة. أدى هذا الخطأ إلى تعويض الصاروخ بشكلٍ مُفرطٍ عن الاضطرابات الطفيفة في مساره، مما أدى إلى توجيهه على نحوٍ غير قابلٍ للسيطرة نحو مناطق مأهولة وممرّات شحن. بعد ٢٩٣ ثانية من الإطلاق، لم يكن أمام التحكم الأرضي خيار سوى إرسال أمر تدمير إلى المركبة. وسقطت أطنان من المعدن والإلكترونيات العالية التقنية ووقود الصواريخ في المحيط الأطلنطي.

في خلال أسبوعٍ واحد، صدرت التقارير الأولية حول أسباب الفشل المدوّي الذي أُعلن عنه على نطاقٍ واسع، وكان معظمها يذكر ذلك الخطأ الصغير في برنامج الكمبيوتر. كان العنوان الرئيسي في صحيفة «نيويورك تايمز» يقول: «صاروخ الزهرة يضع بسبب شرّطة». أدخلت تلك اللحظة الفاصلة مفهوم خطأ البرمجة إلى الوعي العام. وفتحت عيون الكثير من الناس على العواقب الكارثية المحتملة لفشل البرمجيات. وبحلول نهاية الستينيات من القرن العشرين، كانت تقارير مشكلات البرمجة أمراً شائعاً ومألوفاً، وأثرت أخطاء البرامج في الموثوقية، وقوّضت الإنتاجية، وأدّت أحياناً إلى مخاطر كبيرة.

أدرك مطوّرو البرمجيات أن التفكير الحوسبي في ذلك العصر لم يكن قادراً على تقديم برامج موثوقٍ بها يمكن الاعتماد عليها. كان معظم التفكير الحوسبي يُركّز على التفكير في نطاق الأفراد؛ أي الممارسات وأدوات التفكير الخاصة بأفراد المُبرمجين. ولم يكن يركّز على نطاق المجموعات؛ أي الممارسات وأدوات التفكير الخاصة بمجموعات المُبرمجين الذين يطوّرون نظم إنتاج كبيرة ذات عمر طويل وقواعد مُستخدمين ضخمة. وسوف نستكشف في هذا الفصل التحوّل المُهم في نطاق التفكير الحوسبي والصعوبات التي نجمت عنه.

(١) أزمات البرمجيات

كانت السنوات الأولى للكمبيوتر ذي البرنامج المُخزّن انتصاراً لهندسة الكمبيوتر. وتصدّر تطوير الأجهزة العناوين الرئيسية، من «العقل الخارق الحاسوبي» إلى «آلة التفكير المذهلة». عرضت الصحافة أجهزة العقل الخارق الحاسوبية التي تزن عشرات الأطنان والتي تعمل أسرع ألف مرة من آلات الحوسبة السابقة، والأهم من ذلك، يُمكنها إجراء العمليات الحسابية أسرع آلاف المرات من أفضل علماء الرياضيات في العالم. في الماضي

كان يُحتَفَى بالرياضيات والمنطق باعتبارهما الميزة التي ميّزت البشر عن الحيوانات، والآن يمكن للآلات أن تفعل كليهما.

سرعان ما انتقل الحماس المبكر لأجهزة الكمبيوتر من «تكوين الأعداد» — كما أطلق أحد رؤاد الحوسبة على الحوسبة العلمية — إلى معالجة البيانات في الرموز التي يُمكنها تمثيل أي معلومات على الإطلاق. وضربت المجلات والصحف أمثلةً على أجهزة الكمبيوتر التي تؤدي مهامً كانت تُعد في الماضي من اختصاص البشر وحدهم: فبرمجت مجموعة منها الكمبيوتر ليلعب الداما، وأخرى برمجت ليلعب الشطرنج، وأخرى برمجت لإثبات المبرهنات الموجودة في كتاب «مبادئ الرياضيات»، وأخرى لإنشاء فأر آلي يجد طريقه عبر متاهة. وتضاعفت استخدامات أجهزة الكمبيوتر في مجالات التجارة والعلوم والتطبيقات الهندسية كل عام. وجاءت كل هذه التطورات من البرمجيات. لقد بدأت ثورة الكمبيوتر بالأجهزة، لكنها سرعان ما أصبحت ثورة برمجيات.

ازداد حجم برامج الكمبيوتر وتعقيدها سريعاً. وبدأت السحب الداكنة تلوح في الأفق في سماء تطوير البرامج. وأصبح المطورون يُدركون بوضوح الصعوبات الكبيرة التي تُواجههم في إنشاء برامج عالية الجودة؛ أو بالأحرى برامج موثوق بها يمكن الاعتماد عليها وقابلة للاستخدام وآمنة. ولم تكن الأدوات الفكرية والإدارية المُطورة حتى ذلك الوقت قوية بما يكفي لإنشاء مثل هذه البرامج. ومن ثمّ بدأ المطورون يتحدثون عن «أزمة البرمجيات».

في اجتماعين شهيرين تحت رعاية منظمة حلف شمال الأطلسي (الناتو) في عامي ١٩٦٨ و١٩٦٩، لجأ مطورو البرمجيات إلى الهندسة بحثاً عن حلٍّ للمشكلات المُتكررة وأحياناً الكارثية للبرمجيات. وأطلق على هذه الحركة اسم «هندسة البرمجيات». كان لدى الهندسة تاريخ طويل من إنتاج نظمٍ موثوق بها طوال الوقت. فقد كان من النادر سقوط الجسور أو تحطم الطائرات أو انهيار البنية التحتية على نطاقٍ واسع. ومن ثمّ، سرعان ما بدأ مهندسو البرامج في جلب أساليب الهندسة وتطويرها في تطوير البرمجيات وإدارتها. كان من أوائل الأشياء التي ركزت عليها هندسة البرمجيات تصميم «تجريدات»، وهي عبارة عن نماذج مبسطة لنظم معقدة لها واجهة بسيطة. تخفي التجريدات الجيدة تفاصيل الآلات التي تنفذها، مما يسمح للمبرمجين بإصلاح الأخطاء دون الحاجة إلى الخوض في تفاصيل الآلات الأساسية. على سبيل المثال: الملف هو عبارة عن تجريدية تُسهّل العملية المعقدة لتخزين البيانات على القرص الصلب (فهو عبارة عن حاوية لسلسلة من

وحدات البت لها وظيفتان رئيسيتان: القراءة والكتابة، وطريقة عمله المعقدة كسجلات موزعة على القرص الصلب خافية تمامًا). ويعتبر تصميم هياكل هرمية للتجريدات هو الطريقة الوحيدة للتعامل مع التعقيد الهائل للبرمجيات. ويُعدُّ العثور على تجريدات جيدة مهارة تصميم أساسية لدى المبرمجين ومُهندسي البرمجيات. كما أن لغات البرمجة التي تسمح للمبرمجين بالتعبير عن تجريداتهم لا غنى عنها.¹

ذكر رائد البرمجيات فريد بروكس في كتابه الكلاسيكي «خرافة شهر العمل» (١٩٧٥)، بُعدين لتحويل البرامج إلى نظم إنتاج. الأول: هو تعميم برنامج حاسوبي واحد على نظامٍ من البرامج المتفاعلة. والثاني: هو إضافة الهياكل والمكونات التي تضمن موثوقية البرامج. وكانت قاعدته العامة هي أن الحركة في أي بُعد من هذين البُعدين تُضاعف الجهد المبذول ثلاث مرات. ويجب أن نتحرك في كلا البُعدين معًا لتحقيق نظم إنتاج موثوق بها؛ وهو ما يفوق جهود إنشاء برنامجٍ واحدٍ بمقدار تسع مرات.

أصبح مطوِّرو البرمجيات، بعد أن صاروا على درايةٍ بالفجوة الواسعة بين البرمجة الأساسية ونظم الإنتاج، مُضطرِّين إلى إيجاد ممارساتٍ جديدةٍ للتفكير الحوسبي لسدِّ هذه الفجوة. فطوَّروا مجموعة من الأشكال الجديدة من التفكير الحوسبي: ممارسات جديدة للتفكيك، والتعقيد، وهياكل المعلومات، والسببية، وسدِّ الفجوات الدلالية، وتجريد البيانات، وهياكل البيانات، والتضمين، وإخفاء المعلومات، والتكرار، وإدارة المشروعات، ودورة حياة البرمجيات. وأصبحت الجوانب النظرية من علوم الكمبيوتر، ولا سيما نظرية التعقيد وإثبات النظرية آلياً، مُفيدةً في هذا المجال.

يمكن وصف التحول الذي تحدَّث عنه بروكس بأنه تحول من التفكير الحوسبي على نطاق الأفراد (تصميم وكتابة برامج فردية) إلى التفكير الحوسبي على نطاق المجموعات (تصميم نظم البرمجيات وإدارة مشروعات البرمجيات التي تنشئها بداية من التصميم ووصولاً إلى الإنتاج والصيانة).

(٢) العلم والهندسة في الحوسبة

بزغت ثورة علمية في منتصف القرن الخامس عشر. ومنذ ذلك الحين، كان هناك القليل من أوجه الاختلاف العملي بين العلم والهندسة؛ إذ يبحث العلماء عن مبادئ الظواهر، وينشئ المهندسون تقنياتٍ لاستغلال هذه الظواهر. وكان العديد من العلماء مهندسين والعديد من المهندسين علماء. أما التفرقة الواضحة التي نراها اليوم بين العلم والهندسة

فهي حديثة.² بدأت هذه التفرقة في أواخر الأربعينيات من القرن العشرين عندما دعا فانيفار بوش إلى إنشاء المؤسسة الوطنية للعلوم بالولايات المتحدة لتوفير الدعم الحكومي للبحث الأساسي. ومنذ ذلك الوقت، بدأت البرامج الأكاديمية في تعريف الهندسة بأنها «تطبيق العلوم والرياضيات لحلّ المسائل المفيدة للناس»؛ مما يعني تعريف الهندسة كمجموعة فرعية من العلوم. ويُخفي هذا التعريف المساهمات الفريدة التي يمكن أن تُقدمها الهندسة إلى مجال البرمجيات. كما يحجب الحاجة إلى التفاعل بين الجانبين العلمي والهندسي للحوسبة لضمان موثوقية البرامج.

وقد وجدنا ثلاثة فروق بين الهندسة والعلم مفيدة للغاية في فهم المساهمات التي يمكن أن يُقدّمها كل منهما في مجال تطوير البرمجيات. الأول يتعلق بطبيعة عملهما. إذ يتولى المهندسون تصميم التقنيات التي تُحقق أغراضاً مفيدة وتطويرها، بينما يبحث العلماء عن قوانين لتفسير الظواهر وتوقعها. ويعطي المهندسون الأولوية إلى التصميم، على عكس العلماء، ولذا تُعد كلمة «التصميم» من بين أكثر الكلمات شيوعاً في الهندسة، بينما لا يشيع استخدامها في العلوم. والتصميم في الهندسة هو عملية إيجاد أدوات عملية وآمنة وغير مُكلفة. ويركز العلماء على البحث عن الأنماط المُتكررة والتحقّق منها، بينما يركز المهندسون على الاستماع إلى العملاء واقتراح التقنيات القيّمة لهم.

الفرق الرئيسي الثاني هو نظرة العلماء والمهندسين إلى المعرفة. يُعامل العلماء المعرفة على أنها بيانات ومعلومات منظمة في «دليل معرفي»، والذي يكون متاحاً للاستخدام بعد ذلك من قبل أي شخص. الطريقة العلمية لإنشاء المعرفة هي عملية يجمع فيها المُراقبون النُموذجيون النزهاء الأدلة التي تدعم المزاعم التي يمكن إضافتها إلى الدليل المعرفي، ويُقيّمونها. أما المهندسون فهم يُعاملون المعرفة على أنها ممارسات ماهرة تمكّنهم من تصميم الأدوات والتقنيات وإنشائها. والمهندسون ليسوا مُراقبين خارجيين؛ إنهم منغمسون في مجتمعات المُستخدمين. وهم يُجسدون الممارسات لإنشاء التقنيات وصيانتها وإصلاحها؛ والاهتمام بالموثوقية والاعتمادية والسلامة في سياق الاستخدام؛ واتباع المعايير الهندسية وقواعد الأخلاقيات.

الفرق الرئيسي الثالث يتعلق بدور التجريديات والنماذج. فالعلم يُركّز على النماذج، والهندسة تهتم بالآلات والأدوات. وهناك فرق جوهري بين وضع نماذج للآلات وإنشائها. التجريديات مفيدة؛ لأنها تبسط النظم المُعقدة. أما الآلات فهي مفيدة؛ لأنها تركز على التفاصيل العملية. إن واضع النظريات يمكن أن يَستخدِم مُصطلحي الأجهزة والبرامج بالتبادل، ولكن المهندس لا يمكن أن يفعل ذلك على الإطلاق.

العبرة المألوفة «الشیطان یکن فی التفصیل» هی شعار المهندس. إذ یجب علی المهندس الاهتمام بكل التفصیل لكي یعمل النظام الذی یُنشئه. أما العالم، فهو یرید التخلُّص من التفصیل حتی تبرز الأنماط المتكررة.

تفسر هذه الفروق سبب صعوبة تصمیم تعلیم خاص بهندسة البرمجیات، یمكنه بالفعل تخریج مطوري برمجیات أكفاء. توجد العدید من مجموعات هندسة البرمجیات فی أقسام علوم الكمبيوتر الذی تُبرز أهمية العلم علی الهندسة. ویواجه التفكير الحوسبي المشكلة نفسها فی الموازنة بین المجالین: عندما یُهیمن أحد المنظورین، یضیع التضافر بینهما.

(٣) التفكير الحوسبي علی نطاق الأفراد

كتب أحد رواد الحوسبة، الذی كان یستخدم واحدًا من أوائل أجهزة الكمبيوتر، فی مذكراته أنه لا یزال یتذكر الذی أدرك فیهِ فجأة أنه سیقضي معظم حیاته الباقية فی البحث عن الأخطاء فی برامجه الخاصة.^٣ فی خمسينیات القرن العشرين، كان الجمیع یعتقدون ذلک؛ إذ كانت كتابة البرامج الذی تعمل بشكلٍ صحیح صعبةً للغاية. كانت البرمجة أرضًا غیر مُستكشفة للجمیع. فی البداية، كان كل ما یستطیع المبرمجون الأوائل عمله هو اقتراض الأفكار والتقنیات من مجالات أخرى وبذل كل ما یملكون من براعة لجعل البرامج تعمل. ولم یکن ثمة ما یُساعدهم فی تجنب ارتكاب الأخطاء أثناء البرمجة. وكان ما یظنونهُ فی الماضي ترجمة مباشرة للخطط الخوارزمية العالیة المستوى إلی تعلیماتٍ تستطیع الآلة فهمها، عبارة عن مجموعة مُركبة من التحدیات، بدايةً من عدم اكتمال مواصفات المشكلة، وامتلاك الآلة لخصائص مميزة فريدة، وضعف الأداء، ومحدودية الذاكرة، وحتى إصلاح الأخطاء. وتبین لهم أن جعل أجهزة الكمبيوتر تعمل أمرٌ ینطوي علی حلقةٍ لا تنتهی من التکیف مع المفاجآت والعقبات.

نتیجة لذلك، أصبحت البرمجة فی خمسينیات القرن العشرين محاطةً بهالة من الغموض. وذكر رواد لغات البرمجة تلك الهالة بوضوح فی مذكراتهم. فكتب أحدهم أن البرمجة فی الخمسينیات كانت «فناً أسود، وسراً غامضاً یقتصر علی المبرمج والمشكلة وجهاز كمبيوتر، وربما مجموعة صغيرة من الروتینات الفرعية ... وبدأ المبرمجون ینظرون إلی أنفسهم كأعضاء فی طائفة كهنوتية تحرس المهارات والألغاز المُعقدة للغاية بالنسبة إلی البشر الفانین العادیین.» ووصف آخرٌ لاحقاً کیف أحب مبرمجو الخمسينیات تعلیماتهم

البرمجية وحيلهم الغامضة.⁴ في حين كتب آخر أن البرمجة لم تبدأ في التطور من حرفة إلى علم حتى الستينيات. وتعجب من أن مبرمجي الخمسينيات، على الرغم من طريقتهم «البُدايية» في التفكير في البرمجة، فقد تمكنوا من إنشاء العديد من البرامج المفيدة.⁵ كان التفكير الحوسبي في عصر الكمبيوتر المبكر غنياً ولكنه مُتَشَطٌّ، وركز على جعل البرامج الفردية تعمل على آلاتٍ مُحددة.

عمل العديد من رؤاد الحوسبة على تسهيل عمل المبرمج وتقليل الأخطاء التي يقع فيها. وقاموا بذلك من خلال تطوير وتعديل منهجية البرمجة ولغات البرمجة، وتصميم نظم تشغيل متطورة. وبدأت ابتكاراتهم بمبادئ هيكليّة للوحدات النمطية في آلات الخمسينيات من القرن العشرين، مما جعل المُفكرين الحوسبيين يُفكرون بشكلٍ متزايد في الروتينات الفرعية، ووحدات الماكرو التي تختصر أجزاء الكود المُستخدمة بشكلٍ مُتكرر، والوحدات النمطية المحولة برمجيّاً بشكلٍ منفصل، وبرامج الربط التي تضع الوحدات النمطية المحولة برمجيّاً المُترجمة في برامج كاملة، ومكتبات الوحدات النمطية الجاهزة القابلة للتنفيذ، ونظم التحكم في الإصدارات التي تتبع جميع الوحدات النمطية للبرامج التي يُنشئها فريق من المبرمجين ويُعدلونها. ساعدت جميع هذه الأدوات في إدارة تعقيدات البرامج وتقليل أخطائها.

مع تزايد معرفة مصممي اللغات البرمجية بممارسات البرمجة، طوّروا لغات برمجة عالية المستوى، مثل «فورتران» و«كوبول» حوالي عام ١٩٥٨. سمحت هذه اللغات للمُبرمجين بصياغة عبارات الخوارزمية التي يتم ترجمتها تلقائياً بواسطة برنامج التحويل البرمجي إلى تعليمات آلة؛ مما خفّف عن المُبرمجين عبء البرمجة المباشرة لتعليمات الآلة. وعندما رأى مُصمّمو اللغات أن المُبرمجين غالباً ما يبدؤون بتصميم هياكل البيانات ثم مجموعة صغيرة من الروتينات الفرعية التي تقوم بعمليات على الهياكل، أعلنوا ممارسة «تجريد البيانات». ومع الوقت، تطوّر تجريد البيانات إلى لغات البرمجة الموجهة للكائنات. وأصبح تجريد البيانات سمةً رئيسية أخرى من سمات التفكير الحوسبي؛ فهو يخفي الآليات الداخلية لمكوّنات البرنامج، بينما يسمح باستخدام هذه المكوّنات من خلال واجهات مُعرّفة جيّداً. ويمكن للمبرمجين، بالاستعانة بتجريد البيانات، أن يركّزوا بسهولة أكبر على ما تفعله المكوّنات بدلاً من التركيز على آلية عمل هذه المكوّنات.

ساهم مصمّمو نظم التشغيل بمجموعة كبيرة من المبادئ المهمة في التفكير الحوسبي خلال تلك الحقبة نفسها. تسمح نظم التشغيل لمُستخدمين مُتعددين بمشاركة جهاز واحد

من خلال جدولة الموارد، وحل التعارضات، وتخصيص الذاكرة بين برامج المستخدمين، وإرسال مهام الحوسبة المتعددة إلى المعالجات. وقَدِّم مصمِّمو نظم التشغيل فكرة النظام باعتباره «مجتمعًا من العمليات المتعاونة»، حيث إن العملية هي برنامج مُستقل التنفيذ في ذاكرة خاصة لا يُمكن الوصول إليها بواسطة عمليات أخرى، وحيث تنتظر كل عملية لأداء خدمة مُحددة عند الطلب. ابتكر مُصممو نظم التشغيل الذاكرة الافتراضية لأتمتة عمليات نقل البيانات بين مستويات الذاكرة، ونُظم الملفات لتخزين بيانات المستخدم وحمايتها، ونظم الرسائل بين العمليات لتبادل البيانات والطلبات. واخترعوا النواة لتوفير مجموعة من البرامج المهنية والموثوق بها للغاية لجميع وظائف نظام التشغيل الأساسية. تعزل النواة العمليات وتمنع أن يمتد تأثير الأخطاء التي تقع في أي عملية إلى عملية أخرى.

يرث التفكير الحوسبي في الوقت الحالي الكثير من المبادئ المتعلقة بمنهجية البرمجة، بما في ذلك التجزئة إلى وحدات نمطية، والتجريد، وإخفاء المعلومات، والتكوين الهرمي، والتكرار وأنماط التصميم، وإدارة الكائنات الرقمية، والتصوُّر، والتحقُّق، وإصلاح الأخطاء. وتتطلَّب هذه الأدوات المفاهيمية مهارةً وخبرة كبيرة في التصميم. وقد ظهر التصميم كأحد مجالات التطوير الرئيسية في الحوسبة؛ وستنحَدِّث عنه بالتفصيل في الفصل السادس. تساعد مبادئ التفكير الحوسبي المتعلقة باللُّغات والمنهجية ونظم التشغيل جميعًا في زيادة الإنتاجية والحدُّ من الأخطاء أو القضاء عليها كليًا. أصبح العديد من هذه الممارسات راسخًا جدًّا في التفكير الحوسبي لدرجة أن مُستخدمي التفكير الحوسبي في حلِّ المشاكل اعتبروها من أساسياته. وقد أكملت هذه التطوُّرات الهندسية الجانب الرياضي للبرمجة، والذي ركَّز في تلك الأيام على هيكلة البرامج لتسهيل إثبات صحتِّها شكليًّا، كما ركَّز على ممارساتٍ معينة مثل استخدام التكرار.

(٤) أزمة تطوير البرمجيات

لماذا ظهرت أزمة البرمجيات مع كلِّ هذه التطوُّرات في التفكير الحوسبي؟ في حوسبة الخمسينيات من القرن العشرين، كانت الآلة هي المُنتج. ولم تكن البرمجيات — مثل برامج التحكم في الآلات — شيئًا يُعَبَّأ ويُباع. وركز معظم المُبرمجين على برامج لاستخدامهم الشخصي أو لاستخدام مجموعات العمل القريبة جدًّا منهم، ولكنهم لم يهتمُّوا بإنشاء برامج للاستخدام خارج مؤسَّساتهم. ومن ثَمَّ، أدَّت أدوات التفكير الحوسبي الخاصة

بـ «البرمجة على نطاق الأفراد» إلى دعم الاستخدام الشخصي جيدًا، ولكنها لم تدعم التطوير على نطاق واسع للمنتجات البرمجية المعقدة.

بدأت صناعة البرمجيات في التطور من عددٍ قليل من مقاولي البرمجيات في الخمسينيات إلى مُطوِّري البرمجيات المؤسسية في الستينيات، ثم إلى البرمجيات الجاهزة للبيع في السبعينيات وما بعدها. وفي كل عَقْدٍ من هذه العقود، نمت إيرادات صناعة البرمجيات عشرة أضعاف.

في ستينيات القرن العشرين، اكتشف مُطورو البرامج أن بيع البرمجيات ليس بالأمر السهل. إذ كان الكثير من مشروعات البرمجيات إما يتأخَّر تنفيذه، أو يتجاوز الميزانية المحددة له، أو يكون مليئًا بالثغرات البرمجية والأخطاء لدرجة عدم جدواه، أو لا يتم تسليمه على الإطلاق. كذلك كانت صيانة البرامج وتحسينها وإصلاح الأخطاء الموجودة فيها بعد التسليم أمرًا مُكلفًا وصعبًا، وفي بعض الأحيان غير مُجدٍ. واحتوت نظم البرمجيات في كثيرٍ من الأحيان على أخطاء برمجية كامنة جعلت تطبيقاتها غير آمنة للبشر أو تسببت في أعطالٍ باهظة الثمن مثل تدمير مركبة الفضاء «مارينر».

غالبًا ما تسبَّب مطورو البرامج الذين لديهم معرفة قليلة بمجال العمل المُستهدف في فجواتٍ كبيرة بين احتياجات العملاء ووظائف نظم الكمبيوتر. ومن ثَمَّ، اكتشف مطورو البرامج أن مبادئ التصميم المعروفة لم تكن كافيةً لتوفير برامج موثوق بها وأمنة ويمكن الاعتماد عليها وسهلة الاستخدام. وأدرك المبرمجون المحترفون أن مهاراتهم في التفكير الحوسبي لم تصل إلى المستوى المطلوب؛ إذ إن هناك اختلافًا نوعيًا بين برنامج كتبه مبرمج واحد ونظام يتطلَّب فريقًا من ٣٠٠ مبرمج.

حاولت شركات البرمجيات تقليل هذه المشكلات بطريقتين. الأولى: هي توظيف مُبرمجين ذوي مهارات كبيرة، يُمكنهم إنتاج كمياتٍ أكبر بكثير من التعليمات البرمجية يوميًا، بأقل عدد من الأخطاء مقارنة بالمُبرمجين المبتدئين. وارتفعت رواتب المبرمجين المهرة، وأصبح تطوير البرامج من أعلى المهن أجرًا في الولايات المتحدة.

أما الطريقة الثانية: فهي التخلي عن مسئولية الأخطاء. تبنت شركات البرمجيات سياسة «عدم الضمان»؛ حيث يُرَخَّص البرنامج للمستخدم فقط بعد موافقته على أن الشركة لن تتحمَّل المسئولية عن الأضرار الناجمة عن الأخطاء في التعليمات البرمجية. ساهمت هذه السياسة بشكلٍ كبير في إصابة الجمهور بخيبة أمل في ثورة الكمبيوتر.

اعترف مُطَوِّرو البرامج الرائدون أن أدوات البرمجة على نطاق الأفراد لم تكن ببساطة كافية للبرمجة على نطاق المجموعات. فقد وصلوا إلى مرحلة أصبح فيها من الصعب إنشاء برامج موثوق بها بالأدوات المتوفرة. وأعلن عددٌ من الشخصيات الرائدة في صناعة البرمجيات والأكاديميين ومُطوري البرامج عن أزمةٍ في مجال البرمجيات، ونظّموا مؤتمريّن تحت رعاية منظمة حلف شمال الأطلسي (الناٲو) في عامي ١٩٦٨ و١٩٦٩ لحلّ هذه الأزمة.

(٥) التفكير الحوسبي على نطاق المجموعات

ماذا يحدث عندما ننقل من برامج فردية ذات مُستخدمين فرديّين إلى نظم مكوّنة من العديد من البرامج التي يستخدمها العديد من المُستخدمين؟ المهارات والقدرات المطلوبة لكتابة برنامج مكوّن من ألف سطر من التعليمات البرمجية تختلف عن تلك اللازمة لإنشاء برنامج يتكوّن من مليون سطر. والسبب الرئيسي وراء هذا الاختلاف هو أن نظم البرمجيات الكبيرة يجب إنشاؤها من قبل فرق عمل. وكان على مطوِّري البرامج تعلم كيفية تنظيم الفرق وإدارتها للنجاح في تطوير البرامج.

كان فريد بروكس مدير فريق مؤلّف من ٣٠٠ مبرمج أنشأوا نظام التشغيل «آي بي إم ٣٦٠» في الستينيّات من القرن العشرين. وأصبح نظامهم في النهاية مكوّنًا من ١٠ ملايين سطر من التعليمات البرمجية. وقد وثّق بروكس تجربته بالتفصيل في كتابه «خرافة شهر العمل» (١٩٧٥) وأعطى روتينًا قائمًا على التجربة العملية للتفكير الحوسبي لتصميم النظم الكبيرة وتنظيمها. إحدى ملاحظاته الشهيرة هي أن زيادة عدد الأشخاص لا يعني بالضرورة تقليل الوقت بالنسبة نفسها؛ فالفريق المكوّن من ١٢ مبرمجًا لا يستطيع في شهرٍ واحد إكمال وظيفة استغرقت من مبرمج واحد ١٢ شهرًا لتنفيذها. ملاحظة أخرى هي أن هيكل البرنامج يتماثل في النهاية مع الهيكل التنظيمي للمؤسسة التي أنشأته. وخلص بروكس إلى أن إدارة الفريق كانت تحديًا أكبر من مشكلات التقنية التي يتعيّن على الفريق حلّها.

على الرغم من اتفاق الحضور في مؤتمريّ الناٲو على وجود «مشكلة برمجيات» كبيرة وأن مبادئ الهندسة قد تساعد في حلّ هذه المشكلة، فإنهم لم يتفقوا على نوع الهندسة الذي سيؤدي المهمة. اعتقد المهندسون التقليديون أن الحل هو التصميم القائم على تحمّل الأخطاء، وفكر المنظومة، وإدارة المشروعات. أما علماء الكمبيوتر المُهتمون

بالنظريات، فتطلَّعوا إلى البرهان الرياضي (التحقُّق الشكلي) لتحديد ما إذا كانت البرامج تفي بمواصفاتها دون أخطاء، وقَدِّموا أساليب مثل البرمجة المهيكلية وتحليل الخوارزميات لتسهيل فهم البرامج والتحقُّق منها.

لم يؤثر أيُّ من هذين النهجين بشكلٍ كبير على مشكلة البرمجيات. لم تنجح هندسة النظم التقليدية بسبب وجود فرقٍ كبير بين البرمجيات والنظم الكبيرة الملموسة، مثل الكباري والمباني والطائرات والسفن؛ فقد يؤدي خطأ في بت واحدٍ من التعليمات البرمجية إلى كارثة مُحَقَّقة مثل تحطُّم صاروخ، في حين أن فقدان جزء صغير من المادة قد يؤدي إلى الخطأ من قيمة نظام كبير ولكن لن يتسبَّب في تحطُّمه. لم ينجح البرهان الرياضي لأنه كان صعباً للغاية بالنسبة إلى النظم الكبيرة، ولم يقل شيئاً عن الجوانب البشرية مثل سهولة الاستخدام، كما لم يتعامل مع أعطال الأجهزة مثل تعطل مُكوِّن من مُكوِّنات الكمبيوتر أو الضوضاء التي تفسد الإشارات. كان رائدا البرمجيات برايان رانديل وفريد بروكس من بين الأكثر بصيرةً وبُعْداً للنظر في شرح سبب صعوبة نظم البرمجيات. قال رانديل إن المشكلة لم تكن البرمجة في حدِّ ذاتها، بل «التطوير المتعدِّد الأشخاص لبرامج مُتعددة الإصدارات». وقال بروكس في كتابه الذي صدر عام ١٩٧٥ إن مسألة تحويل البرنامج إلى مُنتَج من خلال تحويله إلى نظامٍ يمكن لغير المُبرمجين استخدامه بأمانٍ وبشكل موثوق به كانت أكثر تحدياً بكثيرٍ من كتابة البرنامج في المقام الأول.

(٦) مبادئ التصميم والأنماط والتلميحات

يمكن أن يؤدي التصميم الماهر إلى تحسيناتٍ هائلة في حجم البرمجيات وتعقيدها. وتُعَد نظم التشغيل مثلاً جيداً. تقترب نظم التشغيل الحديثة مثل Windows 10 أو MacOS X أو Linux من ١٠٠ مليون سطر من التعليمات البرمجية. يُعَد إنتاج مثل هذه النظم بموثوقية عالية انتصاراً لهندسة البرمجيات. تحتوي جميع هذه النظم على «نواة»، وهي مجموعة من وظائف البرمجيات للعمليات الأساسية جداً في النظام مثل بدء تنفيذ برنامج أو تبادل الرسائل بين البرامج أو قراءة الملفات. لم تتغيَّر وظائف النواة كثيراً منذ السبعينيات، ولكن أحجام النواة زادت زيادة هائلة من حوالي ٢٠ ألف تعليمة في النظم المبكرة إلى ٢٠ مليون تعليمة اليوم. وقد أدَّت هذه الزيادة في الحجم إلى زيادة الوقوع في

أخطاء برمجية. ويُرجع نيكولس فيرت ذلك إلى إهدار الموارد الرخيصة؛ أي دورات المعالج ووحدات البت الخاصة بالتخزين. فقد كتب:

لقد أصبح هذا الإهدار موجودًا دائمًا ويُمثل نقصًا خطيرًا في حسّ الجودة. ويمكن بسهولة إخفاء عدم كفاءة البرامج عن طريق الحصول على مُعالجات أسرع وإخفاء التصميم السيئ للبيانات باستخدام أجهزة تخزين أكبر. ولكن آثارها الجانبية هي انخفاض الجودة؛ بعبارة أخرى: الموثوقية والمتانة وسهولة الاستخدام. إن التصميم الجيد والدقيق يستغرق وقتًا طويلاً ومُكلفًا. ولكنه لا يزال أرخص من البرمجيات غير الموثوق بها وصعبة الاستخدام، عندما نضع في اعتبارنا تكلفة «الصيانة». وكما أن الوضع مُزعج، فإن لا مبالاة العملاء أيضًا مُزعجة. (فيرت ٢٠٠٨)

لُخصّت أهداف البرمجة على نطاق المجموعات في خمسة أهداف: إمكانية الاعتماد عليها، والموثوقية، وقابلية الاستخدام، والسلامة، والأمن. ولتحقيق هذه الأهداف، يستخدم مطوِّرو البرامج ثلاثة أنواع من ممارسات التفكير الحوسبي: مبادئ التصميم، والأنماط، والتلميحات.

مبادئ التصميم هي وصفٌ للمهارات والاستراتيجيات التي يتبعها المطوِّرون عند اتخاذ قرارات التصميم. توجَّههم المبادئ نحو تصميماتٍ تُلبّي الأهداف الخمسة. أنماط التصميم هي وصف للمواقف الشائعة التي من المُحتمل أن يُواجهها المبرمج. وتقدِّم توجيهاتٍ حول كيفية هيكلة البرنامج أو عملية كتابته للحصول على أفضل النتائج. تلميحات التصميم هي قواعد إرشادية أو نصائح، تكون مُفيدة للغاية لأصحاب المهارات المُتقدمة في تطوير النظم.

المبادئ

ورقة البحث الكلاسيكية التي كتبها جيروم سالتزر ومايكل شرودر حول حماية المعلومات هي مثال مُمتاز لمبادئ التصميم (انظر جدول ٥-١).⁶ ومبادئ التصميم هي طرقٌ للتفكير في النظام الكلي للمكونات البرمجية، من أجل تحقيق الأهداف الخمسة والحدّ من كشف المعلومات الحساسة أو تعريضها للخطر. وتتجسد المبادئ في المهارات وطرق التفكير التي يكتسبها مطوِّرو النظام بمرور الوقت من إنشاء نظم الحوسبة المُعقدة. وتنطبق على أي نظامٍ كبيرٍ يستوعب العديد من المُستخدمين وعمليات الخدمة.

جدول ٥-١: مبادئ حماية المعلومات لساتلزر وشرودر

المبدأ	التوجيه
الاقتصاد في الآلية	إبقاء التصميم بسيطاً وصغيراً.
الإعدادات الافتراضية لضمان السلامة عند حدوث أعطال	رفض الوصول افتراضياً؛ وعدم منح إمكانية الوصول إلا بعد الحصول على إذن صريح.
الوساطة الكاملة	فحص كل عمليات الوصول إلى كل الكائنات.
التصميم المفتوح	عدم الاعتماد على جهل المهاجمين بالتصميم.
فصل الصلاحيات	منح إمكانية الوصول بناءً على أكثر من معلومة واحدة
أقل صلاحيات	إجبار كل عملية على العمل بأقل امتيازاتٍ ضرورية لمهمتها.
أقل آلية مشتركة	جعل المعلومات المشتركة غير قابلة للوصول من قِبل العمليات الفردية لتجنب تلفها.
التقبل النفسي	جعل إعدادات الحماية سهلة الاستخدام، على الأقل بالقدر نفسه من سهولة عدم استخدامها.

الأنماط

في أوائل التسعينيات من القرن العشرين، أسست مجموعة من المبرمجين حركة «مجتمع أنماط البرمجيات»، المستوحاة من فكرة أنماط التصميم للمهندس المعماري كريستوفر ألكساندر.⁷ وكانت فكرتهم أنهم إذا تمكّنوا من وصف نمط شائع من أنماط استخدام البرمجيات تم حلّه من قِبل مبرمجين ماهرين، فيمكنهم استخلاص جوهر هذا النمط حتى يتمكن المبرمجون الآخرون من تقليده. ويصف نمط البرمجيات عددًا كبيرًا من المواقف التي من المحتمل أن يواجهها المبرمج ويُقدّم توجيهاتٍ حول كيفية هيكلة البرنامج بحيث ينطبق عليه النمط.⁸ وهناك العشرات من الأنماط المعروفة. ومنها نمط الحالات المفردة، الذي يقصر عدد حالات الكائن على حالة واحدة، والنمط التكراري الذي يطبق الوصول التسلسلي إلى عناصر البيانات. يؤكد مجتمع الأنماط على فكرة التجريبية لأن أعضائه يصرون بلا هوادة على اختبار الأفكار مع المستخدمين المحتملين والتعلّم من تجاربهم.

التلميحات

لَخَصْ باتلر لامبسون، وهو مُصمم رائع ومتميز، عددًا من الإرشادات لمُصممي نُظم التشغيل المتقدمين.⁹ قال لامبسون: «إن تصميم نظام كمبيوتر يختلف تمامًا عن تصميم خوارزمية. فالواجهة الخارجية تكون أقل دقةً في تحديدها وأكثر تعقيدًا وأكثر عرضةً للتغيير. يحتوي النظام على هيكلٍ داخلي أكبر بكثير، ومن ثَم العديد من الواجهات الداخلية. ومقياس النجاح في التصميم غير واضح.» وقال إن المُصممين الأقل مهارةً غالبًا ما يغرقون في بحرٍ من الاحتمالات، ولا يعرفون كيف سيؤثر اختيار حالي على الخيارات المُستقبلية لأداء النظام. وأطلق على عباراته «تلميحات التصميم»؛ لأنها أحكام يتعلّمها المُصممون المهرة بمرور الوقت؛ فهي تؤكد على الفن الذي ينطوي عليه التصميم. في جدول ٥-٢، ندرج تلميحات لامبسون لثلاثة أبعاد من تطوير النظام (تُمثلها الصفوف) والجوانب الرئيسية من أهداف التصميم (تُمثلها الأعمدة). وعلى الرغم من أنها قد تبدو عباراتٍ عامة، فإنها ذات مغزى كبير في تشكيل مهارات التفكير الحوسبي لدى المُصممين المتقدمين.

جدول ٥-٢: تلميحات التصميم كما حدّدها لامبسون

الصحة والملاءمة	السرعة	تحمل الأخطاء
حالات الاستخدام	افصل سيناريوهات الحالات المعتادة عن الحالات الأكثر تطرفًا.	اجعل التنفيذ عبر النظام بأكمله (من الطرف إلى الطرف).
الواجهة	احرص على أن تكون الواجهة بسيطة.	اجعل التنفيذ عبر النظام بأكمله (من الطرف إلى الطرف).
رُكّز على شيءٍ واحد واتقن تنفيذه.	قسّم الموارد.	أنشئ سجلًا بالتحديثات.

الصحة والملاءمة	السرعة	تحمل الأخطاء
ابتعد عن التعميمات، وتعامل مع الاحتياجات الفردية.	استخدم مبدأ التحليل الثابت وأجرِ عمليات التحقق في وقت تجميع البرنامج لاكتشاف الأخطاء مبكراً.	تأكد من تنفيذ كل الإجراءات معاً أو عدم تنفيذها لضمان الاتساق.
اجعل الأولوية لدقة التصميم وصحته.	ترجم التعليمات البرمجية ترجمة ديناميكية للإسراع من التنفيذ.	
اجعل الميزات الفعالة واضحة أمام المستخدمين، ولا تبالي في التبسيط. استخدم وسائط الإجراءات لزيادة المرونة. اترك الأمر للعميل، ولا تفرض أوضاعاً افتراضية. حافظ على استقرار الواجهة وتجنب التغييرات المتكررة. احتفظ بأساس ثابت لموثوقية الاستخدام وال تطوير.		
التنفيذ	توقع إعادة التصميم أو إعادة هيكلته.	خزن نتائج العمليات المهمة لتجنب تكرار عمليات الحوسبة.
أوجز في التفاصيل. اعد استخدام الحلول الجيدة متى كانت ملائمة	استخدم التلميحات استخدم الحلول البسيطة المباشرة لأنها أكثر فاعلية من الحلول المعقدة.	تأكد من تنفيذ كل الإجراءات معاً أو عدم تنفيذها لضمان الاتساق. استخدم التلميحات.
قسّم المشكلات المعقدة إلى أجزاء أبسط.	انقل المهام إلى الخلفية لضمان سرعة الاستجابة.	

الصحة والملاءمة	السرعة	تحمل الأخطاء
استخدم مبدأ المعالجة على دفعات لتقليل العبء الإضافي وزيادة الكفاءة.		

(٧) مبادئ التصميم الخاصة بالبرمجيات

تُسجَل أدبيات هندسة البرمجيات عددًا كبيرًا من مبادئ التصميم التي دُرست على نطاقٍ واسع ووجد أنها تدعم التصميم الجيد بقوة. وقد شُفرت أفضل هذه المبادئ على هيئة هياكل تظهر في لغات البرمجة وبرامج التطبيقات ونُظم التشغيل. وكثيرًا ما تُذكر هذه المبادئ في مناقشات التفكير الحوسبي وتكمن جذورها في العديد من التقاليد الفكرية المختلفة الموضحة في الفصول السابقة من هذا الكتاب. وتنقسم إلى ثلاث فئات رئيسية:

- التجميع الهرمي
- الآلات الافتراضية
- الوحدات التابعة – وحدات الخدمة

تهدف هذه الهياكل إلى المساعدة في الأنماط المتكررة التي يواجهها المصممون.

التجميع الهرمي

يعني التجميع الهرمي أن الكائنات (مكونات البرامج ومكونات الأجهزة المادية المتعارف عليها) تتكوّن من مجموعاتٍ من الكائنات الأصغر مُتصلة من خلال واجهات معرّفة جيدًا. ويُمكنك التفاعل مع الكائن على أنه وحدة واحدة من خلال واجهته وعدم الاهتمام بأجزائه الفردية. أما عندما تنظر إلى الداخل، فلن تحتاج إلى الاهتمام بما يحدث في البيئة الخارجية. وبناءً عليه، فهناك تسلسل هرمي مكوّن من تجمعات أصغر تُشكل تجمّعات أكبر. وتكون التجمّعات في كل مستوى من التسلسل الهرمي معزولةً عن التفاصيل في المستويات الأدنى والأعلى.

هناك قائمة طويلة من جوانب التجزئة إلى وحداتٍ نمطية هرمية الشكل. و«التفكيك» يعني تقسيم نظامٍ كبيرٍ إلى مكوناتٍ أصغر تسهل إدارتها. أما «التجزئة» إلى وحداتٍ نمطية، فهي عملية تنفيذ المكونات بوصفها وحداتٍ نمطية يُمكن تصميم كلٍّ منها وترجمته وتخزينه على حدة، ثم تجميعها في النظام الكامل. وتتفاعل الوحدات النمطية عبر «واجهات» معرّفة بدقة. ويمكن تخزين الوحدات النمطية في المكتبات و«إعادة استخدامها» لأغراضٍ أخرى. و«التجريد» يعني تحديد نموذجٍ مُبسّطٍ من شيءٍ ما وتحديد المعاملات (الدوال) التي تنطبق عليه. و«المستويات» هي شكل هيكلي تتشارك فيه المكونات النظرية واجهةً مشتركة.¹⁰ يؤدي «إخفاء المعلومات» إلى إخفاء تفاصيل التنفيذ عن المُستخدمين؛ مما يحمي المُستخدمين من الأخطاء الناجمة عن التغيير في التفاصيل ويحمي الوحدة النمطية من الأخطاء الناجمة عن التغييرات الخارجية. أما «التضمين» فيذهب إلى أبعد من ذلك، من خلال حماية أي شيءٍ خارج أي وحدةٍ نمطية غير موثوق فيها من الأخطاء التي تقع داخل هذه الوحدة النمطية.

مفهوم «الكائن» هو شكل مُتقدم من أشكال التضمين نشأ من إحدى ممارسات البرمجة التي أُطلق عليها «تجريد البيانات» في الستينيات من القرن العشرين، وتطوّر حالياً إلى أكثر من مائة لغةٍ متطورة موجهة للكائنات. والكائن هو عبارة عن كيانٍ مجرد لا يمكن عرضه وتعديله إلا من خلال مجموعة مُحددة من العمليات. ويُخفى هيكله الداخلي وحالته. على سبيل المثال: يظهر الملف للمُستخدمين كحاوية لسلسلةٍ من وحدات البت، ولا يمكن التعامل معه إلا من خلال عمليات الفتح والإغلاق والقراءة أو الكتابة، ويكون هيكله الداخلي المُخفي هو مجموعة من السجّلات المبعثرة عبر قرص. ولا يهم المُستخدمين هيكلُ الملف في القرص، ومن ثم فهو مَخفي عنهم. أما «فئة» الكائنات، فهي مجموعة من الكائنات ذات الواجهة الواحدة، وتُنظّم الفئات في تسلسلٍ هرمي خاص بها. وغالباً ما يبرزع المبرمجون المُبتدئون من الكائنات؛ لأنهم لا يفهمون بعدُ الآلات المُجردة وإخفاء المعلومات والمزامنة.

الآلات الافتراضية

الآلة الافتراضية هي محاكاة لجهاز كمبيوتر بواسطة جهاز كمبيوتر آخر. وقد كان جهاز الكمبيوتر المُتعدد الأغراض الذي ابتكره آلان تورينج هو أول مثالٍ على الآلة الافتراضية. واليوم يُستخدم مصطلح «الآلة الافتراضية» بعدة طرق.

أولاً: يعني مُحاكاة أي آلة حوسبة مجردة؛ أي إنها المنصّة التي يمكن تنفيذ عمليات الحوسبة عليها.

ثانياً: الآلات الافتراضية هي مُحاكاة لأجهزة الكمبيوتر المادية. وتحتوي الآلة الافتراضية على روتينات فرعية تقوم بنفس وظيفة تعليمات الآلة على جهاز الكمبيوتر المادي. وقد ظهرت هذه الفكرة عملياً في أواخر الخمسينيّات من القرن العشرين عندما بدأ جيل ثانٍ من أجهزة الكمبيوتر يحلّ محلّ الجيل الأول. وكان على أجهزة الكمبيوتر الجديدة تشغيل كلّ البرامج المكتوبة للإصدارات السابقة من جهاز الكمبيوتر. ووفقاً لذلك، قدمت الشركات المُصنّعة «وضع محاكاة» يمكن للكمبيوتر الجديد فيه محاكاة تعليمات الكمبيوتر القديم الذي حلّ محله. وقد تطوّر وضع المحاكاة في شكل «في إم وير» و«هاير في»، اللذين يُحاكيان أجهزة الكمبيوتر بأكملها ويشغلان نظم التشغيل الخاصة بها. تحاكي آلات جافا الافتراضية لغة جافا على أي آلة تجارية عن طريق تنفيذ «تعليمات البايت» المكتوبة بلغة جافا التي أنتجتها برامج تحويل لغة جافا؛ مما يسمح بقدر كبير من قابلية تشغيل برامج جافا على أجهزة مُتعدّدة.

ثالثاً: الآلات الافتراضية هي محاكاة لآلة مُضيّفة داخل أقسام ذاكرة منفصلة للآلة المُضيّفة. وهذا هو المبدأ التنظيمي لنظام التشغيل «في إم ٣٧٠» وما بعده من نظم التشغيل المطروحة من شركة «آي بي إم». وتُعدّ الآلة الافتراضية لشركة «آي بي إم» محاكاةً كاملة لجهاز الكمبيوتر الرئيسي المطروح من قبل الشركة نفسها، ومطابقة تماماً للجهاز الأصلي باستثناء أنها تحتوي على ذاكرة رئيسية أقل. ويسمح هذا النهج للآلة الافتراضية بالعمل تقريباً بنفس سرعة الآلة الحقيقية، ومن ثمّ لا يُوجد تأثير سلبي واضح على الأداء.

رابعاً: الآلة الافتراضية هي بيئة قياسية لتنفيذ أي برنامج داخل نظام تشغيل. ابتُكرت هذه الفكرة في نظام تشغيل مالتيكس في معهد ماساتشوستس للتكنولوجيا (١٩٦٨) ونظام تشغيل «يونكس» في مختبرات بل (١٩٧٢). تضمّن هذان النظامان العديد من «العمليات»، كلّ منها برنامج قيد التنفيذ على آلة افتراضية. وكانت الآلة الافتراضية ببساطة قالباً قياسياً لتوفير المُدخلات والمُخرجات لبرنامج قيد التشغيل، والاتصال بأي آلات فرعية قد تكون انبثقت عنها. وتُدْمَج كل برامج المُستخدِمين في الآلة الافتراضية القياسية ليتمّ تنفيذها.

برامج الوحدات التابعة ووحدات الخدمة

يُعتبر نموذج «الوحدة التابعة - وحدة الخدمة» طريقة بسيطة من الناحية المفاهيمية لتنظيم التفاعلات بين البرامج في نظام حوسبة موزع (مُتصل عبر الشبكة). فبرنامج وحدة الخدمة هو برنامج مُخصص لتقديم خدمة مُعينة عند الطلب. أما برنامج الوحدة التابعة، فهو برنامج آخر يطلب هذه الخدمة. عادةً (ولكن ليس دائماً) تكون الوحدات التابعة ووحدات الخدمة على أجهزة مُضيفة مختلفة مُتصلة عبر شبكة. وتمرّر الطلبات ونتائجها على شكل رسائل عبر الشبكة. على سبيل المثال: تُخزّن وحدة خدمة ملفات الشبكة جميع ملفات مستخدمي الشبكة، وترسل برامج الوحدات التابعة المثبتة على أجهزة المستخدمين طلبات لقراءة الملفات ونسخها. كما تتفاعل وحدة خدمة التوثيق مع برنامج الوحدة التابعة الموجود على جهاز المُستخدم الذي يرغب في تسجيل الدخول للتحقق من هوية المُستخدم عند تسجيل دخوله. وتتفاعل وحدة خدمة الويب مع مُتصفحات الوحدات التابعة لإرسال صفحات الويب إليها.

على الرغم من أن فكرة «الوحدة التابعة - وحدة الخدمة» بسيطة، فإن تنفيذها غالباً ما يكون بعيداً كل البعد عن البساطة. ويجب على المُصممين إتقان العديد من التفاصيل الدقيقة لتتمّ عمليات الاتصال والتحكّم في الأخطاء والمزامنة بشكلٍ صحيح.

(٨) لا توجد حلول سهلة

في عام ١٩٨٧، كتب فريدريك بروكس مقاله المُعنون «لا توجد حلول سهلة» وذكر فيه تقييمه الشهير لمدى التقدّم المُحرز في هندسة البرمجيات منذ عام ١٩٦٨. وحملت النتائج التي توصّل إليها دروساً مهمة تخصّ التفكير الحوسبي. قال بروكس إن هناك عاملين رئيسيين للتعقيد يؤثران في قدرتنا على إنتاج برامج موثوق بها.

العامل الأول: هو محدودية التكنولوجيا، ولكن يمكن التغلّب على هذا العامل باستخدام التقنيات المُحسّنة، مثل لغات البرمجة العالية المستوى، وبيئات تطوير البرامج التفاعلية، وتصوّر التحكّم وتدفّق البيانات، والأجهزة الأسرع، ونظم التشغيل الأفضل.

العامل الثاني: هو قدرتنا الذهنية على فهم جوهر المشكلات المعقدة. ويُعد التعامل مع التعقيد أمراً أساسياً في تصميم البرمجيات وهيكلتها، ولن يتغيّر أبداً. قال بروكس إن

مشكلة التصميم هي في الغالب مشكلة مفاهيمية؛ وهي الفهم العقلي لوظائف النظام من أجل توفير وتنظيم تصميم بسيط وممتاز.

لمواجهة ذلك، نحتاج إلى تطوير النظم الكبيرة على مراحل سهلة نسبياً، وإعادة استخدام البرامج الحالية قدر الإمكان، واستخدام المزيد من النماذج الأولية السريعة للحصول على تقارير مُبكرة قبل اتخاذ القرارات التقنية. وقال بروكس إن الأهم من ذلك هو أننا بحاجة إلى «إعداد مُصممين عظماء». وكان يرى أن التعامل مع التعقيد مهارة أساسية تتطلب إتقاناً كبيراً. ولهذا كتب بروكس مقاله الشهير الذي قال فيه «لا تُوجد حلول سهلة» لمشكلة التعقيد في تطوير البرمجيات.

العامل الأساسي الذي يكمن وراء إنشاء برمجيات موثوق بها هو قدرة عقولنا على فهم جوهر المشكلات المُعقدة. ويُعد التعامل مع التعقيد أمراً أساسياً في تصميم البرمجيات وهيكلتها، ولن يتغير أبداً.

كانت «مشكلة البرمجيات» التي أُثيرت في مؤتمرَي النانو تخصّ في الغالب إنتاجية المُبرمجين ووجود أخطاء في البرامج تتسبّب في عدم موثوقيتها. ومنذ تلك الأيام، أضافت التطوّرات الجديدة إلى تعقيدات تصميم البرمجيات. ومن هذه التعقيدات:

البرامج الضارّة وعمليات الاختراق: يبحث المُجرمون والمخترقون عمداً عن ثغرات في البرامج المُعقدة، ويستغلّونها لسرقة البيانات أو إتلافها، أو للمُطالبة بقدية لإلغاء تشفير البيانات المُشفرة عمداً.

تحمل الأخطاء: حتى إذا ثبتت صحة البرنامج، فإن هذا الإثبات قائم على افتراض أن الأجهزة المادية تعمل دائماً كما هو مُتوقّع منها. وفي الوقت الحالي، الأجهزة نفسها مُعقدة للغاية لدرجة أن إثبات أنها تعمل بشكل سليم يُمثل تحدياً كبيراً في حدّ ذاته. وقد اكتُشفت العديد من مشاكل الأجهزة في الشرائح التي يُفترض أنها مُختبرة جيداً. ليس ذلك فحسب، بل يمكن أن تتلف الأجهزة وتبدأ في التعطّل بسبب تعطل مكوناتها أو بسبب أحداث غير متوقّعة في العالم تؤدي بها إلى حالة من عدم الاستقرار. ولذا أصبح مهندسو الأجهزة مُهتمّين بشكل متزايد بالتصميمات التي تضمن تحمل الأخطاء؛ على سبيل المثال: النظام الذي يُخلّق نفسه بدلاً من تنفيذ عملية حرجة بشكل غير صحيح. ولا يمكن تنفيذ هذا النوع من تحمل الأخطاء في الأجهزة، والذي يتضمّن دوائر إضافية

يُراقب بعضها البعض، بأي هيكل برمجي. وإثبات صحة البرمجيات ليس كافياً لضمان التشغيل الصحيح.

تأمين الأجهزة: تحدث معظم الهجمات على نظم الكمبيوتر على أدنى مستويات النواة والشبكة، حيث يكون من الصعب للغاية إجراء مراقبة ديناميكية فعالة. وفي الستينيات من القرن العشرين، كان هناك اهتمام كبير بتصميم الأجهزة الذي من شأنه تسهيل حماية المعلومات عن طريق الحد من انتشار الأخطاء وإحباط المحاولات البرمجية لتجاوز الأدونات. ومن ثم، صُممت هياكل متقدمة للغاية للتمكن من تضمين البرامج غير الموثوق بها والحد من انتشار الأخطاء بشكل كبير.

لسوء الحظ، فقدت هذه التطورات في خضم «ثورة أجهزة الكمبيوتر ذات التعليمات المُخَفَّضة» التي اندلعت في الثمانينيات من القرن العشرين. فمن أجل إنشاء وحدات معالجة مركزية أسرع، أزال مُصمِّمو الكمبيوتر مئات التعليمات من وحدات المعالجة المركزية؛ مما قلَّل حجمها إلى شرائح بسيطة للغاية وسريعة جدًا. وأطلقوا على الجيل الجديد من الشرائح أجهزة الكمبيوتر ذات التعليمات المُخَفَّضة. وأدى تخفيض التعليمات إلى عدم إضافة مُلحقات لتضمين البرامج ومراقبتها.

يُطالب خبراء التأمين حاليًا بزيادة مراقبة الأجهزة لمنع هجمات المستوى الأدنى التي تُنفذها البرامج الضارة والمُتسلِّلون. وبالفعل، أصبح تأمين الأجهزة في غاية الأهمية في الوقت الحالي. ومن العوامل المثيرة للقلق أن العديد من الشركات يستعين بمصادر خارجية لإنتاج شرائح أجهزة الكمبيوتر الخاصة بها؛ مما يُتيح للغير إمكانية إضافة نقاط خفية للوصول إلى الأجهزة، ويسمح للمُتسلِّلين بالوصول إلى النظام بسهولة.

خوارزميات تعلُّم الآلة: يُعزى التطور الأخير المُبهر في مجال الذكاء الاصطناعي بشكل أساسي إلى النمو السريع في تقنيات الشبكات العصبية. عند الانتهاء من عملية تدريب الشبكة العصبية، لا أحد يعرف السبب في أن القِيَم المُرجَّحة للاتصالات الداخلية تكون على ما هي عليه أو كيفية إثبات صحَّة الشبكة بالنسبة إلى الإدخالات التي لم تتدرَّب عليها. وبالمثل، لا توجد عمليات مراقبة للأجهزة للكشف عن التعطُّل الوشيك للشبكة العصبية. وقد أُطلق على ذلك اسم «مشكلة الهشاشة»: إلى أي مدًى يُمكننا الوثوق بالذكاء الاصطناعي في أن يُحسِّن التصرُّف عند تعرُّضه لمدخلات خارج نطاق البيانات التي تدرَّب عليها؟

السلامة: يُستخدم العديد من نظم البرامج في التطبيقات التي تُمثل فيها السلامة أهمية قصوى، حيث يمكن أن يؤدي خطأ في البرامج إلى خسارة فادحة في الأرواح أو الممتلكات.

الإنتاج الضخم لبرامج التطبيقات المتنوعة: تختلف اليوم تطبيقات الهواتف المحمولة والألعاب وأدوات أجهزة سطح المكتب والنظم المُستندة إلى الشبكة كثيرًا عن برامج الستينيات والسبعينيات من القرن العشرين. فقد كان من النادر الاستعانة بمصادر خارجية موثوق بها لتطوير البرمجيات. ولم تكن هناك شبكات كبيرة من مطوري التطبيقات الذين يبيعون من خلال متاجر التطبيقات قبل أوائل الألفينيات، أما الآن فمتاجر «أبل» و«أندرويد» تعرض ملايين التطبيقات.

يواجه التفكير الحوسبي باستمرار تحديات فيما يخص تطوّره وتعامله مع هذه المشاكل المعاصرة.

الفصل السادس

التصميم لتلبية احتياجات الإنسان

إن الهدف من وصف كيانات البرامج هو تبسيطها وإزالة تعقيدها، ولكن المبالغة في التبسيط تؤدي إلى الإضرار بجوهر البرنامج ووظيفته الأساسية. فالحكم السديد يأتي من الخبرة، والخبرة تأتي من إساءة الحكم في البداية.

فريدريك بروكس (١٩٨٦)

نحن نبحث عن نوع من الانسجام بين أمرين غير ملموسين: شكل لم نصممه بعد، وسياق لا يُمكننا وصفه بشكل صحيح. إن إجراء عمليات محاكاة لما سوف نُنشئه مُفيد للغاية إذا كنت تستطيع أن تحصل على ملاحظات تقييمية من هؤلاء الذين سيُخبرونك بأخطائك ويُخبرونك كيف تُصلحها.

كريستوفر ألكسندر (١٩٦٤)

كان جورج فورسايت من أوائل رواد الحوسبة الذين رأوا أن الحوسبة تتعامل بشكل أساسي مع القضايا المتعلقة بالتصميم: تصميم أجهزة الكمبيوتر والنظم، وتصميم اللغات للمعالجات والخوارزميات، وتصميم أساليب تمثيل المعلومات ومعالجتها.¹ وكان مهندسو البرمجيات من أوائل مَنْ تعاملوا صراحةً مع التصميم كجزء أساسي من ممارسة الحوسبة. وبالنسبة إلى مهندسي البرمجيات، فإن التصميم يعني تخطيط وإنشاء مُنتجات ونظم برمجية تُحقق المواصفات المُحددة وتكون آمنة وموثوقة بها. كذلك يعني التصميم إنشاء أدوات لدعم إنشاء البرمجيات، بما في ذلك اللغات ذات الصلة وأدوات التحرير والأوامر الصوتية والواجهات الرسومية وممارسات إدارة المشاريع ونظم التحكم

في الإصدارات وبيئات التطوير.² وقد أدى الانتشار الأخير للتطبيقات المفيدة من خلال «متاجر التطبيقات» التجارية إلى دخول الكثير من الأشخاص الذين لم يتلقوا تدريباً رسمياً في هندسة البرمجيات إلى مجال تصميم البرمجيات.

لا يقتصر التصميم فقط على إنشاء النظم. فهو مفهوم مألوف في العديد من المجالات، بما في ذلك الموضة والمنتجات والعمارة. إنه عملية إنشاء وتشكيل أدوات تُلبّي احتياجات الإنسان. وفي مجال البرمجيات، على سبيل المثال، يعني التصميم إنشاء برامج تقوم بالوظائف التي يريد المستخدمون القيام بها. ولا يكفي مصممو البرمجيات بإنشاء البرمجيات التي تُلبّي المواصفات الوظيفية. بل يدعمون عمداً ممارسات مُستخدِمي البرامج وعوالمهم وسياقاتهم وهوياتهم. لا يُعزى النجاح الباهر لهاتف «الآي فون» إلى إمكانيات هذا الهاتف التقنية الكبيرة فحسب، ولكن أيضاً إلى أنه يساعد مُستخدِميهِ في تحديد هوياتهم وإظهار أنفسهم بالصورة الأنيقة المطلوبة. وهناك أيضاً إخفاقات مدوية تُعزى إلى سوء التصميم الذي يُعزز الاستخدام غير الآمن للنظم، مثل شاشات لوحات الطائرات التي لم تُعرض المعلومات الأكثر إلحاحاً في حالات الطوارئ.³

ناقشنا في الفصل الخامس كيف أن مهندسي البرمجيات اكتسبوا الكثير من الخبرة العملية التي عبروا عنها من خلال مبادئ التصميم والأنماط والتلميحات، وكل ذلك لكي تُلبّي البرمجيات الأهداف المرجوة منها، وهي إمكانية الاعتماد عليها، والموثوقية، وقابلية الاستخدام، والسلامة، والأمن. لكنّ اهتمامات التصميم تتجاوز مجرد تحسين عملية إنشاء البرمجيات.

على الرغم من نجاحات هندسة البرمجيات، فإن إخفاقات مشاريع البرمجيات والحوادث الناجمة عنها في ازدياد. ويواصل الأكاديميون نضالهم من أجل إيجاد مناهج دراسية لهندسة البرمجيات يُمكنها أن تُخرّج مطوّري برامج مُحترفين قادرين على قيادة المشاريع حتى اكتمالها دون فشل. يرى ديفيد بارناس، رائد البرمجيات الشهير، أن هذا السعي الأكاديمي محكوم عليه بالفشل في العديد من الأقسام؛ لأن معظم محاولات وضع المناهج الدراسية قامت بوضع «دليل معرفي لهندسة البرمجيات» وتدرّسه بدلاً من قدرات مُصمّمي البرامج المُحترفين الأكفاء.⁴ يتعلم طلاب الحوسبة قواعد هيكلية للبرمجيات ولكن ليس مهارات التصميم المطلوبة لتحقيق برامج جيدة. ويلخّص جدول ٦-١ القدرات التي يعتقد بارناس أنها الأهم. كل هذه القدرات موجهة نحو مجتمعات المُستخدِمين ولا تقتصر على الجوانب الشكلية لعملية تطوير البرمجيات. يُرشدنا التفكير الحوسبي للتصميم إلى طرقٍ لإنشاء نظم حوسبة تكون سلوكياتها مفيدة ونافعة في مجتمعات مُستخدِميها.

التصميم هو مفهوم مألوف في العديد من المجالات، بما في ذلك الموضة والمنتجات والعمارة. إنه عملية إنشاء وتشكيل أدوات تُلبّي احتياجات الإنسان. وفي مجال البرمجيات، فإن التصميم يعني إنشاء برامج تقوم بالوظائف التي يريد المستخدمون القيام بها.

جدول ٦-١: قدرات مطوري البرمجيات

- تصميم واجهات تفاعلية سهلة الاستخدام للمستخدمين.
- تصميم وصيانة برمجيات مُتعددة الإصدارات وقابلة لإعادة الاستخدام.
- ضمان استيفاء منتجات البرمجيات لمعايير الجودة والأمان.
- إنشاء النماذج واستخدامها في تطوير النظام.
- قياس أداء البرمجيات والتنبؤ به وتحليله وتقييمه.
- جدولة عمليات التطوير والصيانة.
- استخدام المقاييس في تطوير النظام.
- إدارة المشاريع المعقدة.

(١) ما المقصود بالتصميم؟

تحول العديد من مطوري البرامج إلى التصميم من أجل تفكيرٍ جديد يُساعدهم في مواجهة تحديات مجال البرمجيات. وقد ترك تاريخ التصميم الطويل في الحوسبة العديد من الأسئلة مفتوحةً أمام المُصمِّمين: ما الفرق بين هندسة البرمجيات والتصميم؟ لماذا استغرق الأمر ٥٠ عامًا حتى تُصبح التصريحات المبكرة حول التصميم بهذه الأهمية؟ ما مدى أهمية التصميم بالنسبة إلى التفكير الحوسبي؟

تتخذ هندسة البرمجيات نهجًا نحو التصميم هو نهج شبه شكلي لإنشاء مجموعة من الوحدات النمطية والواجهات لتحقيق غرض وظيفي مُعين. ويتحدّد هذا الغرض في مجموعة من المتطلبات، كلٌّ منها عبارة عن جُملة برمجية مُحددة قابلة للاختبار. وتتبع عملية الهندسة التقليدية مسارًا خطيًا يبدأ من تحديد المتطلبات وصولاً إلى إنشاء نظام عامل يُسلّم إلى المستخدم:

- المتطلبات
- المواصفات الشكلية

- إنشاء النظام
- اختبار القبول
- التسليم إلى العميل

يُمكن لمهندسي البرمجيات تنفيذ هذه العملية بمنأى عن الأنظار، مُتجنِّبين أي تفاعلٍ مع المستخدمين بين مرحلتَي تحديد المُتطلَّبات والتسليم. تتوافق العملية مع المفاهيم المبكرة في الحوسبة التي تُفيد بأن البرامج هي عبارة عن تعليماتٍ برمجية قابلة للتنفيذ على الآلة للخوارزميات التي تُلبي مواصفات وظيفية مُعينة، وأن المُبرمجين يحتاجون إلى وقتٍ هادئٍ لإنجاز الأمور على النحو الأمثل.

لكن التجربة أظهرت أن عملية الهندسة التقليدية لا تصلح مع النظم المُعقدة. فثلاث مشاريع البرمجيات تقريباً تُسلم في الوقت المُحدَّد وفي إطار الميزانية الموضوعة لها، بينما يتأخَّر تسليم الثلث الآخر أو يتجاوز الميزانية الموضوعة له، أما الثلث المُتبقى فلا يُسلم أبداً إلى العميل. ويُعدُّ أحد أكبر التحديات هو العدد الكبير من الوحدات النمطية والواجهات التي يجب تصميمها وبرمجتها وتتبعها واختبارها؛ فنُظم التشغيل الحديثة، على سبيل المثال، تتكوَّن من مئات الآلاف من الوحدات النمطية. ومن التحديات الأخرى الكبيرة تحديد المُتطلبات بشكلٍ سليم؛ فالكثير من مشاريع البرمجيات تُلبي متطلباتها الشكلية، ولكن العملاء يرونها غير كافية في النهاية. من وجهة نظر المهندس، أُهمَل مُتطلب ضروري. أما من وجهة نظر المُستخدم، فإن المُتطلب المُهمَل واضح لأي فردٍ في مجتمع المُستخدمين. تكمن الفجوة في أن ما هو واضح لمجتمع المُستخدمين قد لا يكون واضحاً للمهندس، الذي لم يكن على درايةٍ بمشكلةٍ كانت جزءاً من سياق المُستخدم غير المُعلن.

تتخذ هندسة البرمجيات نهجاً نحو التصميم هو نهج شبه شكلي لإنشاء مجموعة من الوحدات النمطية والواجهات لتحقيق غرض وظيفي مُعين. ويركز نهج التصميم على العالم الافتراضي الذي يخلقه البرنامج، وعلى الممارسات التي تُشارك المُستخدمين في ذلك العالم، وعلى احتياجات المُستخدم التي يُلبيها هذا العالم.

استجاب المهندسون إلى هذه المشكلات عن طريق محاولة تحسين عملية الإنشاء. فوضعوا أساليب مُتطورة للمقابلات الشخصية لاستخراج المُتطلبات من العملاء، ومن ثمَّ تقليل خطر إهمال أحد المُتطلبات المهمة. وشَفَرُوا «أنماط التصميم» التي اتبعها المُصمِّمون

الناجحون، حتى يتمكن المصممون الأقل خبرةً من تجنب الأخطاء. وقدموا «أساليب مرنة» لإدارة المشاريع يشارك فيها العملاء بشكل صريح في جميع مراحل المشروع الهندسي. غالبًا ما تتميز الإدارة المرنة بالعديد من النماذج المتكررة بسرعة تحت مراجعة مستمرة من قبل الفرق التي تضم ممثلي العملاء.

لم توقف هذه التحسينات التي أُدخلت في عملية الإنشاء موجة فشل النظم، بل أبطأتها. ومن ثم دعا بعض المصممين إلى إجراء تحول جذري في التفكير. وقد وصف تيري وينوجراد، أحد رؤاد الذكاء الاصطناعي والتصميم، هذا التحول على النحو التالي:⁵

لقد ركّز تعليم مُحترفي الكمبيوتر غالبًا على فهم آليات الحوسبة وعلى الأساليب الهندسية التي تهدف إلى ضمان عمل هذه الآليات كما يريد المبرمج. كان التركيز على الكائنات قيد التصميم: الأجهزة والبرامج. وكان الاهتمام الأساسي هو تنفيذ وظيفة محددة بكفاءة. فعندما يقول مهندسو البرمجيات أو المبرمجون إن برنامجًا ما يعمل، فإنهم يقصدون عادةً أنه قوي وموثوق به ويُلبى مواصفاته الوظيفية. فهذه الأشياء مُهمة للغاية. وأي مُصمم يُقدم على تجاهلها سوف يُخاطر بحدوث كارثة.

ولكن، يُعد هذا المنظور الداخلي — رغم تركيزه على الوظيفة والإنشاء — أحادي الجانب. ولكي نتمكن من تصميم برامج تعمل بشكل سليم، فإننا بحاجة إلى الانتقال من منظور المنشئ إلى منظور المُصمم، مع وضع النظام والمُستخدمين والسياق ككل في الاعتبار. عندما يقول المُصمم إن شيئًا ما يعمل بشكل سليم (على سبيل المثال: تصميم غلاف كتاب أو تصميم مجمع سكني)، فإن المصطلح يعكس معنى أوسع. ينتج التصميم الجيد كائنًا يُفيد الناس في سياق القيم والاحتياجات، للحصول على نتائج عالية الجودة وتجربة مُرضية.

قدّم وينوجراد وغيره مصطلح «العالم الافتراضي» في تصميم البرمجيات. تنشئ البرمجيات عالمًا؛ أو بالأحرى سياقًا يتصور فيه المستخدم تجارب ويتعامل معها ويستجيب إليها. ويُطلق على المستخدم الذي يدخل هذا العالم ويتصرّف وفقًا لقواعده ومنطقه اسم «المواطن» أو «السكن»؛ لأن العالم يبدو حقيقيًا أثناء وجود المستخدم فيه. والنقطة الرئيسية هي أن العالم الافتراضي ليس مفهومًا أو تكوينًا ذهنيًا للمستخدم أو المُصمم، بل هو تجربة تبدو حقيقية.

تُعد الألعاب عبر الإنترنت أمثلةً على العوالم الافتراضية. وفيها يهزم اللاعبون الوحوش، ويبحثون عن الكنوز، ويحصلون على مكافآت عند تحقيق المهام، ويتقدمون في المستوى والخبرة. ويقول العديد من اللاعبين إن عالم اللعبة يكون حقيقياً بقدر العالم اليومي عندما يكونون داخله. تنشئ هذه الألعاب عالماً من خلال وجود هدف محدد، وساحة لعب ومعدات، وقواعد وقيم، وقواعد للسلوك المسموح به وغير المسموح به، واستراتيجيات للفوز أو التقدم في اللعبة. لكن فكرة إنشاء عالم افتراضي لا تقتصر على ألعاب الترفيه. فشبكات التواصل الاجتماعي الحالية والخدمات مثل: «أوبر» و«إير بي إن بي» و«إيباي»، كلها تبدو وكأنها ألعاب متعددة اللاعبين، حيث تتطور الاحتمالات المتاحة لك باعتبارك لاعباً وتتحول وفقاً لاختيارات الآخرين وأفعالهم. حتى برامج المُستخدم الفردي مثل: جداول البيانات ومعالجات النصوص وبرامج الرسم، كلها تنشئ عوالم خاصة بها حيث توجد ساحة لعب محددة بدقة ومجموعة من القواعد والاستراتيجيات الأساسية التي يتعين على الجميع اتباعها.

منذ عام ٢٠٠٥، عندما قدمت شركة «أبل» متجر التطبيقات الخاص بها وجعلت أجهزة «الآي فون» قابلةً للتخصيص إلى ما لا نهاية، حيث يمكن للمستخدمين تنزيل التطبيقات التي تناسبهم، ازدهر تطوير برامج التطبيقات. تُقدم متاجر تطبيقات «أبل» و«أندرويد» عبر الإنترنت أكثر من ستة ملايين تطبيق. وأصبحت البرامج سلعةً تجارية تباع في السوق. والنجاح في هذه السوق يكون من نصيب التطبيقات التي يُقيّمها العديد من العملاء بأنها تطبيقات «عالية الجودة».

(٢) جودة البرامج ورضا المُستخدم

في السبعينيات من القرن العشرين، سعى مهندسو البرمجيات إلى جعل جودة البرامج قابلةً للقياس، بناءً على الفرضية القديمة المتمثلة في أن القياس يساعدنا على تحقيق المزيد. وابتكروا نماذج لقياس جودة البرامج. وأصبحت نماذجهم في النهاية معياراً لمنظمة المعايير الدولية (الأيزو)، ووحدة أساسية من وحدات التفكير الحوسبي في هندسة البرمجيات. وتُدرج معايير الأيزو عشرين عاملاً قابلاً للقياس لتقييم الجودة العامة لأي نظام برمجي:

- الصحة
- الموثوقية
- الاكتمال

- سهولة الاستخدام
- الكفاءة
- قابلية الصيانة
- قابلية الاختبار
- التوافقية
- المرونة
- قابلية إعادة الاستخدام
- قابلية النقل
- الوضوح
- قابلية التعديل
- التوثيق
- التحايل على الأعطال
- قابلية الفهم
- الصلاحية
- الفاعلية الوظيفية
- الشمولية
- التوفير والاقتصاد

كان المفترض أن تكون جميع هذه المقاييس خصائص للبرنامج قابلة للقياس بموضوعية. ومن الصعب جدًا تصميم نظام برمجي يُحقق درجات عالية في جميع العوامل العشرين. يُلاحظ أن اثنين من الأهداف الخمسة التقليدية — الأمن والسلامة — غير مُدرَجين في هذه القائمة؛ لأنه لم يعرف أحدٌ كيفية قياس البرامج من هذين الجانبين. لم يقل أحد إن الجودة بسيطة ومباشرة.

تُلقي سوق التطبيقات الجديدة والمزدهرة الضوء على الجودة بوصفها تقييماً للمستخدمين وليس بوصفها إحدى خصائص البرامج. فلا يُوجد معيار ثابت للجودة؛ إذ إن ما يراه أحد المُستخدمين ذا جودة عالية قد يراه الآخر غير ذلك. ويُولى اهتمام أكبر للتصميم بالمعنى الذي حدّده وينوجراد. فكيف يُقيّم المُستخدمون الجودة، ومن ثمّ التصميم الجيد؟ يقدم جدول ٦-٢ ستة مستويات مميزة للرضا في تجربة المستخدم.⁶

يتطلَّب كل مستوى على سُلَّم جودة البرامج مهارة من مهارات التفكير الحوسبي. فالمستويات الدنيا تنطوي على استخدام التفكير الحوسبي على نحوٍ غير مُنظَّم، أما المستويات العليا فتتطوي على استخدام التفكير الحوسبي على نحوٍ مُنظَّم لإنشاء تصميم يتوافق مع ممارسات العملاء ومشاكلهم واهتماماتهم المتطورة. وكلما ارتفع المستوى على سُلَّم الجودة، زادت جوانب التفكير الحوسبي الاحترافية والمتقدمة. وكلما ارتفع المستوى على سُلَّم الجودة، تتوسَّع مهارة التفكير الحوسبي من المُتطلَّبات الرسمية إلى اهتمامات العملاء ومُستقبلهم؛ ويرتفع مستوى رضا العملاء.

المستوى -١: برامج غير موثوق بها

لا يحظى البرنامج بثقة العملاء. وربما يرجع السبب إلى وجود خطأ بالبرنامج، أو إلى تسبُّبه في تعطل أنظمة العملاء، أو إلى كونه يحتجز بياناتهم مُقابل فدية، أو يحمل فيروسات. وقد يظنُّ المرء أن العملاء سيتجنبون البرامج غير الموثوق بها. ولكن الحقيقة هي أنهم كثيرًا ما يستخدمون البرامج غير الموثوق بها، وغالبًا بعد استقطابهم من خلال العروض الاحتيالية أو التصيُّد أو زيارات المواقع الإلكترونية المُخترقة أو ما شابه ذلك. غالبًا ما يتم تجميع البرامج في هذا المستوى دون تفكيرٍ جدِّي في الأهداف الخمسة التقليدية، وتهدف العديد منها إلى استغلال نقاط الضعف لدى العملاء.

جدول ٦-٢: مستويات جودة البرامج وتقييم رضا العملاء

مستوى الجودة	مستوى مهارة التفكير الحوسبي
٤ برامج مُثيرة للإعجاب	تصميم برامج تتوقَّع تطوُّر ممارسات العملاء ومخاوفهم بعد استخدام البرامج
٣ برامج لا تنجم عنها عواقب سلبية	تصميم برامج تتجنَّب الأخطاء المُحتملة لدى العملاء
٢ برامج تتلاءم مع بيئة العمل	تصميم برامج تتوافق مع ممارسات العملاء والمعايير الاجتماعية
١ برامج تفي بكلِّ الوعود الأساسية	تصميم برامج لتلبية مُتطلَّبات جميع العملاء من خلال الاستخدام المنظَّم للتفكير الحوسبي في مجالي البرمجة وهندسة البرمجيات

مستوى الجودة	مستوى مهارة التفكير الحوسبي
٠ برامج موثوق بها نسبياً، يتردد العملاء في استخدامها ويبدون رضا متحفّظاً نحوها	تصميم برامج لا تهتمُّ بالعملاء، مع استخدام مهارات التفكير الحوسبي الأساسية فقط.
١- برامج غير موثوق بها	تصميم برامج تستغلُّ العملاء، مع استخدام محدود لمهارات التفكير الحوسبي

المستوى صفر: الرضا المتحفّظ

يثق العديد من العملاء ببعض الادعاءات التي يُقدمها صانعو البرامج ولكن ليس جميعها، مما يجعلهم على استعدادٍ متحفّظ لاستخدام البرامج. ويتم إصدار الكثير من البرامج وبها أخطاء وثرغرات أمنية، والتي لا يُصلحها المطوّرون إلا بعد سماع شكاوى العملاء وإبلاغهم عن الأخطاء. وتجع منتديات المُستخدمين بالقصص حول المشاكل التي صادفتهم في البرامج ومُطالبتهم بحلول وإصلاحات لها، وعادة ما يغيب ممثلو المطوّرين عن هذه المنتديات.

باستخدام بعض الممارسات الوسيطة للتفكير الحوسبي، ينجح المطوّرون في جعل البرامج تعمل على الرغم من عيوب التصميم وثرغراته التي تتطلب حلولاً. وقد يتسامح المطوّرون مع بيئة التطوير العشوائية وغير المنظمة؛ لأنهم يشعرون أنهم واقعون تحت ضغط كبير لإخراج برنامج قابل للاستخدام إلى السوق قبل أن تصدر برامج منافسة، ويعتقدون أن العملاء سيتسامحون مع العديد من الأخطاء، ويتجنبون المسؤولية من خلال اتفاقيات عدم تحمّل المسؤولية التي يجب على العملاء توقيعها قبل إلغاء قفل البرنامج. وهذا النهج شائع في صناعة البرمجيات. ولكنه يتعرّض للنقد بسبب أن كثرة الأخطاء في البرنامج قد تؤدّي أيضاً إلى ثغرات أمنية. والعملاء المُتشككون لا يكون لديهم ولاء تجاه البرنامج وسوف يتخلون عنه ويذهبون إلى أي منتج آخر يُقدّم عرضاً أفضل.

المستوى ١: برامج تفي بكل الوعود الأساسية

يقيّم العميل المنتج بأنه قد قدّم بالضبط ما وعد به واتفق عليه. يعتمد هذا المستوى من الاكتمال المبدئي على التفكير الحوسبي الأكثر تقدماً. وتُلبي معايير الأيزو هذا المستوى

جيدًا. وغالبًا ما يكون مُطوِّرو البرمجيات في هذا المستوى مُوجَّهين نحو المعايير وتهدف ممارساتهم إلى إنتاج منتجات مناسبة وموثوق بها.

المستوى ٢: برامج تتلاءم مع بيئة العمل

في هذا المستوى يتجاوز التصميم تلبية المتطلبات المعلنة. ويهدف إلى مواكبة البرامج للممارسات الحالية للعملاء واحترام الأمور الثقافية الحساسة وغيرها من المعايير الاجتماعية. ويقيّم العملاء البرنامج بأنه يتناسب تمامًا مع بيئة العمل. ويُعد جهاز الصراف الآلي لدى البنوك مثالًا جيدًا على هذا النوع من المواكبة. يقوم الصراف الآلي بتنفيذ المعاملات المصرفية المألوفة، مما يُتيح للعملاء استخدام الصراف الآلي على الفور دون الحاجة إلى تعلُّم أي شيء خاص أو جديد. ويلاحظ العميل أن البرنامج يُحسِّن قدرة العميل على إنجاز العمل وتنفيذ المهام المهمة.

المستوى ٣: برامج لا تنجم عنها عواقب سلبية

في هذا المستوى يكون المُصمِّم قد درس مجموعة من الطرق المحتملة التي يمكن أن يتسبَّب من خلالها البرنامج في أعطال للعملاء، وينشئ قواعد تشغيل ومعايير تحقُّق لتجنُّبها. بعد فترة من الاستخدام، لا يواجه العملاء أي مشاكل غير متوقَّعة تُسبب تعطل العمل أو تُسبب خسائر. يُقيّم العملاء تصميم المنتج بأنه قد جرى التفكير فيه جيدًا وأنه توقَّع المشاكل التي لم تكن واضحة في البداية. ولا يسبب البرنامج عواقب سلبية غالبًا ما تنشأ في البرامج ذات الجودة الأدنى، مثل التعرُّض للاختراق وللفيروسات، وسهولة التأثير بأخطاء المُستخدم دون توفير إمكانية إلغاء الإجراءات أو العودة إلى حالة سابقة جيدة، وكذلك التعرُّض مع ممارسات المؤسسة، وتضييع الجهود مقابل مكاسب إنتاجية هامشية، والإحباط من العواقب السلبية الأخرى للعملاء أو مؤسساتهم.

في هذا المستوى قد يلجأ المُصمِّمون أيضًا إلى تضمين وظائف لم يطلبها العميل ولكنها ستحول دون شعوره بالإحباط في المستقبل. ومن أمثلة هذه الوظائف نُظُم النسخ الاحتياطي المستمر؛ إذ يمكن للمستخدم استرداد أي نسخة سابقة من ملف ونقل نظام الملفات بأكمله إلى جهاز كمبيوتر جديد بسرعة. وتُعد الأدوات المُستخدمة في إعادة إنشاء الملفات أو الأدلة التالفة مثالًا آخر. مثال ثالث هو عناصر التحكم الداخلية في الإدارة،

وهي تسمح للمصمم بمواصلة العمل مع العميل بعد تثبيت البرنامج من أجل تعديل البرنامج لو تبين أنه تسبب في عواقب سلبية. هذه الإجراءات — تَوَقُّع الأعطال مُسَبِّقًا وتوفّر خدمات الإصلاح بعد التسليم — ضرورية في إنتاج البرامج حتى يكسب رضا المُستخدِم في هذا المستوى. لا تكفي مهارات التفكير الحوسبي في مجالي البرمجة وهندسة البرمجيات لتوجيه مُطوِّري البرمجيات إلى هذا الاتجاه؛ بل يتطلّب الأمر التفكير الحوسبي الموجه نحو التصميم.

المستوى ٤: برامج مثيرة للإعجاب

يقدم البرنامج أعلى مستوى من الجودة، ويتجاوز مجرد تلبية احتياجات العميل ويُحسِّن بشكل كبير حياة المُستخدِم. ويعبر المُستخدِم عن إعجابه الشديد بالمنتج وغالبًا ما يوصي به للآخرين. يُقدَّر العملاء أن المنتج يفهم عالم العملاء ويساهم في توفير الراحة والرفاهية لهم. ولا يمكن للتفكير الحوسبي في مجالي البرمجة وهندسة البرمجيات الاقتراب من هذا؛ لأن إثارة إعجاب العميل لا يمكن أن تُعد من الشروط المطلوبة لنجاح البرامج.

قليل جدًا من النظم البرمجية قد أثارت إعجاب العملاء بحق. وتشمل بعض الأمثلة المبكرة نظام التشغيل «يونكس»، الذي كان منظمًا وأتاح عمليات قوية باستخدام أوامر بسيطة؛ ونظام التشغيل «ماكنتوش» الذي أنشأته شركة «أبل»، الذي أتاح سطح مكتب مختلفًا تمامًا وسهّل الاستخدام وذا عرض نقطي؛ ونظام التشغيل «في إيه إكس في إم إس» الذي أنشأته شركة «ديجيتال إكويمننت كوربوريشن»، وكان مستقرًا بشكلٍ مذهل واحتفظ بالإصدارات السابقة من الملفات للتعافي السريع؛ وبرنامج «فيزيكالك»، الذي كان أول برنامج جداول إلكترونية يُتيح سهولة الإجراءات الحاسوبية ويجعلها متاحة للجميع؛ وبرنامج «لوتس ١-٢-٣»، الذي خَلَفَ «فيزيكالك»، وسمح بصيغ عشوائية في الخلايا وفتح نموذجًا برمجيًا جديدًا؛ وبرنامج «ورد» الذي أنشأته شركة مايكروسوفت، والذي جعل التنسيق الاحترافي للمستندات سهلًا، وحلَّ محل معظم مُعالجات النصوص الأخرى التي كانت موجودة في السوق؛ وبعض الهواتف الذكية، التي تُوفّر بيئة آمنة إلى حدٍّ ما لتحميل التطبيقات التي تخصَّ الجهاز حسب ذوق المُستخدِم وهويته.

أثارت بعض تطبيقات الهواتف الذكية إعجاب مُستخدميها وأسعدتهم أيما سعادة، على سبيل المثال: تُقدم العديد من شركات الطيران ودور النشر والمجلات تطبيقات تُتيح

إمكانية الوصول المباشر إلى محتواها عبر الهاتف المحمول. وتُتيح بعض التطبيقات للمستخدمين إمكانية الوصول إلى الشبكات حيث تُجمَع البيانات من جهاتٍ أخرى كثيرة لتمنح المُستخدم شيئاً يوفّر الكثير من الوقت والجهد. على سبيل المثال: أنشأت شركة أمازون خدمة قارئ «كيندل» التي تُمكن المُستخدمين من شراء الكتب الإلكترونية من متجر أمازون والبدء في قراءتها على الفور من أي جهازٍ يحتوي على تطبيق «كيندل». وتستخدم خرائط جوجل و«أبل» معلومات الموقع من الهواتف الذكية للكشف عن الزحام المروري، وتدمجه في خرائط الشوارع، وتقترح طُرُقاً بديلةً لتجنُّب المناطق المزدحمة. جمعت «بليزارد إنترتينمنت» ما يصل إلى ١٠ ملايين مشترك في لعبة «وورلد أف ووركرافت» عبر الإنترنت بسبب ثرائها، وسهولة الدخول فيها، ورسوماتها التفصيلية. كذلك يسمح تطبيق «أوبر» للمستخدمين باستدعاء الرُّحلات التي يأتي سائقوها إلى موقعهم بالضبط في غضون دقائق. في كل حالةٍ من هذه الحالات، وجد العملاء أن بإمكانهم استخدام التطبيق في القيام بأشياء كانت مُستحيلة سابقاً، بما يتجاوز توقعاتهم.

الشيء المُثير للاهتمام في هذه الأمثلة هو أن العديد منها لم يُحقّق معايير الأيزو المهمة، مثل قابلية النقل أو السرعة أو الكفاءة أو الموثوقية. ومع ذلك، تجاهل العملاء هذه العيوب وأصبحوا مُشترِكين مُتحمسين ومُخلصين للشركات المطوّرة لهذه البرامج.

يعتمد مطوّرو البرامج على ميزات جديدة مع تطور تقنية الذكاء الاصطناعي. ويتطلع الكثير من الناس إلى السيارات من دون سائق، أو المُساعدين الشخصيّين الذين يعرفون روتينك اليومي ويتغلّبون على مشكلة النسيان لديك، وأدوات الواقع الافتراضي التي تسمح لك بالتجول في الأماكن البعيدة، أو التدريب على مهارةٍ أو بيئة جديدة دون التعرُّض للمخاطر، أو الوصول إلى أنواعٍ جديدة من الترفيه. ولذا، لم يكن التفكير الحوسبي الذي يأخذ التصميم إلى أعماق الجوانب التنظيمية والبشرية والاجتماعية للحوسبة مُهمّاً كما هو اليوم.

لكن إعجاب العملاء لن يدوم إذا كان مَبْنِياً على البرنامج فحسب. فبعدَ التعود على البيئة الجديدة، ستتسع آفاق العملاء ويطمحون إلى المزيد. ولذا، نجد أن نظم «يونكس» و«ماكنتوش» و«في إم إس» و«فيزيكالك» و«وورد» لم تُعد مُرضيةً للكثيرين اليوم. ويستثمر مُنتجو البرامج الآن جهداً كبيراً في معرفة عملائهم وتوقع ما سيجلب لهم السعادة ويثير إعجابهم بعد ذلك.

(٣) التفكير الحوسبي الموجّه نحو التصميم

يُركز التفكير الحوسبي الموجّه نحو هندسة البرمجيات على التنفيذ الصحيح للمتطلبات الوظيفية المذكورة بوضوح في البرنامج. وتكون مقاييس نجاحه خصائص يُمكن ملاحظتها في البرنامج أو في بيانات استخدامه.

أما التفكير الحوسبي الموجّه نحو التصميم، فهو يُركز أيضًا على إنشاء عوالم افتراضية يستطيع المُستخدمون البقاء فيها وتحقيق هدفٍ مُعين مُهمٍّ لهم. وتكون مقاييس نجاحه هي تقييمات مستوى الرضا والجودة من قبل المُستخدمين.

يُعتبر التفكير الحوسبي الموجّه نحو هندسة البرمجيات مفيدًا بشكل خاص للنظم الكبيرة التي يجب أن تعمل على نحو موثوق به في البيئات التي تقتضي الحرص على السلامة. فمن الأهمية بمكان أن تكون نظم مراقبة الحركة الجوية ونظم التحكم في محطات الطاقة النووية وفي العربات الجوّالة على سطح المريخ قد رُوِّعَت فيها الهندسة البرمجية السليمة. أما التفكير الحوسبي الموجّه نحو التصميم، فهو مُفيد للغاية في البرامج التي يجب أن تتلاءم مع مجتمعات العملاء، وتُسهّل استخدامها، وتقدم قيمةً عظيمة. والتفكير الحوسبي الموجّه نحو التصميم لا يستغني عن التفكير الحوسبي الموجّه نحو هندسة البرمجيات؛ فهو يستغلُّ أي فرصة لتضمين وظائف رائعة لم يطلبها العملاء بعد. وكما ناقشنا آنفًا في هذا الفصل، فلتوصيف التفكير الحوسبي الموجّه نحو التصميم، اقترحنا ستة مستويات يُقيّم بها العملاء جودة البرنامج ودرجة رضاهم عنه. تُعد صحة البرنامج ضرورية بالطبع ولكنها تؤدي إلى رضا العملاء على المستوى الأول فحسب. أما المستوى الأعلى من الرضا، فينشأ في سياق العلاقة بين العميل ومُطوّر البرنامج. فسيقول العميل الراضي جدًّا عن البرنامج إن مُطوّر البرنامج قد بذل جهدًا في فهم بيئة العمل الخاصة بالعميل، وإنه متاح للمساعدة في حلِّ المشكلات واقتناص الفرص، وإنه ربما يشارك في بعض المخاطر في المشروعات الجديدة، وبشكلٍ عام يهتم بالعميل. وينظر مطوّر البرامج حاليًّا إلى التصميمات والخدمات التي تُسعد العميل سعادةً حقيقية. وعندما ينجحون، فإننا نرى موجاتٍ جديدة من التطبيقات المنافسة.

الفصل السابع

العلوم الحوسبيّة

لا تُحاول العلوم الشرح، بل لا تحاول التفسير، وإنما تركز بشكلٍ رئيسي على إنشاء نماذج.

جون فون نيومان (١٩٥٥)

تُشير العلوم الحوسبيّة إلى فروع كل مجال علمي مُختص في استخدام الحوسبة، مثل الفيزياء الحوسبيّة والمعلوماتية البيولوجية والعلوم الإنسانية الرقمية. على الرغم من أن الأساليب العددية كانت سمةً من سمات العلم منذ قرون، فإن محاكاة النظم المُعقدة نادرًا ما كانت تُطبّق قبل أجهزة الكمبيوتر. طوّر العلماء نماذج رياضية، يُعبرون عنها عادةً في صورة مجموعاتٍ من المعادلات التفاضلية، ولكن إذا لم يتمكّنوا من إيجاد حلول مُغلقة الشكل للمعادلات، فإن تعقيد النماذج عادةً ما كان يحول دون توصّلهم إلى أي أسلوبٍ فعّال لحساب النتائج. على الرغم من أن أجهزة الكمبيوتر بدأت تدريجيًا تغزو جميع مجالات العلم في الخمسينيات من القرن العشرين، فإن أجهزة الكمبيوتر الفائقة في الثمانينيات كانت نقطة تحوّل في حشد قوة الحوسبة لحلّ عددٍ متزايد من هذه المعادلات من خلال المحاكاة. وهذا بدوره أدى إلى تزايد نماذج المحاكاة في العلم بشكلٍ هائل، وأدى بعضها إلى اكتشافات حازت جوائز نوبل. وبحلول منتصف الثمانينيات من القرن العشرين، كان العديد من العلماء يعتبرون المحاكاة باستخدام الكمبيوتر طريقة جديدة لممارسة العلم، إلى جانب الطرق التقليدية القائمة على النظرية والتجربة.

في ثمانينيات القرن العشرين، اجتمع علماء من مجالات علمية عديدة لتحديد «المشكلات ذات التحديات الكبرى»؛ وهي المشكلات التي أعطتها نماذجهم حلولاً تتطلّب عمليات حوسبيّة ضخمة. ومن خلال استقراء قانون مور بشأن تضاعف سرعة الشريحة

الإلكترونية كل عامين، تمكنوا من التنبؤ بدقة كبيرة بالموعد الذي ستنتج فيه الحوسبة حلولاً لهذه التحديات. على سبيل المثال: توقع مهندسو الطيران أنه بحلول عام ١٩٩٥، سيكون بإمكانهم تصميم طائرة ركاب آمنة باستخدام المحاكاة كبديل لاختبارات النفق الهوائي؛ وقد حققت شركة «بوينج» ذلك مع طائرتها رقم «٧٧٧»، التي أجرت رحلاتها التجريبية الأولى في عام ١٩٩٤.

أصبحت عمليات المحاكاة باستخدام الكمبيوتر جيدة جداً لدرجة أنه يمكن استخدامها كمَنَصَّات تجريبية. ومن خلال المحاكاة، تمكن العلماء من استكشاف سلوك النظم المعقدة التي لم تكن هناك نماذج تحليلية لها. كذلك فتحت المحاكاة الباب أمام طريقة جديدة لاستكشاف العمليات الداخلية للطبيعة من خلال تفسير العمليات الطبيعية على أنها عمليات معلوماتية ومحاكاتها لفهم كيفية عملها.

كان التحول الحوسبي للعلوم وأساليبه وأدواته الجديدة واسع الانتشار، وكان التغيير جذرياً. ووُصِفَت الأساليب الحوسبيّة بأنها أهم تحول في النموذج العلمي منذ ميكانيكا الكم. وكانت الثورة في مجال العلوم الحوسبيّة بمثابة البشير لظهور موجة جديدة من التفكير الحوسبي. ولكن على عكس الموجات السابقة من التفكير الحوسبي — التي بدأها علماء الكمبيوتر — بدأ العلماء في مجالات أخرى موجة التفكير الحوسبي الجديدة. وأصبحت العلوم الحوسبيّة قوة دافعة رئيسية في تطوير التفكير الحوسبي خارج نطاق الحوسبة.

خلال الثمانينيات والتسعينيات من القرن العشرين، قدم التفكير الحوسبي مجموعة الأدوات العقلية للعلوم الحوسبيّة الجديدة، والتي طُورت على نحوٍ مشترك عبر العديد من المجالات. وأصبح التفكير الحوسبي مهارة ضرورية للباحثين في المجالات التي يمكن فيها تفسير الظواهر الطبيعية بوصفها عمليات معلوماتية. وللمفارقة، فإنه في الوقت الذي رأى فيه العلماء السابقون أن الحوسبة ليست علماً لأنه لا توجد عمليات معلوماتية طبيعية، وجد الجيل الجديد من علماء الحوسبة عمليات معلوماتية في كل أنحاء الطبيعة. وكما هو الحال مع علماء الكمبيوتر في الخمسينيات والستينيات من القرن العشرين، تعلم علماء الحوسبة التفكير الحوسبي من ممارسة تصميم عمليات الحوسبة لاستكشاف الظواهر وحل المشكلات في مجالاتهم.

في هذا الفصل، سوف نوضح كيف أصبح التفكير الحوسبي جوهرياً في العلوم، ونشرح عددًا من ممارسات التفكير الحوسبي في العلوم الحوسبيّة، ونناقش الطرق

الجديدة التي يفسر بها علماء الحوسبة مجالهم. وقد جلب عصر الحوسبة الإلكترونيّة بعض التطورات المذهلة للعلوم في ثلاثة جوانب: المحاكاة، والتفسير المعلوماتي للطبيعة، والأساليب العددية.

(١) العلوم والحوسبة: أصدقاء قدامى

لطالما كانت العلوم والحوسبة أصدقاء قدامى منذ قرون. على مدار معظم تاريخ العلوم والتكنولوجيا، كان العلماء يؤدون نوعين شائعين من الأدوار. الأول: هو الدور التجريبي، ويعني جمع البيانات لاستكشاف الظواهر وفصلها، ووصف التكرارات، والكشف عن متى تعمل الفرضية ومتى لا تعمل. والثاني: هو الدور التنظيري، الذي يُعنى بتصميم النماذج الرياضية لشرح ما هو معروف بالفعل واستخدام النماذج للتنبؤ بما هو غير معروف. وكان كلا الدورين موجودين في العلوم قبل ظهور أجهزة الكمبيوتر بفترة طويلة. استخدم العلماء الحوسبة في كلا الدورين. وأنتج أصحاب الدور التجريبي بيانات كان يجب تحليلها وتصنيفها وبيان مطابقتها للقوانين المصوغة رياضياً. واستخدم أصحاب الدور التنظيري التفاضل والتكامل لصياغة نماذج رياضية للعمليات الفيزيائية. وفي كلا الدورين، لم يتمكّنوا من التعامل مع المسائل الكبيرة جدّاً؛ لأنّ عمليات الحوسبة كانت موسّعة ومُعقدة للغاية.

ثم ظهر دور ثالث، يتمثّل في العلماء الذين رأوا فرصاً جديدة في استخدام أجهزة الكمبيوتر كمحاكيّات لم يستخدمها لا التجريبيون ولا المنظّرون. وكان رواد الحوسبة في كلية مور، موطن مشروع المُكامل الرقمي الإلكتروني، من أوائل مَنْ رأوا أنّ المحاكاة بالكمبيوتر يُمكن أن تُحوّل أيّ كمبيوتر إلى مُختبر. ورأوا أنّ تقييم النماذج وإنتاج بياناتٍ للتحليل بمثابة حدود جديدة للعلوم. ويتطلّب عبور هذه الحدود طرقاً جديدة لدمج النمذجة والمحاكاة في الأبحاث، وكذلك دمج أنواع جديدة من التفكير الحوسبي ذات صلة مباشرة بالعلوم.

تتطلب عمليات النمذجة والمحاكاة الواسعة النطاق ترقّيات كبيرة في البرمجيات الرياضية. وقد شارك محللو الأعداد، وهم مجموعة من علماء الكمبيوتر الأوائل، بشكلٍ كبير في جهود تحسين البرمجيات الرياضية من أجل حساب النماذج الرياضية بكفاءة على أجهزة الكمبيوتر. وكانت لديهم تخوّفات خاصة بشأن تمثيل الأعداد وإجراء العمليات الحسابية الطويلة باستخدام آلاتٍ لا يُمكنها أن تُقدّم سوى مستوياتٍ دقةٍ محدودة؛ وكانت السيطرة على أخطاء التقريب وزيادة السرعة الحسابية من المخاوف الرئيسية.

في أواخر الثمانينيات من القرن العشرين، قدّر جون رايس، أحد رواد البرمجيات الرياضية، أن البرمجيات الرياضية تحسّنت في الأداء بعامل ١٢١٠ منذ الخمسينيات من القرن العشرين. ويرجع السبب في ١٢١٠ من هذا التحسّن إلى زيادة سرعة الأجهزة، وفي ١٢١٠ الأخرى إلى تحسين الخوارزميات. ولم يكن قانون مور هو السبب الوحيد في تحسن الأساليب العددية. فقد تولّت براعة مُحلّي الأعداد بقية العمل.

يبدو أن فكرة استخدام التفاضل والتكامل لتقييم النماذج الرياضية بدت بديهية بالنسبة إلى واضعي النماذج؛ لأن معادلاتهم كانت عادةً معادلات تفاضلية. ويمكن وصف العديد من العمليات الفيزيائية من خلال ربط قيمة دالة عند نقطة ما بقيم الدالة عند نقاط الجوار. على سبيل المثال: يمكن لواضعي النماذج الذين يعرفون أن معدل تغير الدالة $f(t)$ هو دالة أخرى $g(t)$ تستطيع حساب قيم $f(t)$ بسلسلة من الخطوات الزمنية الصغيرة بحجم Δt باستخدام معادلة الفروق $f(t + \Delta t) = f(t) + g(t)\Delta t$ ويكون تسلسل النقاط الزمنية التي يفصل بينها Δt هو عيّنة من السلسلة الزمنية للدالة. ويمكن تطبيق هذه الفكرة بسهولة على الدوال عبر إحداثيات الفراغ (x, y) من خلال ربط $f(x, y)$ بـ $f(x + \Delta x, y)$ و $f(x, y + \Delta y)$ على شبكة ثنائية الأبعاد. صاغ جون فون نيومان، عالم الرياضيات الموسوعي الذي ساعد في تصميم أول جهاز كمبيوتر ذي برنامج مُخزن، خوارزميات لحلّ نُظم المعادلات التفاضلية على الشبكات المنفصلة.

ونظرًا إلى تعقيد عمليات الحوسبة المتضمنة في هذه المحاكاة، أصبحت أجهزة الكمبيوتر الفائقة العالية الأداء مهمة جدًا في العلوم. ذلك أنها تنفرد دون غيرها بقدرتها على حل المعادلات التفاضلية عدديًا عبر شبكات معقدة. وباستخدام أجهزة الكمبيوتر الفائقة، حلّ علماء الحوسبة مشكلات التحديات الكبرى التي ظهرت في أواخر الثمانينيات من القرن العشرين.

لمئات السنين، كانت النظرية والتجربة الطريقتين الوحيدتين لممارسة العلم. وغيّرت أجهزة الكمبيوتر الفائقة هذا الأمر، مما مهّد الطريق أمام نهج جديد لممارسة العلم قائم على الاستكشاف الحوسبي والنمذجة. وكان هذا أهم تحوّل في النموذج العلمي منذ ميكانيكا الكم. وأذنت الثورة التي حدثت في العلوم الحوسبية بقدوم موجة جديدة من التفكير الحوسبي.

مع غزو الحوسبة للعلوم حدث شيء غير متوقّع. فبدلاً من أن تصبح الحوسبة أكثر تشابهاً مع العلوم الأخرى، أصبحت العلوم الأخرى أكثر تشابهاً مع الحوسبة. ووجد

العلماء الذين استخدموا أجهزة الكمبيوتر أنفسهم يُفكِّرون بشكلٍ مختلف — بمنهج حوسبي — ويصمّمون طرقاً جديدة لإحراز التقدُّم في العلم. فمن خلال محاكاة تدفقات الهواء حول جَنَاح الطائرة باستخدام معادلة نافيه-ستوكس، من خلال تقسيمها إلى شبكة تحيط بالطائرة، لم يُعد مهندسو الطيران في حاجة إلى الأنفاق الهوائية ولا إلى إجراء العديد من الرحلات التجريبية. وأجرى علماء الفلك محاكاة لاصطدام المجرات. كما أجرى علماء الاقتصاد الكلي محاكاة للسيناريوهات الموجودة في الاقتصادات الوطنية والعالمية. وأجرى علماء الكيمياء محاكاة لتدهور الدروع الحرارية لمسبار الفضاء عند دخول الغلاف الجوي. سمحت المحاكاة للعلماء بالوصول إلى أبعادٍ لا يمكن للنظرية والتجربة الوصول إليها. وأصبحت المحاكاة طريقةً جديدة لممارسة العلم. وأصبح العلماء مُستكشفين حوسبيين، بالإضافة إلى كونهم تجريبيين ومنظرين.

لثلاث السنين، كانت النظرية والتجربة الطريقتين الوحيدتين لممارسة العلم. وغيّرت أجهزة الكمبيوتر الفائقة هذا الأمر، مما مهّد الطريق أمام نهج جديد لممارسة العلم قائم على الاستكشاف الحوسبي والنمذجة. وكان هذا أهم تحول في النموذج العلمي منذ ميكانيكا الكم. وأذنت الثورة التي حدثت في العلوم الحوسبيّة بقدوم موجة جديدة من التفكير الحوسبي.

وكما سمح التحليل العددي بمحاكاة أفضل، سمحت المحاكاة الأفضل بظهور نموذج علمي آخر جديد: تفسير الظواهر التي تحدث في العالم من خلال العمليات المعلوماتية. يمكن تعلّم الكثير عن عملية فيزيائية ما من خلال تفسيرها كعملية معلوماتية ومحاكاة العملية المعلوماتية على جهاز كمبيوتر. على سبيل المثال: أصبح هذا النموذج العلمي ركيزة أساسية لعلم الأحياء الحديث، ولا سيما مع التسلسل الجيني وتعديل الجينات.¹ بالنسبة إلى الكميات التي خضعت للنمذجة، تتصرّف العملية الحقيقية كما لو كانت عملية معلوماتية. وغالبًا ما يتم الجمع بين نهج المحاكاة والنهج التفسيري، كما هو الحال عندما تُوفّر العملية المعلوماتية محاكاةً للعملية الفيزيائية التي تضع نموذجًا لها.

ظهر مصطلح «العلوم الحوسبيّة»، ومصطلح «التفكير الحوسبي» المرتبط به، خلال الثمانينيات من القرن العشرين. وفي عام ١٩٨٢، حصل كينيث ويلسون على جائزة نوبل في الفيزياء لتطويره نماذج حوسبيّة أنتجت اكتشافاتٍ جديدة مُذهلة حول التغيرات الطّورية في المواد. وصمم أساليب حوسبيّة لإيجاد قيم معادلات مجموعات إعادة المعالجة،

والتي استخدمها لمراقبة كيفية تغير المادة من طَوَّر إلى آخر، مثل اتجاه القوة المغناطيسية في المغناطيس الفريتي. وقد أطلق حملة للاعتراف بالعلوم الحوسبيَّة واحترامها. وجادل بأن جميع التخصصات العلمية لديها مشاكل «تحديات كبرى» ستؤدي إلى عمليات حوسبة ضخمة.² استخدم ويلسون ومُلهَمون آخرون مصطلح «العلوم الحوسبيَّة» للفروع الناشئة من العلوم التي جعلت الحوسبة أسلوبها الرئيسي. ورأى الكثير منهم الحوسبة كنموذج جديد للعلوم، يُكمل النماذج التقليدية القائمة على النظرية والتجربة. ونتيجةً لاقتناعهم بالفوائد التي سيجلبها التفكير الحوسبي للعلوم، أطلقوا حركةً سياسية لتدبير التمويل اللازم للبحث في العلوم الحوسبيَّة، والتي بلغت ذروتها بإصدار «قانون الحوسبة العالية الأداء» في عام ١٩٩١ من قِبَل الكونجرس الأمريكي، وإبراز التفكير الحوسبي في العلوم أمام الرأي العام.

من الجدير بالذكر أن كلاً من العلوم الحوسبيَّة والتفكير الحوسبي في العلوم ظهرا من داخل المجالات العلمية؛ ولم يُجلبا من علوم الكمبيوتر. في الواقع، تباطأ علماء الكمبيوتر في الانضمام إلى الحركة. وعلى الرغم من أن المُحلِّلين العدديِّين غالباً ما شعروا بأنهم منبوذون من الرياضيات خلال خمسينيات القرن العشرين ومنبوذون من الحوسبة خلال السبعينيات، فقد كانوا مشاركين طبيعيين في العلوم الحوسبيَّة. ولحُسن الحظ، لم يستمرَّ هذا الوضع؛ فالمحللون العدديون أعضاء مُهمُّون في مجال الحوسبة.

لقد أثبتت الحوسبة أنها مفيدة للغاية لإحراز التقدُّم في العلوم والهندسة لدرجة أن كل مجالٍ من مجالات العلوم والهندسة تقريباً طوَّر فرعاً «حوسبياً». وفي العديد من المجالات، أصبح الفرع الحوسبي ذا أهمية حاسمة للمجال. على سبيل المثال: يُنظر إلى علم الأحياء على أنه علم معلوماتي.³ ويصمم الكيميائيون الجزيئات ويُجرون محاكاةً لها لمعرفة كيف سيكون سلوكها تحت الظروف الواقعية. وتختبر شركات الأدوية الجزيئات من خلال المحاكاة لمعرفة ما إذا كانت ستُجدي في مكافحة بعض الأمراض. تنتشر الأساليب الحوسبيَّة في المجالات غير التجريبية عادةً، مثل العلوم الإنسانية والاجتماعية. وسوف يستمرُّ هذا التوجُّه. وسوف تغزو الحوسبة أعماق جميع المجالات.

نظرًا إلى أن التفكير الحوسبي قد ساهم في تقدُّم العلوم — من خلال توفير أساليب أفضل للتحليل العددي والمحاكاة المُتقدمة والتفسير المعلوماتي للعمليات الفيزيائية — سيُقرر العديد من الأشخاص تعلُّم المهارات المطلوبة من المُصمِّمين والمُفكرين الحوسبيين.

(٢) التفكير الحوسبي في العلوم

يتمتع التفكير الحوسبي في العلوم بجانبين. أولاً: جانب المهارات العقلية التي تسهّل تصميم النماذج الحوسبيّة للعمليات الطبيعية ولأساليب تقييم النماذج. وتتردّد عبارة «النمذجة والمحاكاة» بشكلٍ مُتكرر في هذا الجانب من التفكير الحوسبي في العلوم. ونالت مصطلحات الحوسبة استحساناً بين علماء الحوسبة؛ لأنها ميزت الأساليب الحوسبيّة الجديدة لممارسة العلم عن الأساليب التقليدية للنظرية والتجريب.

الجانب الثاني من التفكير الحوسبي في العلوم هو مهارة تفسير العالم كعملياتٍ معلوماتية. بدلاً من طرح سؤال عن الحوسبة، مثل: «هل يمكن أتمتة عملية معلوماتية بكفاءة؟» يسأل علماء الحوسبة: هل يمكن للعمليات المعلوماتية التي جرت مُحاكاتها أن تكون نسخةً من عملية حقيقية؟ وما نوع العملية المعلوماتية التي تنشئ ظاهرةً تمّت ملاحظتها؟ ما الآلية الحوسبيّة وراء عملية تمّت ملاحظتها؟ على سبيل المثال: يدرس العديد من علماء الأحياء التفاعلات بين الحمض النووي والبروتين كعملياتٍ معلوماتية أملاً في تصميم حمض نووي مُستقبلي يشفي الأمراض ويُطيل العمر. ويأمل علماء الفيزياء أنه من خلال تفسير الفيزياء كعملياتٍ معلوماتية، يُمكنهم معرفة المزيد عن الجسيمات التي يصعب اكتشافها من خلال مُحاكاة الجسيمات.

نرى إذن أن التفكير الحوسبي في العلوم الحوسبيّة له اتجاه مختلف عن التفكير الحوسبي في علوم الكمبيوتر. تهتمُّ العلوم الحوسبيّة باستخدام النمذجة والمحاكاة لاستكشاف الظواهر واختبار الفرضيات والتنبؤ بها في مجالاتها الخاصة. أما علوم الكمبيوتر فتهتمُّ بتصميم الخوارزميات لحلّ المشكلات. وغالباً ما لا يتولّى العلماء والمهندسون الذين يُصمّمون عمليات المحاكاة بصياغة بيانات المشكلة؛ وإنما يدرسون سلوك الظواهر. أما الأشخاص الذين يعملون في مجال الحوسبة، فهم لا يستخدمون عمليات المحاكاة لفهم آلية عمل الطبيعة؛ وإنما كل ما يفعلونه هو أنهم يُصممون برامج لإنجاز مهامٍّ للمستخدمين.

يجب أن يضع الأشخاص الذين يعملون في مجال الحوسبة، والعلماء الذين يتطلّعون إلى التعاون، هذا الفرق في اعتبارهم. وسوف ينجح التعاون بينهم إذا طوّر الأشخاص الذين يعملون في مجال الحوسبة فهماً للمجال العلمي، وطوّر العلماء فهماً لمجال الحوسبة. على سبيل المثال: شهد واحدٌ منّا (بيتر) شخصياً انفصلاً بين علماء الحوسبة وعلماء الكمبيوتر في الثمانينيات من القرن العشرين. دعا فريق من حملة الدكتوراه في مجال علم ديناميكا

الموائع الحوسبيّة حملةً الدكتوراه في علوم الكمبيوتر للانضمام إليهم، فاكتشفوا أن علماء الكمبيوتر لا يفهمون ديناميكا الموائع بما فيه الكفاية ليتمكّنوا من التعاون بشكل مُثمر. ولم يستطيعوا التفكير في ديناميكا الموائع الحوسبيّة بالسهولة نفسها التي يتمتّع بها خبراء ديناميكا الموائع. وانتهى الأمر بتعامل علماء ديناميكا الموائع مع علماء الكمبيوتر كمُبرمجين وليس كأقرانٍ لهم، مما أثار استياء علماء الكمبيوتر إلى حدّ كبير.

(٣) النماذج الحوسبيّة

يمكن أن يكون مصطلح «النموذج الحوسبي» أيضًا مصدرًا لسوء الفهم. فالنماذج الحوسبيّة بالنسبة إلى العالم هي مجموعات من المعادلات، وغالبًا ما تكون معادلات تفاضلية تصف عملية فيزيائية، ويمكن استخدام هذه المعادلات حوسبيًا لتوليد بياناتٍ عديدة حول العملية. وغالبًا ما تكون عمليات المحاكاة هي الخوارزميات التي تقوم بذلك. في المقابل، النموذج الحوسبي في الحوسبة هو آلة مجردة تعمل على تشغيل برامج مكتوبة بإحدى لغات البرمجة. وغالبًا ما يُستشهد بآلة تورينج في الحوسبة على أنها النموذج النظري الأساسي لجميع عمليات الحوسبة، على الرغم من أنها بُدائية للغاية لدرجة تمنعها من أن تكون مفيدة لمُعظم الأغراض.

يستخدم العلماء بشكلٍ روتيني الآلات المجردة بالمعنى الحوسبي؛ لأن كل لغة برمجة مألوفة مُرتبطة بآلة مجردة. على سبيل المثال: تُقدم لغة فورتران آلة مجردة تُجيد على وجه الخصوص إيجاد قيم التعبيرات الرياضية. وتُقدّم لغة جافا آلة مجردة تستضيف عددًا كبيرًا من «الكائنات» المُستقلّة التي تتبادل إرسال الرسائل واستقبالها بشكلٍ متزامن فيما بينها. كما تحتوي لغة «سي++» أيضًا على كائناتٍ ولكنها أقرب إلى الآلة الفعلية، ومن ثم تُنتج نصوصًا برمجية قابلة للتنفيذ على قدر أكبر من الكفاءة.

تُعد النماذج الحوسبيّة في العلوم الحوسبية بمثابة مُخططات لمحاكاة العمليات المعلوماتية. وتتحوّل هذه المخططات إلى برامج كمبيوتر يجري تشغيلها على آلات تصوّرية لمحاكاة العمليات الطبيعية.

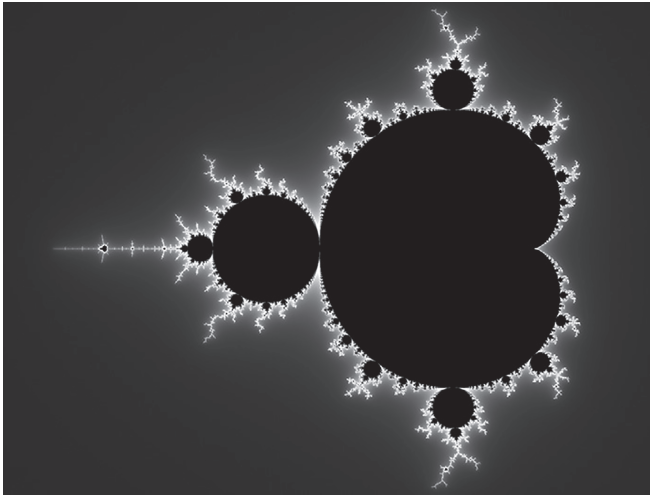
(٤) النمذجة والمحاكاة

تتمتع العلوم الحوسبيّة بمجموعةٍ غنية من الأساليب لنمذجة العمليات الطبيعية ومحاكاتها وتنفيذها. وسوف نُلقِي نظرةً على خمسة أمثلة توضح هذه المجموعة، كما سنُشير إلى بعض سمات التفكير الحوسبي الأساسية في النماذج وعمليات المحاكاة.

مجموعة ماندلبرو

تمرّ العديد من عمليات المحاكاة عبر جميع النقاط الموجودة على شبكة ما، وتحسب دالة عند كل نقطة، ثم تُصوّر النتيجة عن طريق تعيين ألوان للأعداد على نقاط الشبكة. وتُعد مجموعة ماندلبرو مثالاً جيداً على عملية حوسبيّة تكشف سلوكيات لم يشته بها أحد من خلال فحص المعادلات. في تصوّر ماندلبرو، لكل نقطة على الشبكة، يحسب الكمبيوتر متسلسلة من القيم بناءً على معادلة بسيطة على الأعداد المركبة، ويُعين ألواناً لتلك النقاط: إذا تقاربت المتسلسلة المحسوبة (ظلت في نطاق مُعين)، فلون النقطة باللون الأسود، أما إذا تباعدت، فلونها باللون الأزرق أو الأصفر. والآن كرّر هذا لكل النقاط على الشبكة.⁴

عند تعيين لون كل نقطة إلى بكسل، تظهر مجموعة ماندلبرو على شاشة الرسومات. لم يظن أحد أن مثل هذه العملية الحوسبيّة البسيطة سوف تُنتج مثل هذا الكائن الجميل الغامض (انظر الشكل التالي). يمكن للمرء تحديد مُربع صغير في أي موضع على الرسم، وتكبيره، وتغطيته بشبكة، وحساب الألوان لجميع نقاط الشبكة، وعندئذٍ سيرى المزيد من نسخ مجموعة ماندلبرو تظهر بأحجام أصغر. ويكشف كلُّ تكبير جديد المزيد من المجموعات. ويستمر الأمر هكذا إلى ما لا نهاية. وقد أطلق ماندلبرو على هذا السلوك المتكرّر ذاتياً في جميع المقاييس مصطلح «الكسيريّات» (أو الفراكتلات).



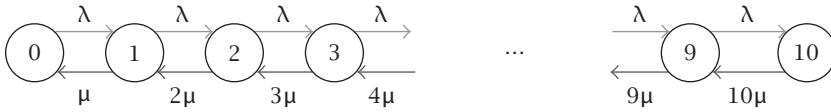
كانت فكرة الكسريات (التمائل الذاتي على المقاييس المختلفة) هي مفتاح خوارزميات زمرة إعادة التنظيم (الاستنظام) التي وصل إليها كين ويلسون والتي أدت إلى اكتشافات جديدة في الفيزياء عندما تمت محاكاتها على جهاز كمبيوتر فائق، وبفضلها حصل على جائزة نوبل. تُستخدم فكرة الكسريات في نظم التصور لحوسبة صور رسومية واقعية، مثل الأشجار أو الأفاق، بسرعة.

مهندسو الاتصالات

عند تصميم أول مكاتب السنترالات الهاتفية في أوائل القرن العشرين، واجه مهندسو الاتصالات مشكلة خطيرة في التصميم. في مدينة بها عدد K من العملاء، هناك احتمال وجود K^2 من الاتصالات. إن ضمان اتصال كل عميل بأي عميل آخر في أي وقت يرغب فيه سيكون معقدًا ومكلفًا للغاية، خاصة أن معظم العملاء لا يتحدثون على الإطلاق معظم الوقت. للتغلب على عامل التعقيد والتكلفة، قرّر المهندسون إنشاء مفاتيح تبديل يُمكنها التعامل مع ما يصل إلى عدد N من المكالمات في الوقت نفسه (علمًا بأن N أقل بكثير من K). وبطبيعة الحال، هذا يُعرضنا لاحتمال عدم تمكّن العميل من الحصول على نغمة الاتصال إذا كان مكتب السنترال يُنفذ بالفعل عدد N من المكالمات. وكانت المشكلة التي تواجه المهندسين في التصميم هي كيف يختارون N بحيث تقل احتمالية مواجهة إشارة «مشغول»، بحيث تُصبح مثلًا ٠,٠٠١. يوفر النموذج الحوسبي التجريبي حلًا لذلك. يحتوي النموذج على الحالات $n = 1, 2, 3, \dots$ ، حيث N يمثل عدد المكالمات الجارية وصولًا إلى الحد الأقصى N ، وهو هنا $n = 10$. تحدث طلبات بدء مكالمات جديدة بشكل عشوائي بمعدل λ . وينتهي المتصلون الفرديون المكالمات بشكل عشوائي بمعدل μ . ويؤدي وصول مكالمات جديدة إلى زيادة الحالة بمقدار ١، كما يؤدي إنهاء المكالمات إلى تقليل الحالة بمقدار ١. يُمثل مخطط الحالة الموجود في الشكل التالي التنقل عبر الحالات المُمكنة. يعرف مهندسو الاتصالات $p(n)$ بأنه جزء الوقت الذي يكون فيه النظام في الحالة n ويُمكنهم إثبات معادلة الفروق $p(n) = (\frac{\lambda}{n\mu})p(n-1)$. يحسبون جميع الاحتمالات من خلال تخمين $p(0)$ ، وحساب كل $p(n)$ من سابقه $p(n-1)$ ثم إعادة تنظيمه بحيث يكون مجموع كل $p(n)$ هو ١. ثم يوجدون أكبر N بحيث يكون $p(N)$ أقل من الحد الأدنى المُستهدف. على سبيل المثال: إذا وجدوا $p(N) = 0.001$ عندما يكون $N = 10$ ، فإنهم

يتوقَّعون أن يكون لدى المُتصل الجديد فرصة قدرها ٠,٠٠١ لعدم الحصول على نغمة الاتصال عندما تكون القدرة الاستيعابية للسنترال هي ١٠ مكالمات.

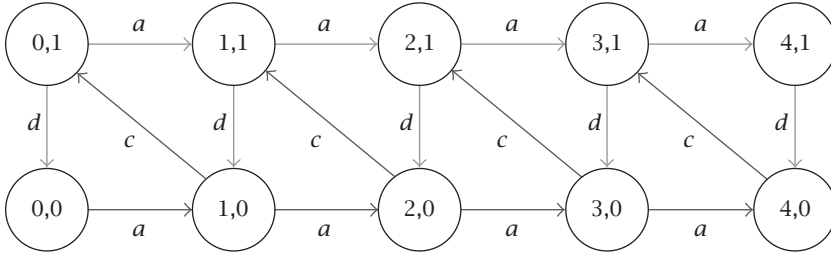
كانت الفكرة الرئيسية هنا هي نمذجة العملية الفيزيائية بفضاء حالة يُمثل جميع الحالات المُمكنة للنظام، وربطه بالتغيرات التي تُمثل معدلات التدفُّق العشوائية بين أزواج الحالات. وعن طريق استدعاء مبدأ توازن التدفُّق — مجموع التدفُّق الداخل إلى حالةٍ ما يساوي مجموع التدفُّق الخارج منها — حصل المهندسون على مجموعةٍ من المعادلات التي تربط نسب الوقت $p(s)$ التي تشغلها كل حالة s . ويمكنهم بعد ذلك حساب قيم $p(s)$ بتطبيق هذه المعادلات. هذا الشكل من النمذجة شائع جدًّا في نظرية الانتظار وتقييم أداء النظام لأن جميع المقاييس المهمة، مثل الإنتاجية وزمن الاستجابة واحتمالات التدفُّق الزائد، يمكن حسابها بسهولة من $p(s)$.



غرفة انتظار الطبيب

استخدم المهندسون أيضًا نماذج فضاء الحالة لإنشاء وحدات التحكم في النظم. في هذا المثال (انظر الشكل التالي)، ترغب الطيبة في إنشاء وحدة تحكم إلكترونية لمكتبها، تتكوّن من غرفة انتظار تتسع لأربعة أشخاص وغرفة علاج تتسع لشخص واحد. يدخل المرضى غرفة الانتظار ويجلسون. وبمجرد أن تُصبح الطيبة متاحة، تستدعي المريض التالي إلى غرفة العلاج. عند الانتهاء، يُغادر المريض من باب مُنفصل. ترغب الطيبة في أن تُضيء لمبة بيان في غرفة العلاج عندما يكون هناك مريض في غرفة الانتظار، وأخرى تضيء في غرفة الانتظار عندما تكون الطيبة مشغولةً بعلاج شخصٍ ما. يستخدم المهندس الذي يُصمّم وحدة التحكم نموذجًا حوسبيًا بالحالات (n, t) حيث $n = 0, 1, 2, 3, 4$ هو عدد المرضى الموجودين في غرفة الانتظار و $t = 0, 1$ هو عدد المرضى الموجودين في غرفة العلاج. تُطبق وحدة التحكم مُخطّط الحالة السابق. وتضيء لمبة البيان في غرفة العلاج كلما كان $n > 0$ ، وتضيء لمبة البيان في غرفة الانتظار كلما كان $t > 0$. يحدث الانتقال من حالةٍ إلى الأخرى

في ثلاثة أحداث: وصول المريض (a) ومغادرة المريض (d)، واستدعاء المريض من قبل الطبيب (c)، توجد أجهزة استشعار في الأبواب الثلاثة للمكتب للإشارة إلى هذه الأحداث.

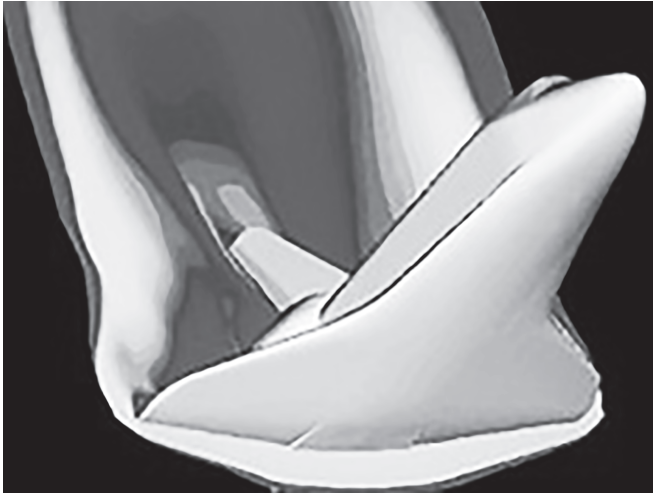
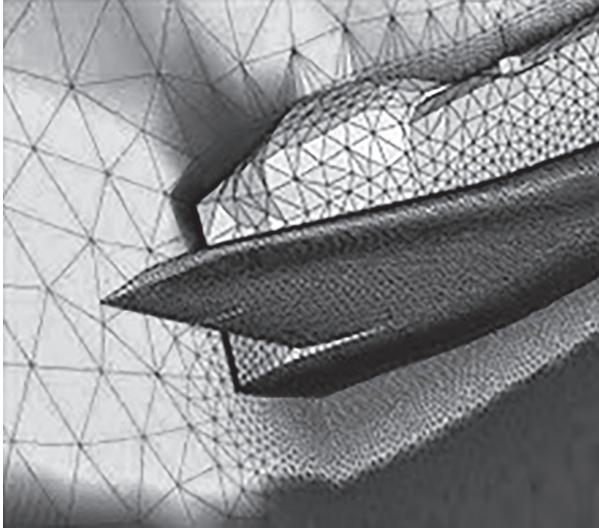


في هذه الحالة، لا يُستخدم النموذج لتقييم احتمالات حدوث الحالة، ولكن لتخطيط حالات الدائرة الإلكترونية والتنقل بينها. ومن الممكن بالتأكيد تفسير مخطط الحالة كما في المثال السابق، حيث a و b و c هي معدلات التدفق بين الحالات.

محاكاة الطائرات

يستخدم مهندسو الطيران محاكاة ديناميكيات السوائل الحوسبية لعمل نموذج لتدفقات الهواء حول الطائرة المقترحة. وقد أتقنوا ذلك لدرجة أنهم يستطيعون اختبار تصميمات الطائرات الجديدة دون أنفاق هوائية، وتصميمات مكوك الفضاء دون رحلات تجريبية. تتمثل الخطوة الأولى في إنشاء شبكة ثلاثية الأبعاد للمساحة المحيطة بالطائرة (انظر الشكل التالي). يكون تباعد نقاط الشبكة أصغر بالقرب من جسم الطائرة؛ حيث تكون التغيرات في حركة الهواء أكبر. ثم تُحوّل المعادلات التفاضلية لتدفق الهواء إلى معادلات الفرق على الشبكة، ويُنتج الكمبيوتر الفائق ملفات مجال تدفق الهواء وقوى الضغط الواقعة على كل جزء من الطائرة بمرور الوقت. وتُحوّل النتائج العددية إلى صور مُظَلَّلة (كما هو موضح في الشكل التالي) لتُصور أجزاء الطائرة التي تتعرض لأقصى معدلات الضغط.

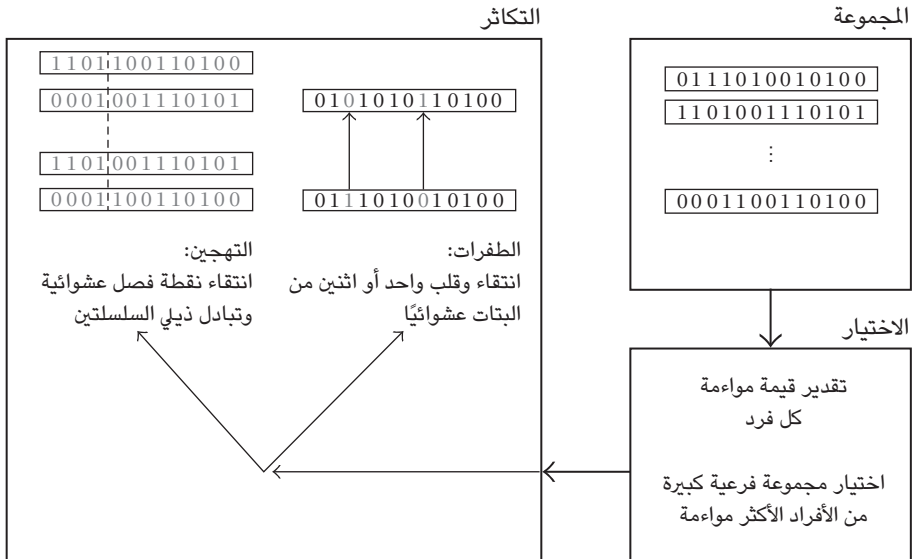
هذا الشكل من النمذجة شائع في العلوم. وتتم نمذجة العملية الفيزيائية على شكل معادلات تفاضلية تربط قيم العملية عند نقطة ما في الفضاء بقيم العملية عند نقاط الجوار. وتتم نمذجة الفضاء الذي ستدرس العملية فيه باستخدام شبكة. وتُستخدم معادلة الفروق لربط قيمة كل نقطة من نقاط الشبكة بقيم نقاط الجوار. ويحوّل



العرض الرسومي مجال القِيم على الشبكة إلى صورةٍ ملوّنة. ويمكن إعادة حوسبة الشبكة بأكملها للخطوة الزمنية التالية، مما يُعطي تصوّرًا متحرّكًا.

الخوارزميات الجينية

منذ الخمسينيات من القرن العشرين، أجرى علماء الوراثة تجارب محاكاة التطور البيولوجي باستخدام الكمبيوتر، ودرسوا كيفية انتقال السمات المختلفة وكيفية تطور مجموعة ما لتتكيف مع ظروفها. وفي عام ١٩٧٥، عدل جون هولاند فكرة هذه المحاكاة كأسلوب عام لإيجاد حلول شبه مثالية للمشاكل المعقدة في مجالات عديدة. تكمن الفكرة، كما هو موضح في المخطط التنظيمي في الشكل التالي، في إعداد مجموعة من الحلول المرجحة للمشكلة، مشفرة كسلاسل من وحدات البت. ويتم إيجاد قيمة كل سلسلة من وحدات البت بواسطة دالة ملائمة، ويُختار الأعضاء الأكثر ملاءمة من المجموعة للتكاثر عن طريق الطفرات والتهجين. تُعدل سلسلة وحدات البت بالطفرات عند قلب بت واحد أو عدة وحدات من البت منها عشوائيًا. ويُعدل زوج من سلاسل وحدات البت بالتهجين عن طريق تحديد نقطة فصل عشوائية وتبادل ذيلي السلسلتين. ويُنتج هذا مجموعة جديدة. وتُكرر العملية عدة مرات حتى لا يكون هناك المزيد من التحسينات في الأفراد الأكثر ملاءمة أو حتى تنضب ميزانية الحوسبة. وتُعد هذه العملية رائعة في إيجاد حلول شبه مثالية لمشاكل التحسين التي ما كان لحلولها المباشرة أن تكون ممكنة لولا ذلك.



(٥) التحديات الكبرى والمشكلات العصبية على الحل

لقد تغيّرت الحوسبة بشكلٍ كبير منذ نشوء النمذجة الحوسبيّة. في الثمانينيات من القرن العشرين، كان النظام المُضيف لنماذج التحديات الكبرى هو الكمبيوتر الفائق. أما اليوم، فإن النظام المُضيف هو «السحابة»؛ وهو نظام موزّع على نطاقٍ واسع للبيانات وموارد المُعالجة حول العالم. وتُتيح خدمات السحابة التجارية للمستخدمين استدعاء كميات هائلة من سعة التخزين والمُعالجة التي يحتاجونها عند الحاجة إليها. بالإضافة إلى ذلك، لم تُعد أجهزة الكمبيوتر مُقيدةً بالتعامل مع عمليات الحوسبة المحدودة؛ بمعنى أنها لم تُعد تبدأ مهمة الحوسبة المُحددة وتؤديها ثم تتوقف عن العمل. بل أصبحت الأجهزة الآن تستفيد من تدفّقاتٍ لا نهاية لها من البيانات وقوة المُعالجة حسب الحاجة، ويعتمد المُستخدمون على استمرار تشغيلها إلى أجل غير مُسمى. بالاستعانة بكل هذه القدرة الحوسبيّة العملاقة والرخيصة، يمكن لمزيد من الأشخاص أن يعملوا في التصميم الحوسبي ويُعالجوا مشكلات التحديات الكبرى.

ومع ذلك، هناك حدودٌ مهمةٍ لما يمكن أن تفعله كل هذه الإمكانيات الحوسبيّة. أحدها أن معظم الأساليب الحوسبيّة تركز على مهمةٍ بعينها؛ فهي جيدة جدًا في المهمة المُحددة التي صُممت من أجلها، ولكنها ليست بالجودة نفسها في المهام التي تبدو متشابهة معها. يمكن التغلّب على هذا في كثيرٍ من الأحيان بتصميمٍ جديدٍ يسدّ الفجوة الموجودة في التصميم القديم. ولنأخذ التعرف على الوجوه مثلاً. قبل عَقْدٍ من الزمان، لم تكن أساليب اكتشاف الوجوه والتعرف عليها في الصور جيدةً جدًا؛ وكان على الناس النظر إلى الصور بأنفسهم. أما اليوم، فقد استُخدمت خوارزميات التعلّم العميق (الشبكات العصبية) لتصميم برامج التعرف على الوجوه الآلية الموثوق بها للغاية، وسد الفجوة السابقة. تُدرّب هذه البرامج عن طريق عرض عددٍ كبيرٍ من حالات الصور المُسمّاة. ولكن برامج التعرف على الوجوه «ضعيفة»؛ بمعنى أن لا أحد يعلم كيف ستتصرّف الآلة عند تقديم مدخلاتٍ خارج المجموعات التي تدرّبت عليها. وحفّز ذلك علماء الحوسبة على النظر إلى الآلات التي تتعلّم دون مجموعات تدريب. ومن الأمثلة الحديثة على ذلك آلة تعلّمت ممارسة لعبة «جو» اللوحية من خلال التنافس مع آلات أخرى، وأصبحت في النهاية جيدةً بما يكفي لهزيمة أفضل لاعب «جو» في العالم في مباراة من خمس جولات.

أثار التعلّم الذاتي للآلات تخوفاً آخر؛ ألا وهو قابلية التفسير. يريد المُصممون والمستخدمون أن يعرفوا كيف توصلت الآلة إلى استنتاجها. إن فكرة أن الآلة يمكن أن

تصل إلى استنتاج تُصبح منطقية عندما ننظر إلى الخوارزميات على أنها إجراءات متتابعة؛ لأن النتيجة يمكن تفسيرها عن طريق فحص الخطوات المتبعة. ولكن عندما لا تكون الخوارزميات إجراءات متتابعة، كما في حالة خوارزميات التعرف على الوجوه وممارسة لعبة «جو»، فإن التفسير يُصبح مستحيلاً. فكل ما يوجد بداخلها هو كتلة مُعقدة من الروابط يصعب فهمها. إنها المشكلة نفسها التي نواجهها مع البشر؛ إذ كيف نفسر أسباب تصرّفاتنا؟ إذا سألنا مباشرة، فقد لا نعرف الإجابة، ومن المؤكد أنه لا يمكن اكتشافها بتشريح أدمغتنا. لا بدّ من وجود طرق أخرى لمعرفة متى يمكن الوثوق بالآلات ومتى لا يمكن الوثوق بها. ولا يزال التفكير الحوسبي المرتبط بتعلّم الآلة في مراحله الأولى.

هناك حدّ آخر لما يمكن القيام به باستخدام الإمكانيات الحوسبيّة، وهو وجود العديد من المشكلات التي لا يمكن حلّها بالحوسبة على الإطلاق. قدّمنا أمثلة في الفصل الثالث، وهي إما ليست حوسبيّة على الإطلاق، أو مُعقدة للغاية بحيث تتجاوز أي قدرة حوسبيّة يُمكننا توفيرها. ولكن التعقيد ليس العائق الوحيد. فهناك عائق آخر وهو أن بعض المشكلات تتجاوز بشكلٍ جوهري نطاق العلم والتكنولوجيا ولا يمكن حلّها بالأساليب العلمية والتكنولوجية. ومن هذه المشكلات «المشكلات العسية على الحل»؛ خاصة القضايا المتعلقة بتفاعل المجتمعات والتقنيات. يصعب حلّ هذه المشكلات عندما تمتلك الفصائل قوّة كافية لرفض أي اقتراح لا يُعجبها ولكنها لا تمتلك القوة الكافية لتحقيق التوافق في الآراء. وهناك العديد من الأمثلة: تُنتج ملايين السيارات «النظيفة» مُجمّعة ضباباً دخانياً غير صحّي في المدن المزدحمة. وتُعزز تكنولوجيا المعلومات الجديدة زيادة التفاوت في الدخل حيث يحصل المُصمّمون على دخل أكبر بكثير من المُستخدمين. ويكافح التعليم في مجالات العلوم والتكنولوجيا والهندسة والرياضيات لمعرفة كيفية إعداد الطلاب لمواجهة الحيرة بشأن مُستقبل العمل، وشبكات الأمان الاجتماعية، والتكنولوجيا، وتغيّر المناخ. حلول هذه المشكلات ليست علمية أو تقنية أو حوسبيّة بل سنصل إليها عن طريق التعاون الاجتماعي بين المجموعات التي تُقدّم الآن أساليب مُتنافسة ومتضاربة. وعلى الرغم من أن التفكير الحوسبي يمكن أن يساعد من خلال تصوّر الآثار الواسعة النطاق للأفعال الفردية، فإن التوافق الاجتماعي والعمل الاجتماعي وحدهما هما اللذان يُمكنهما حل المشكلات العسية على الحل.

يُعد التفكير الحوسبي قوة مؤثرة داخل العلم. فهو يركز على «الطريقة الحوسبيّة» لممارسة العلم ويجعل مُمارسيه مُصمّمين حوسبيين مهرة في مجالاتهم العلمية. إنه يجلب

تفسيرات جديدة للمعلومات في مجموعة متنوعة من التخصصات. ويقضي ممارسو التفكير الحوسبي في العلوم الكثير من وقتهم في نمذجة العمليات الفيزيائية، وتصميم أساليب حلّ لهذه العمليات، وتشغيل عمليات المحاكاة، وتصوّر النتائج.

الفصل الثامن

تعليم التفكير الحوسبي للجميع

تتمثل فكرتي الأساسية في أن البرمجة أقوى وسيلة لتنمية التفكير المتطور والدقيق اللازم للرياضيات والقواعد النحوية والفيزياء والإحصاء وجميع المواد «الصعبة». وربما أدرجت الفلسفة والتحليل التاريخي أيضًا في القائمة. باختصار، أؤمن أكثر من أي وقت مضى بأن البرمجة يجب أن تكون جزءًا أساسيًا من التطور الفكري للشباب.

سيمور بابيرت (بابيرت، ٢٠٠٥)

خلال فترة التسعينيات من القرن العشرين، كان تعليم التفكير الحوسبي في الغالب حكرًا على الجامعات؛ إذ لم يكن متاحًا في أي مكان آخر إلا نادرًا. وكانت لدى المدارس بداية من مرحلة رياض الأطفال وحتى المرحلة الثانوية مجموعة من دورات الكمبيوتر؛ تركز في معظمها على محو الأمية في مجال الكمبيوتر، ويُرَكِّز القليل منها على موضوعات في البرمجة. وجاءت نقطة التحول بعد عام ٢٠٠٠ عندما رأى الكثيرون مدى انتشار الحوسبة في العمل اليومي والحياة المنزلية. وبدأ مسئولو التعليم وصنّاع السياسات في الاتفاق على أن فهم آليات التحول الرقمي من المهارات المهمة في القرن الحادي والعشرين.

دخل مفهوم الخوارزمية الذي كان غامضًا من قبل إلى المحادثات اليومية حيث يُشيد الناس بالفوائد التي حصلوا عليها من الخوارزميات في عمليات البحث على شبكة الويب، وفي إعداد ضريبة الدخل، والتسوّق عبر الإنترنت، والجداول الإلكترونية، والمستندات المنسقة بدقة، والعروض التقديمية الجاهزة للعرض، والدورات المحوسبة، ثم لاحقًا في الهواتف الذكية، والشبكات الاجتماعية، وطلبات استدعاء السيارات، والإيجارات القصيرة

الأجل، والمواعدة والبحث عن الأصدقاء، وما إلى ذلك. ويبدو أن فهم الآلية العامة لكل شيء أصبحت مطلباً أساسياً للتعامل مع العالم الحديث. ولذا، فقد حان الوقت أخيراً لإدخال الحوسبة في التعليم بدءاً من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي.

(١) تعليم الحوسبة

كان إدخال تعليم الحوسبة في المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي يتطلب كفاً من نوع مختلف تماماً عن إدخال تعليم الحوسبة في الجامعات. فقد فشلت العديد من المشاريع التجريبية لإدخال أجهزة الكمبيوتر في المدارس نظراً إلى قلة أعداد المعلمين الذين لديهم أي خبرة في التعامل مع أجهزة الكمبيوتر وعدم توفير دعم سياسي قوي في مجالس إدارة المدارس. بحلول الثمانينيات من القرن العشرين بدأ تحول جذري، حيث حصل المزيد من الآباء والمعلمين على أجهزة كمبيوتر منزلية، وبدءوا يلمسون الأهمية المتزايدة للحوسبة في مجالات عملهم. وكانت دورات «محو الأمية في مجال الكمبيوتر» التي قُدمت في ذلك الوقت مُخبية للأمال بشكل عام من منظور التفكير الحوسبي؛ لأنها ركزت على استخدام أدوات مثل برامج معالجة الكلمات وجداول البيانات، وليس على البرمجة.

كان إدخال تعليم الحوسبة في المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي يتطلب كفاً من نوع مختلف تماماً عن إدخال تعليم الحوسبة في الجامعات. وكانت دورات «محو الأمية في مجال الكمبيوتر» ومن بعدها دورات «الإلمام بتكنولوجيا المعلومات» مُخبية للأمال بشكل عام. ولكن في عام ٢٠٠٦، بدأت حركة التفكير الحوسبي، التي حَمَسَت مسؤولي التعليم ومجالس إدارات المدارس على إدخال دورات الكمبيوتر في جميع مراحل التعليم ما قبل الجامعي.

في أواخر التسعينيات من القرن العشرين، وفي الوقت نفسه الذي بدأ فيه الإنترنت يُصبح سلعةً منزلية، اكتسبت حركة تعليمية جديدة زخمًا، حيث فضّلت «الإلمام بتكنولوجيا المعلومات» على «محو الأمية في مجال الكمبيوتر» ودعمت كِتَابًا دراسيًا شهيرًا يحمل الاسم نفسه. وقامت هذه الحركة على فكرة جذّابة مفادها أن الإلمام بلُغات البرمجة وممارسات الحوسبة سيكون داعماً قوياً في عالم التحول الرقمي الناشئ. وأن إدخال تعليم الحوسبة إلى المدارس سيؤدي إلى تمكين الأطفال من أن يُصبحوا مُستخدمين أذكياء لتكنولوجيا

الحوسبة، وتعريفهم بالقيود والمخاطر التي تكتنف العمليات الخوارزمية التي تكمن وراء الوظائف الناشئة، مثل: الشراء عبر الإنترنت والبحث على الإنترنت وخدمات الأخبار والاتصالات ووسائل التواصل الاجتماعي لاحقاً. وعلى الرغم من جاذبية الفكرة، لم تُسفر حركة «الإلمام بتكنولوجيا المعلومات» عن تغييرٍ واسع النطاق في تعليم الحوسبة في المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي.

ثم في عام ٢٠٠٦، اقترحت جانيت وينج أن التفكير الحوسبي هو ما يُريده الجميع؛ وليس محو الأمية في مجال الكمبيوتر أو الإلمام بتكنولوجيا المعلومات.¹ وقد أحدث رأيها دويماً قوياً. وفي السنوات القليلة التالية في المؤسسة الوطنية للعلوم بالولايات المتحدة، حشدت ٤٨ مليون دولار من الموارد وأقنعت العديد من الأشخاص بإدراج دورات الكمبيوتر في جميع مراحل التعليم من رياض الأطفال إلى التعليم الثانوي. وكان من بين نجاحاتهم الرئيسية جعل منظمات التعليم تُصدر تعريفات للتفكير الحوسبي والمناهج الدراسية المرتبطة به في المستويات الدراسية المختلفة، وتدريب المعلمين على مبادئ علوم الكمبيوتر، وبدء مجموعة جديدة من الدورات التمهيدية لمبادئ علوم الكمبيوتر في الجامعات، وتطوير مناهج وامتحانات جديدة لتحديد المستوى المتقدم لربط المدارس الثانوية بهذه الدورات التمهيدية الجديدة. وهكذا أصبح التفكير الحوسبي هو التوجّه السائد.

ولكن كما أشرنا سابقاً، لم يكن هذا النجاح سهلاً. فعادةً ما كانت مجالس إدارات المدارس في مؤسسات التعليم ما قبل الجامعي من رياض الأطفال وحتى التعليم الثانوي تتحفّظ على إضافة مناهج دراسية للحوسبة في مدارسها. أحدثت حركة التفكير الحوسبي تحولاً في تفكير العديد من مجالس إدارات المدارس. فبدون هذه الحركة، لم نكن لننحدث عن التفكير الحوسبي في التعليم ما قبل الجامعي بالقدر الذي نتحدث به اليوم. وسوف نُوضح، في هذا الفصل، تطوّر تعليم الحوسبة بوصفه سلسلة من الموجات التي بدأت بنوع التفكير الحوسبي المُتاح في خمسينيات القرن العشرين (تحويل المشكلات إلى خوارزميات وحل المشكلات الرياضية)، ثم انتقلت إلى «عصف العقول» لبابيرت، ثم إلى محو الأمية في مجال الكمبيوتر والإلمام بتكنولوجيا المعلومات، ثم وصلت في الآونة الأخيرة إلى نسخة حديثة من التفكير الحوسبي مُصمّمة للأطفال في المدارس.

(٢) أدوات تفكير ذات أغراض عامة

بدأ التعليم الأكاديمي لآلات الحوسبة الذاتية التشغيل في أواخر الأربعينيات من القرن العشرين، عندما بدأ رؤاد الحوسبة ببرامج تعليمية حول الأساليب العددية لحلّ المشكلات

باستخدام آلات الحوسبة الواسعة النطاق. انتشرت هذه الجهود المبكرة في خمسينيات القرن الماضي عندما تطلّب الإنتاج الضخم لأجهزة الكمبيوتر ذات البرامج المخزّنة عددًا كبيرًا من الأشخاص القادرين على برمجتها. بعد الدخول المبكر للشركات الخاصة، بدأ المدرسون الجامعيون في تنظيم مؤتمراتٍ لمناقشة تعليم الحوسبة في منتصف الخمسينيات من القرن العشرين. وبحلول عام ١٩٦٠، قدمت حوالي ١٥٠ جامعة أمريكية تدريبًا في مجال الحوسبة. ومع ذلك، لم يكن هناك رأيٌ موحدٌ حول ما يحتاج الناس إلى معرفته عن الحوسبة؛ تعتمد البرامج الفردية على الخصوصيات المحلية مثل الوظائف المحددة واحتياجات الشركات والأجندات الشخصية لأعضاء هيئة التدريس الخاصة وعقود البحث واهتمامات أصحاب المصلحة الآخرين.²

حتى في تلك الأيام الأولى، وصف بعض مُعلّمي الحوسبة رؤاهم للحوسبة بوصفها «أداة تفكير لأجل التعلُّم»؛ أي أداة للتعامل مع المشكلات والأسئلة في العديد من المجالات إلى جانب علوم الكمبيوتر. كان آلان برليس، الذي أسس قسم علوم الكمبيوتر في جامعة كارنيجي ميلون، مناصرًا صريحًا لهذه الرؤية. وقال إن الحوسبة ستؤتمت العمليات في العديد من المجالات، وإن الأشخاص في تلك المجالات سيُحوّلون المشكلات إلى خوارزميات. وكان يقصد بذلك التفكير في المشكلات وتطوير حلولٍ حوسبيّة لها. واستشهد جورج فورسايت ببرليس في خطابه عام ١٩٥٨ لدى جمعية الرياضيات الأمريكية: «على الرغم من أننا نعتقد أننا نعرف شيئًا ما عندما نتعلّمه، ونقتنع بأننا نعرفه عندما نستطيع تعليمه، فإن الحقيقة هي أننا لا نُدرّكه حقًا حتى نتمكن من تحويله إلى نصٍّ برمجي نُدخّله إلى جهاز كمبيوتر آلي!» وبعد عقد من الزمان، ردّد فورسايت الادعاء بأن الحوسبة توفّر أدواتٍ ذهنية عامة الأغراض تخدم الشخص مدى الحياة. كان كلٌّ من برليس وفورسايت يؤمنان بأن جميع الأشخاص في جميع المجالات سيستفيدون من تعلّم الطرق الإجرائية للقيام بالأشياء حوسبيًا. وكانا يعتقدان أن النماذج الحوسبيّة ستكون مُفيدة في جميع المجالات.

ازداد طموح رؤى تعليم الحوسبة فيما سيتمكّن التفكير الحوسبي من تحقيقه. جادل مارفين مينسكي، وهو رائد مشهور في مجال الذكاء الاصطناعي، في خطابه بمناسبة حصوله على جائزة تورينج عام ١٩٦٩، بأن الحوسبة ستتفوّق على الرياضيات من حيث الأهمية للتعليم المبكر.³ جادل دونالد كنوث، وهو رائد في فهم الخوارزميات، بأن تعليم الكمبيوتر القيام بشيءٍ ما يُجبره على الدقة ويؤدي إلى فهمٍ أعمقٍ مقارنةً بالوسائل

التقليدية للتفكير.⁴ ورأى رائد آخر أن الخليفة الحديث لـ «الشخص الكلاسيكي» سيكون «شخص تورينج».⁵ وكتبَ اثنان من مُعلمي الحوسبة المشهورين أن المعرفة الإجرائية للحوسبة تخلق ثورةً في كيفية تفكير الناس وتعبيرهم عن أنفسهم.⁶ وقد تبين أن كل هذا التفاؤل بشأن قدرة مهارات الحوسبة على حلّ المشكلات العامة كان سابقاً لأوانه، كما سنناقش لاحقاً.

(٣) التفكير الحوسبي ليس قابلاً للنقل بسهولة

ركزت الموجة الأولى لإدخال التفكير الحوسبي في مراحل التعليم ما قبل الجامعي على البرمجة. ففي منتصف الستينيات من القرن العشرين، حصلت بعض المدارس الثانوية الأمريكية على أجهزة كمبيوتر «بي دي بي-٨» الصغيرة من إنتاج شركة «ديجيتال إكوبمينت كوربوريشن»، ونظّم المُعلمون المغامرون دوراتٍ حولها. وأدت العديد من المبادرات لاستخدام أجهزة الكمبيوتر في المدارس خلال الستينيات والسبعينيات من القرن العشرين إلى بعض الابتكارات البارزة. على سبيل المثال: طُرِح جهاز «ليتل مان كمبيوتر» لتدريس لغات الآلة وأجهزة الكمبيوتر للطلّاب في عام ١٩٦٥، وطُرِحت إحدى لغات البرمجة المبكرة للأطفال، وهي لغة «لوجو»، في عام ١٩٦٧، وظهر مفهوم دينابوك الشهير، وهو جهاز كمبيوتر محمول للأطفال، في عام ١٩٦٨. وعلى الرغم من انتشار أجهزة الكمبيوتر الصغيرة وبعض أجهزة الكمبيوتر الدقيقة في أواخر السبعينيات، فإن المُعلمين، الذين يفتقرون إلى الموارد المالية والإرادة السياسية، لم يتمكنوا من تحويل الدورات التجريبية إلى خطة واسعة النطاق في المدارس.

برزت لغة البرمجة لوجو بين العديد من المبادرات في ستينيات القرن العشرين. ولم تكن لغة برمجة مستقلة عامة. بل كانت جزءاً من إطار متكامل من الأفكار التربوية والتكنولوجية والتعليمية التي صمّمها سيمور بابيرت بناءً على فهمه العميق لكيفية تعلّم الأطفال. وكان كتابه «عَصَف العقول» الصادر عام ١٩٨٠، والذي كتبه بعد عقد من دراسة لغة لوجو وتجربتها، علامة فارقة في تعليم الحوسبة وتدريس التفكير الحوسبي. وصاغ بابيرت عبارة «التفكير الحوسبي» لممارسة التفكير الإجرائي الذي علّمه للأطفال. وكان يرى أن التعلّم يكون أكثر فعاليةً عندما «يبنى المتعلمون المعرفة»؛ فهم يكتسبون الممارسات من خلال الانغماس في عالم الممارسات. إنهم يبنون معرفتهم من ممارستها بدلاً من أن تُملى عليهم. وأصبحت نظرية التعلّم البنائية شائعة جداً في التعليم. واستمر بابيرت

في الدفاع عن التعلُّم الموجَّه ذاتياً والتعلُّم القائم على المشاريع والعروض التقديمية الهادفة والتعليم القائم على التيسير واستخدام التكنولوجيا لدعم التعلُّم في الفصول الدراسية. وأثرت أفكاره في شركة ليجو لتصميم وتسويق مكعبات الأطفال القابلة للبرمجة والتي تُسمَّى «ليجو مايندستورمس».

يُعَد تعليم مهارات التفكير الحوسبي الأساسية، مثل البرمجة ونمذجة الكمبيوتر، أصعب بكثيرٍ من تعليم جداول البيانات ومعالجة النصوص وغيرها من أدوات تطبيقات الحوسبة. وعلى الرغم من شهرة البنائية، كان من الصعب تسويق الفكرة الجوهرية في كتاب «عَصَف العقول» — ألا وهي التحوُّل من «تعلُّم البرمجة» إلى «البرمجة للتعلُّم» — بين المعلمين. كيف يمكننا نشر تعليم التفكير الحوسبي دون وجود عددٍ كافٍ من المعلمين الراغبين؟ هل يمكننا الاعتماد على مجموعة أصغر من المعلمين المهتمين لتعليم الجميع رافقاً الأمل في أن يتمكن عدد قليل من المعلمين من تعليم التفكير الحوسبي للجميع فرضية الانتقال. وترى هذه الفرضية أن التفكير الحوسبي مهارةٌ ما وراء معرفية يتعلَّمها الطلاب من البرمجة، ويصبح الطلاب الذين يتعلَّمون التفكير الحوسبي في مجالٍ ما أفضل في حلِّ المشكلات في مجالاتٍ أخرى أيضاً. وقد عزَّز هذه الفرضية الرأي القائل بأن تعليم الحوسبة يجب أن يكون عنصراً أساسياً في التعليم ما قبل الجامعي. وقدَّم أكثر المؤيدين المتحمسين للفرضية ادعاءاتٍ مثل «مفهوم الإجراء هو السر الذي يبحث عنه المعلمون منذ فترة طويلة»، و«القيمة التربوية للنهج الخوارزمي تساعد في فهم شتى أنواع المفاهيم». ورأوا أن تعليم البرمجة يُحسِّن مهارات التفكير العامة مثل التفكير المنطقي و«يشحذ العقل» بشكلٍ عام. ستكون فرضية الانتقال مهمة حقاً إذا ثبتت صحتها.

أشار منتقدو فرضية الانتقال إلى قاعدةٍ بحثية في علم الإدراك النمائي، وتعلَّلوا بأنه لا يوجد دليل على انتقال المهارات من البرمجة إلى مجالاتٍ أخرى. فالأبحاث التي أُجريت على البالغين لم تدعم انتقال المهارات المعرفية بين المجالات. والبرمجة نفسها هي شبكةٌ مُعقدة من المهارات تتضمن القدرات الرياضية والمنطق الشرطي والمنطق القياسي والتفكير الإجرائي والمنطق الزمني وسعة الذاكرة. ولم يكن من الواضح أي أجزاء من هذه المهارات تنتقل وأيها لا تنتقل. وبعد الكثير من عمليات التحقق التفصيلية، خلَّص الباحثون التربويون في النهاية إلى أنه لا يوجد دليل كافٍ لقبول فرضية الانتقال. ومن ثمَّ لم يكن من الممكن اعتبارها مبرراً مقنعاً لتعليم الحوسبة في المدارس.

بما أن فرضية الانتقال غير مقبولة، فإن المدارس تحتاج إلى المزيد من المُعلِّمين الذين يفهمون التفكير الحوسبي ويُمكنهم تدريسه في سياقاتٍ مختلفة. والقليل فقط من المُعلِّمين لديهم دراية جيدة بأجهزة الكمبيوتر بالقدر الذي يكفي للقيام بهذه المهمة. ربما يُمكنهم تدريس دورة محو الأمية في مجال الكمبيوتر، ولكن هذه الخطوة البسيطة لن تؤهِّلهم لتعليم التفكير الحوسبي. وفي منتصف العَقد الأول من القرن الحادي والعشرين، عندما بدأت المؤسسة الوطنية للعلوم بالولايات المتحدة في دعم تدريب المزيد من مُعلِّمي الحوسبة، بدأت أعداد المُعلِّمين المؤهَّلين في التراجع.

(٤) من محو الأمية إلى الإلمام بتكنولوجيا المعلومات

كان من شأن المُدافعين الأوائل عن التفكير الخوارزمي أن يشعروا بالذهول من كثرة دورات «محو الأمية في مجال الكمبيوتر» في الثمانينيات والتسعينيات التي ركَّزت على كيفية استخدام تطبيقات سطح المكتب، مثل مُعالجات النصوص وجداول البيانات ولوحات الرسم. وقد وجد الطُلاب والمُعلِّمون المُتحمِّسون أن هذه الدورات مُملة. كان محو الأمية باستخدام برامج سطح المكتب بعيدًا كل البُعد عن طموحاتهم للمشاركة في ثورة الكمبيوتر وتشكيلها. وعرضت الجمعيات المهنية، التي كانت من بينها جمعية آلات الحوسبة وجمعية الكمبيوتر التابعة لمعهد مهندسي الكهرباء والإلكترونيات والجمعية البريطانية للكمبيوتر، مساعدتها على مسؤولي التعليم ما قبل الجامعي في تطوير دورات كمبيوتر أشمل وأعمق، لكنها لم تحضُل على الكثير من الدعم. وفي عام ١٩٩٩، طالبت إحدى لجان مجلس البحوث الوطني الأمريكي برفع مستوى تعليم الكمبيوتر في المدارس ليُصبح أكثر تحدِّيًا ويتجاوز محو الأمية في مجال الكمبيوتر إلى الإلمام بتكنولوجيا المعلومات. قدَّم الإلمام بتكنولوجيا المعلومات قدراتٍ ومفاهيم ومهارات أساسية لبعض مستويات التفكير الحوسبي. واقتُرنت مبادرة مجلس البحوث الوطني الأمريكي بكتاب دراسي بعنوان «الإلمام بتكنولوجيا المعلومات» والذي أصبح شائعًا جدًّا بين معلمي المدارس الثانوية.

أدخلت العديد من المدارس الحوسبة في مناهجها الدراسية لأسبابٍ عملية تتعلَّق بالاستجابة إلى مطالب الآباء ومجالس إدارات المدارس. وسعوا إلى الوصول إلى عمليات المحاكاة والبرامج التعليمية الأخرى، والوصول إلى البرمجة الأساسية، والمشاركة في ثورة الإنترنت، وتعلُّم مهارات القرن الحادي والعشرين، والتحضير للعمل في مجالات العلوم

والتكنولوجيا والهندسة والرياضيات، والمشاركة الاجتماعية الموسعة، وكذلك وسيلة جديدة للأطفال للتعبير عن الإبداع الفردي.⁷ كان المعلمون والآباء مُستعدين لهذه الأهداف؛ لأنهم اعتقدوا أن تعلُّم البرمجة يُعلم مهارات مهمة لا تُعلمها أي مادة أخرى، ولأنهم لم يرغبوا في أن يكون أطفالهم غير مؤهلين في عالم يعتمد بشكل متزايد على المهارات في مجال تكنولوجيا المعلومات والاتصالات.

في مُستهل القرن الحادي والعشرين، كان دخول البرمجة والتصميم الحوسبي إلى المدارس أسهل أيضاً بسبب التقدُّم في منهجية البرمجة والتكنولوجيا والتغيُّرات التي طرأت على ما يحتاج المبرمجون المبتدئون إلى معرفته. كانت لغات جديدة، مثل بايثون، أسهل بكثير في الاستخدام وأخفت التفاصيل الأساسية لنظام التشغيل والأجهزة. وكانت واجهات الاستخدام الرسومية التي تتميز بسمة السحب والإفلات ناجحة للغاية. كما ساعدت الأدوات الفعالة في أتمتة أجزاء كبيرة من عملية البرمجة.

مع كل هذه التطورات في لغات البرمجة والأدوات والأساليب، أضحت البرمجة في متناول المزيد من الطلاب والمُعلمين عن أي وقت مضى. وزادت فرص الإلمام بالحوسبة. ولكن رغم ذلك كُلُّه، لم يكن لدى العديد من المدارس في عام ٢٠١٠ أي دورات كمبيوتر أو مناهج لتحديد المستوى المُتقدم في الحوسبة. ولم يكن الإلمام بتكنولوجيا المعلومات قوياً بما يكفي ليُصبح قوة دافعة.

(٥) إحياء التفكير الحوسبي

أحدث مقال جانيت وينج عام ٢٠٠٦ حول التفكير الحوسبي موجة جديدة في الاتجاه إلى توفير دورات الحوسبة لجميع الطلاب في مراحل التعليم ما قبل الجامعي. وأحدث مصطلح «التفكير الحوسبي» صدًى وحثَّ على العمل رغم عدم نجاح محو الأمية في مجال الكمبيوتر والإلمام بتكنولوجيا المعلومات في ذلك. وحشدت وينج موارد كبيرة في المؤسسة الوطنية للعلوم بالولايات المتحدة لحثِّ عدد كبير من الباحثين على إجراء أبحاثٍ ودراساتٍ حول إدخال التفكير الحوسبي في التعليم، وتدريب عدد كبير من المُعلمين على تدريس التفكير الحوسبي، وعلى حشد المنظمات الخاصة لإنتاج توصيات المناهج الدراسية للتفكير الحوسبي في مراحل التعليم ما قبل الجامعي، وتطوير مناهج وامتحانات لتحديد المستوى المُتقدم في مبادئ الحوسبة. وأصبح مقال وينج واحداً من أكثر المقالات التي يُستشهد

بها في تعليم الحوسبة، وأضحى نقطة تحوُّل في حركة عالمية لإدراج التفكير الحوسبي في مراحل التعليم ما قبل الجامعي.

طورت منظمات كبرى؛ منها جمعية مُعلمي علوم الكمبيوتر والمنظمة البريطانية للحوسبة في المدارس ومنظمة code.org وهيئة المناهج الدراسية والتقييم والتقارير الأسترالية، وأوصت بإطار عمل للمناهج الدراسية لتعليم التفكير الحوسبي في مراحل التعليم ما قبل الجامعي. وروَّجت هذه المنظمات لنوادي التشفير ومُخيمات التشفير والحركة الدولية التي تسمى «ساعة الترميز». وأصبح التفكير الحوسبي كلمة بحث أساسية تحصل على مئات الآلاف من النتائج في الأخبار والمدونات وفصول الكتب والمشاريع البحثية والمقالات حول تعليم الحوسبة.

أدَّى التدفُّق السريع للعديد من الوافدين الجُدد المُحمسين الذين لم يكونوا على دراية بالتاريخ الطويل السابق للتفكير الحوسبي إلى التباس تعريفات التفكير الحوسبي وأهداف تعلُّمه. وابتكر البعض إطارات عمل جديدة لمناهج التفكير الحوسبي في المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي، مما جعلهم يُعيدون صياغة أفكار نُوقِشت لعقود من الزمان، ويحذفون أفكارًا مهمَّة، ويخلطون التفكير الحوسبي باستخدام التطبيقات، ويُدرجون بعض المفاهيم الخاطئة حول الحوسبة والخوارزميات. وأدَّى هذا إلى خلافاتٍ متنوعة بين المجموعات المختلفة التي تستخدم التفكير الحوسبي.⁸ وفيما يلي بعض من نقاط الخلاف الأكثر شيوعًا، التي قد يكون سببها الاختلاف بين التفكير الحوسبي للمبتدئين والتفكير الحوسبي للمُحترفين — فالتفكير الحوسبي الأساسي في مراحل التعليم ما قبل الجامعي يختلف بالتأكيد عن التفكير الحوسبي المُتقدِّم في التعليم العالي — وكذلك الاختلاف في سياقات التطبيق:

- (١) هل التفكير الحوسبي يقتصر على التفكير في آليات إنشاء الخوارزميات أم يشمل التفكير في الآلات وعلوم الحوسبة وهندسة البرمجيات والتصميم؟
- (٢) هل التفكير الحوسبي يدور في الغالب حول البرمجة، أم يشمل أيضًا النظم والشبكات والهيكل، أم لا يتعلَّق حقًّا بأيٍّ منها؟
- (٣) هل تعريف التفكير الحوسبي بأنه صياغة خوارزميات لحلّ المشكلات يُعتبر رؤية ضيقة ومحدودة لنطاق التفكير الحوسبي؟
- (٤) هل الخوارزميات هي فقط تلك التي ينطبق عليها التعريف الصارم من نظرية الحوسبة، أم يمكن تعريفها أيضًا تعريفًا أكثر مرونة؟

- (٥) هل الخوارزميات تتضمن بالضرورة آلة مجردة في الخلفية؟
- (٦) هل الخوارزميات هي في الأساس توجيهات للتحكم في الآلات، أم وسيلة للتعبير عن الإجراءات؟
- (٧) هل استخدام أدوات الحوسبة يُعلّم التفكير الحوسبي؟
- (٨) هل تنفيذ الإجراءات اليومية خطوة بخطوة يُعدّ مظهرًا من مظاهر التفكير الحوسبي؟
- (٩) هل التفكير الحوسبي يتعلّم المرء من ممارسة البرمجة، أم من أنشطة التعلّم المُصمّمة جيدًا التي تعتمد على نظام الخطوات والقواعد؟
- (١٠) هل تعلّم التفكير الحوسبي في سياق الحوسبة ينتقل إلى مهارات حلّ المشكلات في مجالاتٍ أخرى؟
- (١١) هل التفكير الحوسبي يعتمد على المجال، أم أنه مهارة عُليا تصلح في جميع المجالات؟
- (١٢) هل عمليات الحوسبة موجودة في الطبيعة، أم تقتصر على الخوارزميات والآلات؟
- (١٣) هل معالجة المعلومات بواسطة أجهزة الكمبيوتر تختلف عن معالجة المعلومات التي يقوم بها البشر، وهل «أدوات معالجة المعلومات» يمكن أن تشمل أشياء مثل الجزيئات أو الحمض النووي أو الكواركات؟
- (١٤) هل ينبغي تقييم تعلّم الطلاب من خلال البرهنة على مهاراتهم في تصميم عمليات الحوسبة، أم من معرفتهم بمفاهيم رئيسية مُعينة؟
- (١٥) هل ينبغي أن يكون رضا العملاء عن أداء البرنامج جزءًا من تقييم نجاح البرنامج؟
- (١٦) هل تعليم التفكير الحوسبي في المدارس مُلزم بالالتزام بتعريفات صارمة للحوسبة، أم أنه يمكن لأسباب عملية وتربوية أن يكون أكثر مرونة وتحررًا؟

لقد عبّرنا عن موقفنا بشأن هذه الأسئلة على مدار الكتاب. نحن نرى التفكير الحوسبي بوصفه ممارسة بشرية قديمة وغنيّة جرى تحسينها في العصر الحديث للكمبيوتر الإلكتروني. ونرى التفكير الحوسبي بوصفه مهارة عقلية للتفكير في تصميم جميع أنواع عمليات الحوسبة، وهي مهارة يمكن صقلها وتحسينها من خلال الممارسة والخبرة الواسعة في المستويات المتقدمة. ونرى العديد من المستويات والأساليب المختلفة للتفكير الحوسبي بدءًا من مهارات الحوسبة وخبراتها الأساسية وصولًا إلى المهارات

الْمُتَقَدِّمَة والمتخصّصة للغاية. ونرى أن هناك العديد من الطرق الجيدة لتعليم التفكير الحوسبي على مستوى المُبتدئين. ونرى أن الهدف النهائي من التفكير الحوسبي هو إنشاء حلولٍ يمكن تنفيذها بواسطة الآلات. ونرى أن التفكير الحوسبي يعتمد في الغالب على المجال؛ على سبيل المثال: يختلف تفكيرك في عمليات الحوسبة في علم الأحياء عنه في الفيزياء أو الكيمياء أو العلوم الإنسانية. ونرى أن فكرة أن التفكير الحوسبي هو قدرة بشرية فطرية تُمارَس يوميًا عن طريق استخدام أدوات الحوسبة وتنفيذ الإجراءات اليومية الروتينية، هي فكرة غير واقعية. كذلك نرى أن محاولة تعريف الخوارزميات على أنها مجموعة من الخطوات التي يُحتمَل أن تكون غامضةً والتي تُحلُّ بواسطة الحواسيب البشرية هي سوء فهم للحوسبة.

نودُّ أن نشير إلى حركة أخرى لإدخال الحوسبة إلى المدارس بدايةً من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي. تُعرف الحركة باسم «العلوم الحاسوبية من دون كمبيوتر»⁹ وهي تسعى إلى تعليم مفاهيم الحوسبة وممارساتها من خلال الألعاب والخدع والأنشطة المُختلفة. وقد تأسَّست في أواخر التسعينيات من القرن العشرين على يد تيم بيل ومايكل فيلوز وإيان ويتن. واكتسبت شعبية عالمية وأثَّرت في تصميم توصيات المناهج الدراسية لتعليم الحوسبة في المدارس من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي من قِبَل جمعية آلات الحوسبة ومنظمة code.org.

باختصار، نحن نرى المجال رحبًا لاستخدام نهجٍ واسعٍ وتعدُّدي لتعليم التفكير الحوسبي مع الحفاظ على الالتزام بطُرق التفكير والممارسة الحوسبيَّة المحددة. ونأمل في المقام الأول أن يُعطي جميع معلمي الحوسبة لطلابهم إحساسًا بثراء الأبعاد العديدة للحوسبة وجمالها.

الفصل التاسع

مستقبل الحوسبة

التكنولوجيا جزءٌ من حضارتنا. يتحدث الناس أحياناً عن الصراع بين البشر والآلات، ويُمكنك رؤية ذلك في الكثير من روايات الخيال العلمي. لكن الآلات التي نبنتها ليست غزواً من المريخ. نحن نبنت هذه الأدوات لتوسيع نطاق إمكانياتنا وقدراتنا.

راي كريساويل (٢٠١٣)

التفكير الحوسبي هو سعيٌ مُستمر نحو إدراك عمليات وأساليب التفكير المستخدمة في الحوسبة ومحاكاة هذه العمليات والأساليب. وهو في حالة تدفق وتجدد مستمرين. وعلى الرغم من أن العديد من المبادئ الأساسية للتفكير الحوسبي قديمة للغاية، فإن تطوُّر ممارسات الحوسبة وحالة التقدُّم التكنولوجي قد أثرت في كيفية رؤيتنا للتفكير الحوسبي وما هو أساسي فيه. على سبيل المثال: تجعل مجموعات تطوير البرامج المتجددة باستمرار واللُّغات الجديدة والخدمات السحابية مهام التصميم الحوسبي تتحوَّل بعيداً عن عمليات البرمجة البدائية نحو مستوياتٍ أعلى وأعلى من التجريد؛ مما يجعل وظائف الحوسبة أكثر تركيزاً على التصميم. إن البرمجة التقليدية في سبيلها إلى فقد دورها كواجهة أساسية لعمليات الحوسبة، وحلَّت بدلاً منها الأدوات الذكية والأدوات المحددة لكل مجالٍ والتي تُتيح لمزيد من المستخدمين الاستفادة من إمكانيات الكمبيوتر دون برمجته. إن التفكير الحوسبي يتجاوز البرمجة وتطوير البرمجيات.

سنناقش بعض القوى التي تُشكل عالمنا وتأثيراتها المحتملة في كيفية رؤيتنا للحوسبة وتفكيرنا فيها. وسنناقش أيضاً بعض الأسئلة المهمة التي لا يمكن للتفكير الحوسبي مساعدتنا في إيجاد إجابات لها. فالتفكير الحوسبي له حدوده.

(١) نماذج حوسبيّة جديدة

من أكثر الأسباب الواضحة لتغيّر التفكير الحوسبي أن تقنيات الحوسبة تتغير. فطوال الفترة التي آمناً فيها بقانون مور للشرائح المصنوعة من مادة السيليكون، ظلت البنية الأساسية للشرائح في أجهزة الكمبيوتر والأجهزة المحمولة وفيّة لتصميم فون نيومان الذي يرجع إلى عام ١٩٤٥، والمكوّن من وحدات ذاكرة ومعالجة مُنفصلة، مع معالجٍ ينفذ التعليمات المخزنة في الذاكرة خطوةً بخطوة. ويأتي مفهوم «الخطوات الحوسبيّة» في التعريفات الحديثة للتفكير الحوسبي من هذا التصميم، وكذلك من تعريفات آلان تورينج للحوسبة.

ولكن لا يمكن أن يظلّ الاستناد إلى قانون مور قائماً بسبب طبيعة مادة السيليكون وطبيعة عملية صنع الشرائح.^١ لهذا السبب، كان الباحثون يبحثون عن تقنيات جديدة يُمكنها أن تحلّ محلّ التقنيات المعتمدة على بنية فون نيومان والقائمة على السيليكون، وتواصل النمو الأسّي لمعدل سرعة معالجة المعلومات. ومن التقنيات الأساسية المرشحة، أجهزة الكمبيوتر الكمية، والشبكات العصبية، وأجهزة الكمبيوتر القائمة على الحوسبة الانعكاسية، وأجهزة الكمبيوتر القائمة على حوسبة الحمض النووي، وأجهزة الكمبيوتر المُستمتلة على مقاومات الذاكرة، وغيرها. وتُحدّد كل تقنية من هذه التقنيات نموذجاً حوسبياً جديداً هو الهدف للمُصمّمين.

فلنأخذ مثلاً على ذلك كمبيوتر الموجة «دي»؛ وهو كمبيوتر كمي تجاري.^٢ هذا الكمبيوتر مُصمّم لحل مجموعة من المعادلات، معروفة في الفيزياء باسم نموذج إيزينج، والذي يصف كيف تستقرّ بعض النظم في حالات طاقةٍ دُنيا. وتعني «برمجة» جهاز الموجة «دي» تشفير المسألة في صورة مجموعة من معادلات إيزينج وإدخال المعاملات في الآلة؛ ويعني «التنفيذ» السماح للآلة بالاستقرار في حالة طاقةٍ دُنيا تتوافق مع حلّ المعادلات (عدد قليل من الميكروثانية)؛ وتتمثل عملية القراءة من الذاكرة في قراءة وحدات البت الكمية للآلة وتفسيرها على أنها الإجابة. لا يوجد هنا مفهوم «مجموعة التعليمات» أو البرمجة بمعنى «تصميم سلسلةٍ من الخطوات». ويشعر مُعظم علماء الكمبيوتر بالارتباك عند تعليمهم كيفية إعداد كمبيوتر الموجة «دي» لأول مرة؛ إذ تختلف عملية البرمجة هذه عن عملية البرمجة التي عرفوها طوال حياتهم المهنية.^٣ أما علماء الفيزياء المُدرّبون، فيواجهون صعوبة أقل بكثيرٍ في فهم الآلة. ولا يستطيع التعريف الحالي للتفكير الحوسبي

— وهو صياغة مسألة بحيث يمكن حلُّها كسلسلةٍ من الخطوات الحوسبيَّة — وصف التفكير الحوسبي الذي تقوم به هذه الآلة.

حوسبة الحمض النووي (الذي إن إيه) هي تقنية أخرى تجري دراستها في الوقت الحالي. في عام ١٩٩٤، أجرى الباحثون تجربة شَفَّروا فيها المسارات المُحتَمَلة في إحدى الخرائط على هيئة سلاسل الحمض النووي، ثم استخدموا الطُّرُق الكيميائية في ذلك الوقت لتطوير الخليط الأوَّلي إلى خليطٍ تُمثِّل فيه الغالبية العُظمى من السلاسل أقصر مسار في الخريطة.⁴ أُحرِزَ تقدُّمٌ كبير في هذه التقنية. وفي عام ٢٠١٦، استخدم فريق بحث آخر تقنية تعديل الجينات «كريسبر» الحديثة لإدخال صورةٍ في الحمض النووي لبكتيريا. ويواجه علماء الكمبيوتر المُدرَّبون على التفكير من حيث الخطوات الحوسبيَّة صعوبة أكبر من المُختصِّين في علم الأحياء الجزيئي في فهم آلية عمل حوسبة الحمض النووي.

توضح هذه الأمثلة أن التفكير الحوسبي قد توسَّع إلى ما هو أبعدُ من فكرة حلِّ المشكلات بالخطوات الحوسبيَّة. ولذا، أصبح تعريفنا الأوسع للتفكير الحوسبي — تصميم عمليات الحوسبة التي تجعل أجهزة الكمبيوتر تؤدي المهام بالنيابة عنا، بالإضافة إلى شرح العالم وتفسيره كمجموعةٍ مُعقدة من العمليات المعلوماتية — هو الأقرب إلى الصواب.

(٢) التصميم

تُعَدُّ الزيادة المُستمرة في أهمية التصميم سبباً آخرَ لتغيُّر التفكير الحوسبي. لم يُعَدِّ التفكير الحوسبي محصوراً في تطوير البرامج والخوارزميات لحل المشكلات الحوسبيَّة. على سبيل المثال: لا يختصُّ بالخوارزميات إلا جزء صغير من تطوير التطبيقات، ويركز الجزء الأكبر منها على تصميم نُظُمٍ للتعامل مع اهتمامات المجتمع. التصميم بهذا المعنى هو تفاعل مُستمر بين المُصمِّمين والمستخدمين، حيث يشاهدون ردود أفعالهم على نماذج البرامج، ويُقيِّمون البرامج الناجحة وغير الناجحة، ويعدِّلون البرامج وفقاً لذلك. وتُعَدُّ هذه رؤية أوسع للتصميم مقارنة برؤى مُجتمعات البرمجة وهندسة البرمجيات القديمة مثل «النموذج» أو «الخطة» أو «إعداد التجربة». إنها مجموعة مهارات تجمع بين إدراك التوجُّهات السائدة في المجتمعات وتاريخ هذه المجتمعات، بالإضافة إلى المعرفة العميقة

بالتقنيات الحالية وغيرها من العناصر المفيدة. فالتصميم يتطلب فهم البشر في مجتمعاتهم بقدر ما يتطلب فهم التكنولوجيا.

كان من بين الآثار المترتبة على التصميمات الجديدة في الحوسبة أتمتة العديد من المهام المعرفية التي كانت تُعد خارج نطاق آلات الحوسبة حتى وقت قريب. هذا النوع من الأتمتة ربما يؤدي إلى الاستغناء عن الأيدي العاملة، وهو ما يُثير بدوره مخاوف كبيرة بشأن إمكانية أتمتة العديد من الوظائف الحالية، مما يؤدي إلى فقدان العديد من الأشخاص لوظائفهم. ومع ذلك، توجد على الجانب الآخر حقيقة مهمة مفادها أن التقنيات الجديدة تولّد مشكلات جديدة تتطلب تصميمات جديدة، مما يخلق وظائف جديدة للمصممين.

التصميم الحوسبي هو الآن مهارة يُمكنك الحصول عليها في أي مجال، بالإضافة إلى مهاراتك الأساسية في المجال الذي تختص به. فليس عليك أن تكون عالم كمبيوتر لتكون مُصمماً حوسبياً. يمتلك التصميم الحوسبي روح ثورة الحوسبة المضطربة في الوقت الحالي أكثر من التفكير الحوسبي. وقد أظهرت لنا الثورات التكنولوجية السابقة أن التقنيات الجديدة تؤدي في النهاية إلى خلق وظائف أكثر مما تؤدي إلى الاستغناء عن وظائف. وثورات الحوسبة الحالية في تعلّم الآلة وتطوير التطبيقات تخلّق وظائف جديدة للمصممين، بينما تحوّل بعض الوظائف الحالية إلى وظائف قديمة مهملة من خلال أتمتتها. وينبغي للحكومات بذل المزيد من الجهد في برامج التدريب والتعليم حتى يتمكن العمال الذين فقدوا وظائفهم من تعلّم مهارات التصميم الخاصة بالوظائف الجديدة، من أجل المساعدة في تسهيل عملية الانتقال.

يؤدي التركيز الجديد على التصميم إلى إعادة إحياء الجانب الهندسي للحوسبة، وهو جانب أكثر حساسية للتصميم من الجانب العلمي. ويؤكد المهندسون على بعض القضايا المهمة مثل الوثوقية وتحمل الأخطاء وبنية النظام وتصميم النظم ككل؛ إذ غالباً ما كانت هذه الموضوعات تُهمّش في تعريفات التفكير الحوسبي الموجهة نحو النظريات والخوارزميات. علاوة على ذلك، يجلب المهندسون بعض الاهتمامات البشرية إلى التفكير الحوسبي؛ مثل فهم كيفية استخدام المجتمع لأجهزة الكمبيوتر، وتطبيق التقنيات الجديدة في المجتمع، والتوعية بالقضايا الأخلاقية التي تُثيرها التكنولوجيا وبآثارها الجانبية، وتوفير الوسائل التي يمكن من خلالها التأثير في تصرفات الآلات عن طريق قرارات البشر والتزامهم.

(٣) تعلّم الآلة

أصبحت الشبكات العصبية، التي تبلورت لأول مرة في أربعينيات القرن العشرين كنماذج مُحتمَلة لأجهزة الكمبيوتر الإلكترونية، هي التكنولوجيا الرئيسية وراء الذكاء الاصطناعي وتحليلات البيانات اليوم. وكانت الشبكة العصبية نموذجًا رياضيًا يقوم على «إطلاق» خلية عصبية عندما يتجاوز مزيجُ الإشارات من خلايا عصبية أخرى العتبة المُحددة له، وتدخل الخلية العصبية «المطلقة» حالة إثارة تُنقل بعد ذلك إلى الخلايا العصبية الأخرى. كان الدافع وراء محاكاة الدماغ هو أن أجهزة الكمبيوتر الذاتية التشغيل قد تؤدي المهام البشرية على نحوٍ أفضل عندما تكون مبنيةً من مكوناتٍ مُماثلة. بالطبع، دائرة هذه النماذج العصبية ليست مثل الدماغ الحقيقي. فالدوائر المنطقية لأجهزة الكمبيوتر الأولى كانت تعمل بشكلٍ أسرع بكثيرٍ من الدوائر العصبية. أما الآن، فالوضع مختلف؛ فقد أصبحنا نعرف كيفية استخدام بطاقات الرسومات الرخيصة لتسريع عمليات حوسبة الشبكة العصبية. وتقوم شركتنا «آي بي إم» و«إنتل» حاليًا بتسويق شرائحٍ أسرع؛ فهما تُدركان أن هناك حاجة إلى طريقة تفكيرٍ جديدة لاستخدام شرائحهما على أفضل وجه، وتوفّران دوراتٍ تدريبية في تشغيل شرائحهما واستخدامهما.

كانت الشبكات العصبية المبكرة صغيرةً ومن السهل إرباكها عند تقديم مُدخلاتٍ جديدة إليها لم تكن موجودة في مجموعات التدريب الخاصة بها. أما الشبكات العصبية الحديثة، فتتكوّن من العديد من الطبقات، ولديها سعة أعلى بكثيرٍ من الشبكات القديمة، وكذلك فهي لا ترتبك بسهولة. وبفضل شرائح معالجة الرسومات، تستجيب الشبكات العصبية المُدرّبة والمكوّنة من العديد من الطبقات للمُدخلات على الفور تقريبًا. ونظرًا إلى أن الطبقات والعُقد والقيم المرجحة للاتصالات لا تتغير بعد تدريب الشبكة، فإن الأداء لا يعتمد على البيانات المُدخلة. ولأن تطبيقات الشبكات العصبية عادةً لا تحتوي على حلقاتٍ تكرارية، فإنها أيضًا تعمل في وقتٍ ثابت باستخدام مساحاتٍ ثابتة في الذاكرة. وهذا يعني أنه يمكن استخدام الشبكات العصبية في تطبيقاتٍ آتية بتواريخ نهائية أكثر موثوقية مقارنةً بالبرامج التقليدية.

من المميزات الأساسية للشبكات العصبية أنها «تُدرّب» بدلًا من أن «تُبرمج». على سبيل المثال: ليس لدينا خوارزميات شديدة الموثوقية للتعرف على الوجوه، ولكن يمكن تدريب الشبكات العصبية على التعرف إلى وجوهٍ مُعينة بشكلٍ موثوق به للغاية. وغالبًا ما يُقال إن هذه الشبكات «مُبرمجة ذاتيًا» نظرًا إلى عدم وجود مبرمجٍ يُحدّد القيم المرجحة

الداخلية، مع أن خوارزمية تعديل القيمة المُرجَّحة المستخدمة في التدريب يمكن اعتبارها مبرمجًا آليًا. بالنسبة إلى العديد من المشكلات، من الأسهل بكثير إيجاد بيانات تدريب مناسبة أو إنشاؤها بدلاً من كتابة برنامج قائم على مجموعة معروفة مسبقًا من القواعد. وكما ذكرنا في الفصول السابقة، فإن إحدى المشكلات الكبيرة المتعلقة بالشبكات العصبية هي أنه لا توجد طريقة لـ «تفسير» كيفية توليد الشبكة للمُخرجات، كما هو مُمكن مع البرامج التقليدية. وتُعدُّ معرفة الأسباب الكامنة وراء استنتاج مُعين أمرًا مهمًا في العديد من مجالات التطبيق مثل التشخيص الطبي؛ ولكن الشبكات العصبية لا توفر هذه الميزة. ومن ثَمَّ، كان لا بدَّ من تعديل التفكير الحوسبي بحيث يتضمَّن الأدوات المستخدمة لبناء الشبكات العصبية وتدريبها. وما زالت تنتظره تحديات أكبر في تقييم موثوقية الشبكات العصبية وأمانها.

ثمة ميزة أخرى كبيرة للشبكات العصبية، وهي أنها يمكن تدريبها من خلال تفاعلها مع الشبكات العصبية الأخرى بدلاً من إعطائها مجموعات بيانات للتدريب عليها. تم تدريب الشبكة العصبية الخاصة ببرنامج «ألفا جو»، الذي هزم بطل العالم في لعبة «جو» في عام ٢٠١٧، من خلال جعلها تلعب ضد شبكة «ألفا جو» أخرى؛ فتعلَّمت اللعبة بحُكم الممارسة بدلاً من تدريبها على مجموعة كبيرة من مباريات «جو» المسجلة. هذه الطريقة في تدريب الشبكات من خلال السماح لها بالتعلُّم من بعضها جعلها قادرة على إجراء تغييرات ملحوظة على اللعبة.

(٤) التعاون بين الإنسان والكمبيوتر

في عام ١٩٩٧، خسر لاعب الشطرنج جاري كاسباروف الحاصل على لقب «جراند ماستر» (الأستاذ الكبير) أمام كمبيوتر «ديب بلو» الذي أنتجته شركة «آي بي إم». تُمثل هذه المباراة علامة فارقة في تاريخ الشطرنج؛ لأنها كانت المرة الأولى التي يهزم فيها برنامج كمبيوتر أحد أبطال العالم. كان كاسباروف قد لعب العديد من المباريات السابقة ضد أجهزة كمبيوتر أقلَّ مستوى، وفاز بها جميعًا.

لم يعلن كاسباروف عن نهاية لعبة الشطرنج. بدلاً من ذلك، اخترع نوعًا جديدًا من الشطرنج؛ ألا وهو «الشطرنج المُتقدِّم». في «الشطرنج المُتقدِّم»، يتلقَّى كلا اللاعبين في المباراة المساعدة من جهاز كمبيوتر. وقبل تنفيذ الحركة التالية، يستشير اللاعب البشري برنامج الكمبيوتر للحصول على رؤى حول الآثار المُحتملة للحركات. وكان أداء لاعبي

الشطرنج بمساعدة الكمبيوتر أفضل من أدائهم دون مساعدة الكمبيوتر، ولكنه كان أفضل أيضًا من أداء أجهزة الكمبيوتر عندما لعبت وحدها.

إن فكرة أن فريقًا مكونًا من البشر والكمبيوتر يُمكنه دائمًا تقديم أداء أفضل من أداء الآلة الجيدة جدًا وحدها، هي فكرة مثيرة للجدل. فهناك تقارير عن بطولات حديثة في «الشطرنج المتقدم» أدت فيها الفرق المكوّنة من البشر والكمبيوتر أداءً سيئًا مقارنة بالمباريات بين أجهزة الكمبيوتر دون بشر. وفي الطب، لا يؤدي دائمًا الأطباء المشاركون في التشخيص مع أجهزة الكمبيوتر أداءً جيدًا يُضاهي أداء أفضل أجهزة الكمبيوتر التشخيصية.

ومع ذلك، فإن التعاون بين الإنسان والكمبيوتر جذب الكثير من الاهتمام في مجال البحث في الذكاء الاصطناعي؛ لأنه قادر على إجراء عمليات الحوسبة التي لا يستطيع إنسان أو كمبيوتر القيام بها بمفرده. ومن الأمثلة القديمة على ذلك وضع أسماء على الصور الرقمية باستخدام كلمات رئيسية قابلة للبحث. كان القيام بذلك باليد بطيئًا للغاية لدرجة تحول دون أن يكون مفيدًا لتسمية الصور عبر الإنترنت. وفي عام ٢٠٠٦، اخترع «لويس فون أن» من جامعة كارنيجي ميلون لعبة عبر الإنترنت حيث وضع آلاف الأزواج من البشر أسماءً على الصور المقدمة لهم؛ فإذا تطابقت كلماتهم الرئيسية، فإن صورتهم المسماة تدخل قاعدة البيانات القابلة للبحث. كان يُعتقد أن «دالة التسمية» التي تُنفذها الفرق المكوّنة من الإنسان والكمبيوتر هذه غير قابلة للحوسبة. وكانت الفرق المكوّنة من الإنسان والكمبيوتر أكثر قوة من أجهزة الكمبيوتر وحدها.

يتطلب التعاون بين الإنسان والكمبيوتر نوعًا مختلفًا من التفكير الحوسبي عن البرمجة التقليدية للكمبيوتر. ونحن نراقب باهتمام كبير كيف يتطور الجدل حول ما إذا كانت الفرق المكوّنة من البشر والكمبيوتر يُمكنها التفوّق على الآلات في المستقبل.

(٥) القفزات التكنولوجية

في عام ٢٠٠٦، تنبأ راي كريساويل، مخترع تقنيات الحوسبة والمُهتم أيضًا بالمستقبلات، بأنه بحلول عام ٢٠٣٠، سنكون قادرين على إنشاء كمبيوتر بحجم الدماغ، بنفس عدد الخلايا العصبية والوصلات الموجودة في الدماغ.^٥ وقال إن هذا الكمبيوتر، في تصوّره، سوف يطور وعيه الخاص وذكاءه الفائق. أما كيف ستتعامل هذه الآلات مع البشر فهو سؤال لا يمكن الإجابة عنه. وأفضل ما يمكن قوله في هذا الصدد هو أن هذه الآلات

الجديدة ستكون لها اهتمامات مختلفة للغاية عن اهتماماتنا بحيث لا يُمكننا تصوُّر كيف ستُعَامِلنا. وتُسَمَّى لحظةُ إنشائها «التفرد» بسبب عدم القدرة على التنبؤ بما يمكن أن يحدث بعد تطوير الذكاء الاصطناعي للوعي.

وصل كريسوايل إلى استنتاجه من خلال استقراء قانون مور، وهو توقع جوردون مور في عام ١٩٦٥ بأن شريحة السيليكون ستتضاعف قدرتها تقريبًا كل عامين بنفس السعر. وقد اتبعت صناعة شرائح الكمبيوتر هذا القانون عن طريق مضاعفة قدرة الشرائح كل عامين لأكثر من نصف قرن. ويعتبر قانون مور، من نواحٍ عديدة، انتصارًا للتفكير الحوسبي؛ لأن المهندسين كان عليهم أن يبذلوا قصارى جهدهم في التفكير لإيجاد طرق أفضل لإنشاء دوائر الحوسبة. استغل كريسوايل في تحليله ظاهرة القفزات التكنولوجية. فمنذ بداية العصر المعلوماتي في أوائل القرن العشرين، كما يقول، لوحظ نفس تأثير التضاعف في تقنيات ذلك الوقت، مثل آلات البطاقات المثقبة أو آلات الأنابيب المفرغة. وعندما كانت إحدى التقنيات تعجز عن تحقيق هذا التضاعف كل عامين، كانت تقنية أخرى تحلُّ محلها. ويعتبر قانون مور بشأن شرائح السيليكون هو في الواقع الموجة الخامسة من التقنيات التي تُظهر هذا التضاعف كل عامين. وتوقع كريسوايل بثقة حدوث المزيد من القفزات التكنولوجية واستمرار هذا الاتجاه، مما سمح له بالتنبؤ بقدرة المعالجة المتاحة بحلول عام ٢٠٣٠ وما بعده، والوصول إلى ما يُسمَّى بالتفرد.

تحدث القفزات التكنولوجية على نحوٍ معتاد في صناعة الحوسبة. فالاعتماد على تقنية معينة على مرِّ الزمن يتبع دائمًا منحنى على شكل حرف S، مع نموٍّ أسي حتى نقطة الانعطاف، وبعدها يتباطأ النمو مع تشبُّع السوق. ويُولي كبار رجال الأعمال اهتمامًا كبيرًا بنقاط الانعطاف هذه لأن منافسًا يتمتع بتقنية أفضل تنمو بشكلٍ أسي يمكن أن يقلب أعمالهم رأسًا على عقب عندما يتباطأ نموها. وهم يُحاولون توقُّع نقاط الانعطاف هذه من خلال تطوير تقنيات جديدة في مختبراتهم البحثية والقفز إليها عندما تصل تقنيَّتهم الحالية إلى نقطة الانعطاف. وبذلك يُمكنهم ركوب موجة التقنية الجديدة خلال مرحلة نموها الأسي.

على الرغم من أن التفرد هو نتاج للحوسبة والتفكير الحوسبي، فإنه لا يمكن التعامل معه بالتفكير الحوسبي، ولا يُمكننا أيضًا تحسين فهمنا له من خلال التفكير الحوسبي.

(٦) فرضية أن العالم كله عبارة عن كمبيوتر

رأى بعض العلماء أن المعلومات هي أساس كل الفيزياء. فكل جسيم وكل تفاعل هو نتاج لتدفق معلومات وتبادلها على مستوى أكثر عمقاً من أصغر الجسيمات المعروفة. وفي عام ٢٠٠٢، نشر ستيفن وولفرام، وهو عالم فيزياء ومُخترع برنامج «ماتماتيكا»، الذي يُعتبر في حد ذاته انتصاراً للتفكير الحوسبي، كتاباً كبيراً يؤيد هذا الرأي.^٦ وفي عام ٢٠٠٣، جادل الفيلسوف نيك بوستروم بإمكانية أن نكون شخصيات في محاكاة يقوم بها نوع أكثر ذكاءً بكثير يدرس أسلافه. وبينما يرى بعض علماء الفيزياء بعض الجدارة في الادعاء بأن جميع الجسيمات والتفاعلات يمكن تفسيرها بالموجات الاحتمالية لميكانيكا الكم، وهي أشكال من المعلومات، إلا أنهم يعتبرون فكرة أن عالمنا عبارة عن محاكاة رقمية فكرة خيالية للغاية.^٧

تستأثر فرضية أن العالم كله عبارة عن كمبيوتر بإعجاب أولئك الذين يعتقدون أن التفكير الحوسبي والحوسبة هما قوتان هائلتان لا حدود لهما.

(٧) صراعات أيديولوجية حول ما يجب تدريسه

هناك نقاش لا ينتهي حول ما يجب تدريسه في منهج الحوسبة. وهناك نقطتان جوهريتان في هذا النقاش. الأولى: تتعلق باختيار لغة البرمجة وإطار العمل البرمجي اللذين يجب تعريف الطلاب بهما. فهل يجب أن تكون لغة سهلة التعلم ولها بنية وتركيبات غير معقدة بالمرّة مثل لغة بايثون؟ أم يجب أن تكون لغة يستخدمها أرباب عملهم المستقبليون في الصناعة، مثل لغة جافا أو جافاسكريبت؟ وما هي فوائد البدء بإطار عمل يعامل البرامج كمصدر تعليمات لآلة (يُعرف باسم إطار عمل «أمري») مقارنة بإطار عمل يُعامل البرامج كأنها تتألف من وظائف (يُعرف باسم إطار عمل «وظيفي»)؟ هذه النقاشات كانت ركيزة أساسية لاجتماعات هيئة التدريس في علوم الكمبيوتر منذ الستينيات، ومن غير المرجح أن تتوقف في السنوات المقبلة.

النقطة الجوهرية الأخرى هي الصراع ما بين المنظور العلمي-الرياضي والمنظور الهندسي-التصميمي. فالمنظور الأول يعلم تجريدات الأشياء في العالم ويترك للطالب تطبيق التجريد على الحالة التي أمامه. أما المنظور الهندسي-التصميمي، فيركز على جميع التفاصيل التي يجب على المبرمج الاهتمام بها لكي يكون البرنامج الناتج آمناً وموثوقاً

به. وقد كان للمنظور العلمي-الرياضي اليد العليا لسنواتٍ عديدة، ولكن مع بروز أهمية التصميم، اكتسب المنظور الهندسي زخمًا جديدًا. والحقيقة أن كلا المنظورين مُهمَّان للنجاح في الحوسبة؛ فالجانب العلمي والجانب الهندسي يحتاج كلُّ منهما إلى الآخر.

(٨) تأملات في العالم الناشئ

نحن نكتب هذا الكتاب في الذكرى الخمسين لأول توصياتٍ لتطوير منهج دراسي في الحوسبة قدَّمتها جمعية آلات الحوسبة، وهي جمعية تضمُّ خبراء الحوسبة، ونحن المؤلِّفين كلانا ينتمي إليها. لقد شكَّلت ذلك المنهج والمواصفات اللاحقة له العديدُ من العوامل التي تناولناها في الفصول السابقة:

- التركيز القوي على التطوُّر التكنولوجي منذ البداية.
- المقاومة الواسعة لتشكيل أقسام للحوسبة من قبل الأقسام الأكاديمية الأخرى التي لم تقبل الحوسبة كحقْلٍ دراسي مُستقل.
- تطوير شبكة مجتمعية للحوسبة مع بدايات عصر الإنترنت.
- الخلافات الحادة حول أدوار العلوم والرياضيات والهندسة في الحوسبة، والتي تجلَّت في صراعات حول كيفية تدريس هندسة البرمجيات وتكنولوجيا المعلومات، ومدى الاعتماد على الأساليب الشكلية في تطوير البرمجيات.
- التعامل مع ظهور العلوم الحوسبيَّة وتوغُّل الحوسبة في جميع مجالات الحياة تقريبًا.
- صعود وهبوط الذكاء الاصطناعي وادِّعاءاته حول الأتمتة ومُستقبل البشرية.

إن مفاهيمنا وأفكارنا المتجذرة بشأن الحوسبة لا تساعدنا في مواجهة العديد من القضايا الملحة التي تطرأ على العالم من حولنا. لقد أدَّى التواصل العالمي الذي ساعدنا في تحقيقه من خلال الإنترنت إلى العديد من الفوائد، مثل جعل العالم قرية صغيرة وعولة التجارة. ولكنه أدَّى أيضًا إلى نشوء صراعاتٍ بين المنظمات غير الحكومية والدول التقليدية، وحروب تجارية، ودفاعية، وإرهاب، وانعزال واسع النطاق، وأخبار كاذبة، وتضليل، ومعلومات خاطئة، واستقطاب سياسي، وقلق واضح وعدم يقين بشأن كيفية المُضيِّ قدمًا في العالم. لقد أظهر لنا الوصول إلى كمِّ هائل من المعلومات عبر الإنترنت أن المعرفة لا تُغدق علينا ثمار الحكمة بالضرورة، ونحن نتوق إلى قادةٍ حكماء لم نجدهم

بعد. وبدلاً من أن نعيش في مجتمعٍ معلوماتي مُحترم مدعوم بالإنترنت، أضحيْنَا نعيش في مجتمعٍ مُستقطَّب تسوقه وسائل التواصل الاجتماعي. إن العالم الذي نواجهه في حياتنا اليومية مليء بالمفاجآت والأحداث غير المتوقَّعة والحالات الطارئة التي لا تستطيع حتى أفضل آلات التعلُّم لدينا وأفضل وسائل تحليل البيانات التنبؤ بها بشكلٍ كامل. ونحن نشهد الآن تنافُس العديد من الدول على العديد من الموارد، بما في ذلك استخدام البحار والمجالات الجوية، ونفتقر إلى وسائل فعَّالة لحلِّ هذه النزاعات، ونخشى أن تؤدي الصراعات الناتجة إلى حروبٍ أو انهيارات اقتصادية. ونرى أن النشاط البشري الجماعي يؤثر على البيئة العالمية ولكننا لم نجد بعدُ طريقاً مُستدامة لحماية بيئتنا للأجيال القادمة. يضعنا كل هذا أمام سؤالٍ مُهم: كيف يُمكننا تشكيل التعليم في مجال الحوسبة بحيث يتمكن خريجونا من تطوير الحسِّ التصميمي والحكمة والعناية التي يحتاجونها للإبحار في هذا العالم الذي سيكونون مواطنين فيه؟ إن منهجنا الدراسي الحالي، والذي يركز على الدليل المعرفي لعام ٢٠١٣، غير مُجهز لهذه المهمة.

إن مفاهيمنا وأفكارنا المتجذرة بشأن الحوسبة لا تساعدنا في مواجهة العديد من القضايا الملحة التي تطرأ على العالم من حولنا. وقد أظهر لنا الوصول إلى كمِّ هائل من المعلومات عبر الإنترنت أن المعرفة لا تُغدق علينا ثمار الحكمة بالضرورة.

يمكن أن تكون نقطة البداية هي تخصيص مساحةٍ في منهجنا الدراسي المزدحم لإجراء محادثاتٍ حول الأسئلة الكبرى بشأن عواقب الحوسبة في جميع أنحاء العالم. ويجب أن يشارك في هذه المحادثات تخصصات وأجيال متعددة. وتهدف ليس إلى حلِّ المشكلات بل إلى الانخراط في التفكير المُشترك، وتعزيز التفاهم المتبادل والتقدير والاحترام حول هذه القضايا. بعض الأمثلة على الأسئلة الكبرى التي تصلحُ لمثل هذه المحادثات المرتبطة بالتفكير الحوسبي هي:

- ما الذي لا يمكن أتمتته؟ وما الذي يجب أتمتته؟ وإلى أي مدى يمكن أن تأخذنا الأتمتة؟ ولن الحقُّ في تحديد ما يُؤتمت وما لا يُؤتمت؟
- كيف يمكن للكذاء الاصطناعي أن يخلق وظائف أكثر من خلال الأتمتة أكثر من تلك التي يحلُّ محلُّها؟

- كيف يُمكننا مساعدة الأشخاص الذين يُستغنى عن وظائفهم بسبب البرامج والأجهزة التي صمَّناها؟
- كيف نساعد في توليد مُصمِّمين جيِّدين؟
- كيف يُمكننا زيادة الثقة في قرارات الشبكات العصبية عند إعطائها مدخلاتٍ خارج مجموعات تدريبها؟
- كيف يُمكننا مقاومة تشكيل مجتمع مُراقب آلياً؟
- ما الحلول التكنولوجية التي يمكن إيجادها لمشكلة الأمن السيبراني؟
- كيف نجعل عالمنا يمضي قدماً في الوقت الذي أصبحت فيه أجهزة الكمبيوتر مضمَّنة تقريباً في جميع الأجهزة المتصلة بالشبكة العالمية؟
- كيف تؤثر التكنولوجيا الرقمية في السياسة العالمية والقومية والميزان التجاري وتغيُّر المناخ وغيرها من قضايا العولمة؟
- كيف ستؤثر سلاسل الكتل والعملات المشفرة على مشاكلنا المتعلقة بالثقة في الهيئات المركزية؟ وهل هي باهظة الثمن جدًّا بحيث لا يمكن الحفاظ عليها؟
- كيف نحمي المجتمعات التي تعتمد بشكلٍ كبير على الحوسبة من أي هجوم على عنصر أساسي من البنية التحتية، مثل شبكة الكهرباء أو الإنترنت؟
- كيف نُحضّر الناس لتقدير الفرق بين الحكمة ووفرة المعلومات؟
- ما التداعيات الاجتماعية لواجهات الدماغ والكمبيوتر، والغرسات العصبية في أدمغتنا وأجسادنا؟
- ما الانهيارات الاقتصادية المُحتَملة بسبب انخفاض تكلفة العديد من التقنيات المترابطة بشكلٍ كبير؟

نحن لا نعتقد أن أيًّا منَّا لديه إجابات عن أيٍّ من هذه الأسئلة. ولكننا نحتاج إلى إجراء مُحادثات حولها. وفي أثناء القيام بذلك، نحتاج إلى أن نحتضن علماء الرياضيات وعلماء الاجتماع والفلاسفة وعلماء الأنثروبولوجيا والمُحامين والمهندسين وكل من له علاقة بهذا المجال. لقد حان الوقت لنُفكر معاً في تصميم التكنولوجيا وتأثيراتها، ومن ثمَّ نشكل مستقبلنا بالحكمة والتفاهم. لقد حان الوقت للتخلُّي عن الصراعات القديمة التي ورثناها من أزمِنةٍ قد مضت، والعمل معاً إخوة وأخوات، آباء وأمهات، صغاراً وكباراً للإجابة عن هذه الأسئلة المهمة.

خاتمة: الدروس المستفادة

أثناء بحثنا لكتابة هذا الكتاب، تعلّمنا بعض الدروس التي تستحقّ التلخيص هنا.

الدرس الأول: التفكير الحوسبي إضافة وليس بديلاً

يعتقد الجميع أن طرق التفكير (والممارسة) الخاصة بمجالهم قيّمة وتستحقّ التعلّم في العديد من المجالات الأخرى. ويرغب المتحمّسون في نشر سجل النجاح في مجالاتٍ أخرى. قائمة أنواع «التفكير» التي ستُنشر طويلة: ومنها التفكير الحوسبي، والتفكير المنطقي، والتفكير الاقتصادي، والتفكير النظامي، والتفكير الفيزيائي، والتفكير الرياضي، والتفكير الهندسي، والتفكير التصميمي، وغيرها.

استنتاجنا هو أن التفكير الحوسبي غالباً ما يكون إضافة مرحّباً بها في المجالات الأخرى، ولكنه ليس بديلاً لأساليب التفكير الخاصة بهذه المجالات، وليس مهارةً عليا تصلح في جميع المجالات.

الدرس الثاني: التفكير الحوسبي موضوع قديم ومُتفرع خضع لدراسة مُستفيضة

ذاع صيت مصطلح «التفكير الحوسبي» بعد أن أدرجته المؤسسة الوطنية الأمريكية للعلوم في دعوة للتمويل في عام ٢٠٠٧. بالنسبة إلى العديد من الناس، كانت هذه هي المرة الأولى التي يسمعون فيها آراءً حول قيمة الحوسبة في التعليم. وبدا وقتها أن التفكير الحوسبي اختراع جديد، وإنجاز يُبشّر بثورة في التعليم من مرحلة رياض الأطفال إلى مرحلة التعليم

الثانوي. ولكن الحقيقة هي أن البشر كانوا يُمارسون التفكير الحوسبي منذ أكثر من ٤٥٠٠ عام. وقد رُشح للتعليم ما قبل الجامعي منذ ستينيات القرن العشرين. حاول بعض أوائل مُصممي مناهج التفكير الحوسبي للمراحل من رياض الأطفال وحتى مرحلة التعليم الثانوي إنشاء «دليل معرفي» للتفكير الحوسبي من الصفر دون أن يكونوا على دراية بالتاريخ الطويل للتفكير الحوسبي، بما في ذلك المحاولات المُماثلة لإدخال الحوسبة إلى المدارس. ومن ثَمَّ، ارتكبوا، عن غير قصد، بعض الأخطاء المفاهيمية في ادعاءاتهم حول قدرات التفكير الحوسبي وطبيعته. ونحن قلقون لأن التوقعات المبالغ فيها والمشاكل المفاهيمية يمكن بسهولة أن تُصبح جزءاً من فولكلور التفكير الحوسبي، وقد يستغرق الأمر سنواتٍ للتخلُّص منها. ولذا نُحِثُّ معلمي الحوسبة على الرجوع إلى الدليل الضخم الموجود بالفعل من أبحاث تعليم الحوسبة للقيام بهذا الأمر على أكمل وجه.

الدرس الثالث: سرعة أجهزة الكمبيوتر هي المُحرك الرئيسي لثورة الحوسبة

معظم ما تفعله البرامج لنا يُصبح ممكناً بسبب تفاوتات السرعة غير المفهوم بين أجهزة الكمبيوتر والبشر؛ فأجهزة الكمبيوتر أسرع من البشر بمليارات إلى تريليونات المرات. وعلى الرغم من أن البشر يُمكنهم تنفيذ خطوات الحوسبة، فلن يتمكنوا من تنفيذ معظم هذه العمليات الحوسبية في فترة حياتهم. ومن ثَمَّ يمكن للآلات أن تفعل ما هو مُستحيل حرفياً بالنسبة إلى البشر. ومع أن البشر يُمكنهم تنفيذ الخوارزميات الخاصة ببعض مهام معالجة المعلومات، فإن ثورات عصر الكمبيوتر لا تدور حول ما يمكن للناس تنفيذه من الخوارزميات في فترة حياتهم، ولكنها تدور حول ما يمكن للكمبيوتر القيام به بالنيابة عن الناس.

الدرس الرابع: التفكير الحوسبي المُتقدِّم يعتمد على المجال

في حالة المهام المُتقدمة، أنت تحتاج إلى فهم المجال الذي تريد أن تعرف فيه كيف تجعل الكمبيوتر يقوم بمهمة ما نيابةً عنك. على سبيل المثال: فإن مبرمجاً خبيراً لا يعرف شيئاً عن فيزياء الكم سيكون لديه القليل ليُقدمه إلى فريقٍ من علماء الفيزياء الذين يعملون على كمبيوتر كمي. وبالمثل، فإن العمل مع خوارزميات الطبيعة المُعقدة في علم الأحياء يتطلب فهماً كبيراً للعمليات البيولوجية. وتتطلب النماذج الخوارزمية في الكيمياء درايةً عميقة

بالعمليات الكيميائية المناظرة. ويتطلّب إنشاء نظام معلومات لأي مُستشفى فهمًا واسعًا للعمليات المؤسسية والمعلوماتية ولجالات العمل في سياق المُستشفى. ومن ثمّ، فإن الكثير من التفكير الحوسبي المُتقدّم مُحدد بالسياق ومرتبطة ارتباطًا وثيقًا بمجال التطبيق.

الدرس الخامس: التفكير الحوسبي غير أدوات العلم وأساليبه ونظرياته المعرفية

لقد عزّز التفكير الحوسبي حدوث ثورة في العلوم. فقد وجد العلماء في جميع المجالات أن التفكير الحوسبي هو أسلوب جديد لممارسة العلوم، مختلف عن الأساليب الكلاسيكية القائمة على النظرية والتجريب. وتوصّلوا إلى هذا الاكتشاف في ثمانينيات القرن العشرين عندما بدءوا في استخدام أجهزة الكمبيوتر الفائقة للتصدّي لـ «التحديات الكبرى». وكان هذا تحولًا كبيرًا في النموذج مهّد الطريق أمام العديد من الاكتشافات العلمية الجديدة. وطوّر كلُّ مجالٍ نوعه الخاص من التفكير الحوسبي الذي لم يُستجلب من علوم الكمبيوتر. وأثري التفكير الحوسبي في علوم الكمبيوتر من خلال تعاونه مع العلوم الحوسبيّة.

الدرس السادس: الوجه المعروف للتفكير الحوسبي هو التفكير الحوسبي الأساسي

يُوصف التفكير الحوسبي لأغراض المناهج الدراسية من مرحلة رياض الأطفال وحتى مرحلة التعليم الثانوي بأنه مجموعة من مفاهيم البرمجة وقواعدها. ولكن الكثير من المُحترفين يرون التفكير الحوسبي على أنه مهارة تصميم، ويرى العديد من علماء الطبيعة أنه طريقة مُتقدمة للتفسير العلمي. ومثل جميع المهارات، يُمكنك أن تكون مبتدئًا، أو مبتدئًا متقدمًا، أو كفئًا، أو مُتمكنًا، أو خبيرًا، أو مُحترفًا. ويبدو أن العديد من النقاشات حول ماهية التفكير الحوسبي «الحقيقي» لا تستطيع التمييز بين المستويات المختلفة لمهارات التفكير الحوسبي. على سبيل المثال: يدافع المُعلّمون من مرحلة رياض الأطفال إلى مرحلة التعليم الثانوي عن مناهج التفكير الحوسبي التي تستهدف بشكلٍ أساسيّ المبتدئين، وتحتوي على مجموعة صغيرة قابلة للتدريس من الأفكار والممارسات والمهارات المتعلقة بالتفكير الحوسبي. ويرى آخرون التفكير الحوسبي على أنه مهارات مُتقدمة ومُحترفة تتطلّب سنواتٍ عديدة من الممارسة والخبرة. إن الفشل في التمييز يؤدي إلى الصراعات؛ على سبيل المثال: فإن الضجّة حول كيف يُتيح تعلم البرمجة مساراتٍ وظيفية جديدة صامتة حول ما يفعله مصممو الحوسبة المُحترفون. وجهود التعليم مهمة على جميع المستويات من مرحلة التعليم ما قبل الجامعي وحتى التعليم الجامعي وما بعده.

الدرس السابع: التفكير الحوسبي للمبتدئين والتفكير الحوسبي للمحترفين يُشكّلان معاً نسيجاً غنياً من التفكير الحوسبي

طوّر التربويون في قطاع المدارس من مرحلة رياض الأطفال وحتى مرحلة التعليم الثانوي «تفكيراً حوسبياً للمبتدئين»؛ أي أفكاراً وأساليب لتدريس الحوسبة للمبتدئين. وطوّر مُصمّمو نظم البرمجيات المحترفون والعلماء «تفكيراً حوسبياً للمحترفين»؛ أي أساليب متقدمة لتصميم وإنشاء برامج مُعقّدة تعمل بشكلٍ موثوق به وآمن، ولإجراء دراساتٍ علمية. وقد شجّع التضافّر بين هذين الجانبين من التفكير الحوسبي على ثورة الكمبيوتر.

الدرس الثامن: التغيير جزء لا يتجزأ من التفكير الحوسبي

لم يكن هناك قطُّ إجماع حول ماهية التفكير الحوسبي «الحقيقي». وربما لن يكون هناك إجماع كامل على ذلك. ففي كل عَقْدٍ في التاريخ الحديث للحوسبة، كانت هناك إجابات مختلفة عن جوهر التفكير الحوسبي. إن التقدّم في الحوسبة يحافظ على التفكير الحوسبي في حالة تغيّرٍ مُستمر. ويجب أن نعتبر عدم وجود تعريفٍ ثابتٍ دليلاً على حيوية المجال بدلاً من أن نعتبره فشلاً من جانبنا في فهم حقيقة أبدية.

مسرد المصطلحات

التجريد: تبسيط الظواهر المُعقَّدة من خلال تمثيل سماتها البارزة فقط، مع حذف التفاصيل أو إخفائها.

الخوارزمية: وصف أسلوب حساب دالِّ ما، أو بشكلٍ أوسع، حل فِئَةٍ من المسائل الحسابية. وتُحدَّد جميع الخطوات بدقة فائقة بحيث يمكن للآلة تنفيذها.

الذكاء الاصطناعي: فرع من فروع علوم الكمبيوتر يبحث فيما إذا كان يمكن جعل أجهزة الكمبيوتر المدعومة ببرامج مناسبة ذكية (ذكاء اصطناعي قوي)، أو ما إذا كان يمكن لأجهزة الكمبيوتر محاكاة المهام الإدراكية البشرية من خلال العمليات المعلوماتية (ذكاء اصطناعي ضعيف).

الآتمنة: استخدام الآلات لتحلَّ محلَّ المراقبين البشر في العمليات المادية (مثل مصانع الكيماويات أو خطوط التصنيع)، أو في أداء العمليات المعرفية في مجال العمل (مثل مراجعة المُستندات أو معالجة الفواتير)، أو إنشاء كمبيوتر لأداء المهام التي كان يؤديها البشر سابقًا.

البت والبايت: البت هو أصغر وحدة معلومات تُميز بين وجود شيءٍ ما (١) أو عدم وجوده (٠). أما البايت فهو مجموعة من ٨ بت؛ مما يسمح بـ ١٢٨ تركيبة مُمكنة لـ ٨ بت. يمكن أن تمثل المجموعات الكبيرة بما فيه الكفاية من وحدات البت أيَّ شيء قابل للتمثيل بقيم متميزة، مثل الأرقام أو الأحرف أو الأنماط على شاشة العرض أو الألوان.

الجبر البوليني: مجموعة من التعبيرات يمكن تكوينها من المتغيرات المنطقية (كلُّ منها يُمثل بَنًا واحدًا عبارة عن true أو false) إلى جانب عوامل تشغيل مثل OR وAND

وNOT. وتُستخدَم التعبيرات البوليانية في لغات البرمجة لتحديد الشروط التي سَتُنَفَّذُ الجملة البرمجية عند تحقُّقها. كما أنها تُستخدَم لوصف وظائف الدوائر المنطقية داخل أجهزة الكمبيوتر.

وحدة المعالجة المركزية: المُكوّن المادي لأجهزة الكمبيوتر المسئول عن تنفيذ التعليمات الأولية مثل الجمع والطرح والانتقال والمقارنة، ويُقرر التعليمات التي سيجري تنفيذها بعد ذلك. تتضمّن مكونات الكمبيوتر الأخرى الذاكرة (التي تخزن جميع البيانات والتعليمات) وواجهة الإدخال والإخراج (التي تتّصل بالعالم الخارجي).

السحابة الإلكترونية: شبكة عالمية من نُظم التخزين ونُظم المعالجة يمكن الوصول إليها من أي مكانٍ عند الحاجة. ولا يعرف المُستخدِمون الذين يستأجرون مساحة تخزين البيانات ووقت المعالجة مكان تخزين بياناتهم ومعالجتها فعليًا.

برنامج التحويل البرمجي: برنامج يُترجم البرامج المكتوبة بلُغة برمجة عالية المستوى مُخصّصة للبشر إلى شفرة آلية ثنائية مُخصّصة للمعالج.

التعقيد الحوسبي: فرع من فروع علوم الكمبيوتر يدرس الصعوبة الجوهرية لحلّ المشكلات. وتُقاس درجة الصعوبة بخطوات الحوسبة ومساحة الذاكرة المطلوبة. بعض المشكلات مثل البحث عن اسم في قائمة تُعد «سهلة»؛ لأنها يمكن حوسبتها في وقتٍ يتناسب طرديًا مع طول القائمة. وبعضها، مثل إيجاد أقصر جولة في مجموعة من المدن، تُعد «صعبة»؛ لأنها في أسوأ الحالات تتطلّب سردًا وقياسًا لجميع الجولات المُمكنة، ومن ثَمَّ يزيد الوقت المطلوب لإنجازها أسّيًا بزيادة عدد المدن والطُرُق.

النموذج الحوسبي: وصف لآلة مجردة تُنفذ خوارزميات؛ على سبيل المثال: شريحة كمبيوتر تقليدية تنفذ تعليمات الآلة واحدة تلو الأخرى، أو شبكة عصبية تتعرّف على الوجوه في الصور، أو كمبيوتر كمي يفك الشّفرات. في العلوم والهندسة، يشير أيضًا إلى نموذج رياضي لعملية فيزيائية، يمكن محاكاته أو تقييمه بواسطة الكمبيوتر.

الكمبيوتر أو الحاسب: كيان، بشري أو آلي، يُمكنه إجراء العمليات الحسابية ومعالجة الرموز وَفَقًا لمجموعة من القواعد المُحددة بدقة. من القرن السابع عشر إلى ثلاثينيات القرن العشرين، كان مصطلح «الحاسب» — أو «الكمبيوتر» كما تُترجمه عن الإنجليزية — يعني الشخص الذي يقوم بالحساب. وبداية من أربعينيات القرن العشرين، ظهرت

أجهزة الكمبيوتر الإلكترونية الأولى وأطلقوا عليها «أجهزة الكمبيوتر الذاتية التشغيل». وبحلول الخمسينيات، حذفت صفة «الذاتية التشغيل» من الاسم.

تجريد البيانات: ممارسة نشأت مع المبرمجين في ستينيات القرن العشرين لاحتواء بنية بيانات مُعقدة خلف واجهة بسيطة. يمكن للمستخدمين الوصول إلى البيانات فقط من خلال الواجهة؛ ولا يُمكنهم الوصول مباشرة إلى الذاكرة التي تحمل البيانات. ويكون عرض البيانات المرئي من خلال الواجهة مُبسّطاً للغاية؛ ومن هنا جاءت كلمة التجريد. ويُعدُّ الملف مثلاً على ذلك؛ فهو يبدو للمستخدم مثل حاوية لسلسلة خطية من وحدات البت؛ ولا تسمح الواجهة إلا بالقراءة والكتابة فقط. وربما يُنفَّذ الملف الفعلي كمجموعة من الكتل المُبعثرة في أنحاء وسيط التخزين، وكلها مخفية عن المستخدم.

مشكلة القرار: مشكلة مشهورة في المنطق الرياضي ظهرت في أوائل القرن العشرين. إذا كان لدينا نظام منطقي يتكون من البديهيات والقواعد الخاصة بإنشاء براهين للفرضيات، فهل توجد خوارزمية تُحدّد ما إذا كانت فرضية مُعينة صحيحة أم لا؟ لفترة طويلة اعتقد علماء الرياضيات بوجود مثل هذه الخوارزمية، لكن لم يتمكّنوا من العثور عليها. وفي ثلاثينيات القرن العشرين، عرّف عدد من علماء الرياضيات، الذين يعمل كلّ منهم بشكلٍ مُستقل عن الآخر، مفهوم الخوارزمية رسمياً وأثبتوا أنه لا يُوجد حلٌّ عام لمشكلة القرار.

التفكيك: تقسيم شيء مُعقد إلى أجزاء أبسط وأصغر يسهل إدارتها. في البرمجيات، تُصبح الأجزاء وحدات يجري توصيلها معاً عبر الواجهات.

الرقمنة: عملية استخدام الشّفرات الثنائية لتمثيل كائن ما. ويمكن معالجة التمثيل بواسطة الكمبيوتر. على سبيل المثال: يمكن أخذ عيناتٍ من شكل موجة الكلام ٢٠ ألف مرة في الثانية، حيث تنتج كل عينة قراءة لسعة الموجة وتشفيرها كقيمة ١٦ بت. ويمكن بعد ذلك تخزين الكلام الذي تمّت رقمنته ومعالجته على الكمبيوتر.

أهداف DRUSS: في هندسة البرمجيات، هي النظم البرمجية التي تكون موثوقاً بها ويمكن الاعتماد عليها وقابلة للاستخدام وأمنة.

التضمين: استخدام الواجهات لإخفاء الآليات والمعلومات الداخلية عن المُستخدمين الخارجيين من أجل تحسين إمكانية إعادة الاستخدام، وتقييد الوصول، وحماية المعلومات من أخطاء المُستخدمين، وإمكانية الصيانة.

الكسيريّات: مصطلح صاغه عالم الرياضيات بنوا ماندلبرو للمجموعات الذاتية التماثل عند المقاييس المختلفة. على سبيل المثال: يبدو ساحل الدولة مُتعرّجاً في صورة الأقمار الصناعية؛ وكذلك يبدو متعرّجاً عند مشاهدته من طائرة شراعية؛ ولا يزال يبدو متعرّجاً في اللقطات المُقربة للأمواج المتدفّقة على الرمال. وتستخدم الكسيريّات (الفراكلتات) في الرسومات لرسم الكائنات المُعقدة من الأشكال البسيطة التي يمكن تكرارها على جميع المقاييس.

التعميم: توسيع نطاق الحلّ بحيث يشمل فئةً أوسع من المشكلات المُماثلة.

وحدة معالجة الرسومات: شريحة مُضمّنة في الكمبيوتر لتشغيل شاشة العرض الرسومية. ويمكن لوحدة معالجة الرسومات الحديثة الاحتفاظ بتمثيلات ثلاثية الأبعاد للكائنات ويمكن تدويرها إلى أي زاوية أو تحريكها إلى أي مسافة حوسبياً، ثم إسقاط الصورة الناتجة على الشاشة الثنائية الأبعاد، وكل ذلك في الوقت الفعلي.

الخوارزميات التجريبية: إجراءات لإيجاد حلول تقريبية للمشكلات غير القابلة للحل حوسبياً. على سبيل المثال: في الشطرنج نُقيّم الحركات المقترحة بنظام عدّ النقاط للقطع المفقودة؛ وهذا أقلُّ بكثيرٍ في كثافة العمليات الحوسبية من سرد جميع الحركات المُستقبلية المحتملة. تعطي الخوارزميات التجريبية الجيدة حلولاً جيدة جداً في معظم الأوقات.

بنية if-then-else: شكل من أشكال العبارات في لغة البرمجة وتختار هذه البنية بين مسارين أو أكثر من المسارات البديلة في تعليمات البرنامج. على سبيل المثال: العبارة `if sum ≥ 0 then color sum-value black; else color sum-value red` يستخدمها المحاسبون لتمييز الأعداد السالبة باللون الأحمر في جداول البيانات الخاصة بهم.

الحدس: جانب من جوانب الخبرة المُجسّدة، حيث يملك الخبير القدرة على معرفة كيفية التعامل فوراً مع موقف مُعين، بناءً على خبرته السابقة الطويلة. وربما يستطيع الخبير معرفة ما ينبغي فعله، ولكنه لا يستطيع شرح الأسباب.

اللوغاريتم: في الرياضيات، اللوغاريتم لعددٍ مُعين هو الأس الذي يجب رفع أساس ثابت إليه لإنتاج هذا العدد. وهكذا، فإن لوغاريتم الأساس ٢ للعدد ٨ هو ٣ لأن $2^3 = 8$. اللوغاريتمات مفيدة في ضرب الأعداد حيث إن حاصل ضرب عددين يجمع أسيهما. على سبيل المثال: ضرب ٨ في ١٦. ولأن $2^2 \times 2^4 = 2^6$ ، يُمكننا أخذ لوغاريتمات الأساس ٢

لكلا الحَدَّين (هنا ٣ و ٤ على التوالي)، وجمع اللوغاريتمات (مما يُعطينا الناتج ٧)، ثم رفع الأساس ٢ إلى أُس اللوغاريتم الناتج (هنا ٧٢). تُجرى عمليات الضرب باستخدام المساطر المُنزقة عن طريق جمع لوغاريتمات العددين المُضروبين.

الدوائر المنطقية: الدوائر الإلكترونية الأساسية في الكمبيوتر. وهي تجمع بين الإشارات الثنائية وعوامل التشغيل AND و OR و NOT وتخزن النتائج في سجلات، تُعالج بواسطة دوائر منطقية أخرى في دورة الساعة التالية.

تعليمات الآلة: تعليمات الخوارزمية المُشفرة إلى أكواد ثنائية يمكن للكمبيوتر التعرف عليها وتنفيذها.

الشبكة العصبية: شكل من أشكال الدوائر التي تأخذ نمط وحدات بت كبيراً جداً كمُدخل (مثل ١٢ ميجابكسل في الصورة الفوتوغرافية) وتُنتج مُخرجاً (مثل التعرف على الوجوه في الصورة). صُممت مكونات هذه الشبكة لتكون مُشابهة إلى حد ما للخلايا العصبية في الدماغ. وتتعلّم الشبكة من خلال التدريب وليس من خلال البرمجة.

نظام التشغيل: برنامج التحكم الذي يشغل نظام الكمبيوتر. يسمح للمستخدمين بتسجيل الدخول والوصول إلى بياناتهم، ويحمي بيانات المستخدمين من وصول الآخرين إليها دون إذن، ويُخطط الموارد (وحدة المعالجة المركزية والأقراص والذاكرة) بين المستخدمين المتنافسين، ويوفر بيئة يمكن للمستخدمين تشغيل برامجهم فيها.

وحدات البت الكمية (كيوبت): العناصر الأساسية في الكمبيوتر الكمي. وهي النظائر الكمية لوحداث البت الموجودة في الكمبيوتر التقليدي، ولكن لديها خاصية غريبة تُسمّى التراكب، مما يعني أنها يمكن أن تكون في الحالتين ٠ و ١ في الوقت نفسه. تزيد خاصية التراكب بشكل كبير من قوتها التمثيلية والحوسبية. وتُمثّل وحدات البت الكمية عن طريق دوران الإلكترونات أو الحقول المغناطيسية.

حالات التعارض: تحتوي العديد من الدوائر الإلكترونية على مسارات متعددة تربط إدخالاً بمخرج مُعين. وإذا انتقل تغيير الإدخال بسرعات مختلفة عبر المسارات المختلفة، فقد تتذبذب قيمة المُخرج بشكل عشوائي اعتماداً على ترتيب وصول الإشارات. ويمكن أن يتسبّب هذا التذبذب العشوائي في حدوث أعطال في الدوائر اللاحقة التي تستخدم المُخرج. ويمكن أن تظهر حالات التعارض أيضاً في نظم التشغيل عندما يُحاول

مستخدمان الوصول إلى ملف في نفس الوقت، وتعتمد القيمة النهائية للملف على مَنْ وصل في الآخر.

السجلات: سجلات المعالج هي اللبنة الأساسية للتخزين داخل وحدة المعالجة المركزية. ويتكون السجل من مجموعة من الدوائر الإلكترونية القلابة ذات الوضعيتين، وهي عبارة عن دوائر صغيرة يمكنها تخزين ٠ أو ١. ومن ثم، يتكوّن سجل ٨ بت من ٨ دوائر قلابة. وتجمع تعليمات وحدة المعالجة المركزية القيم في السجلات وتُخزن نتائجها في سجلات أخرى.

التمثيل: تعتمد الحوسبة بشكل كبير على تمثيل شيءٍ لشيءٍ آخر. وتتطلب عمليات الحوسبة تمثيل المعلومات في شكل رقمي، مثل قيمتين من الجهد في الدوائر، أو وجود أو عدم وجود اضطرابات على المواد. ونستخدم ٠ و ١ لتمثيل هذه الظواهر الفيزيائية.

المحاكاة: تعتمد عمليات المحاكاة التي يُجريها الكمبيوتر على النماذج الحوسبية للظواهر من أجل تتبع سلوك تلك الظواهر على مدار الزمن. وتتشكل النماذج من النظريات والمتغيرات والمعادلات والمعاملات وغيرها من سمات الظواهر، من أجل وصف النظام الذي يُوضع نموذج له بدقة. وتستخدم المحاكاة عناصر النموذج هذه لرؤية كيفية تغير النظام من وحدة زمنية إلى أخرى.

فرضية الانتقال: فرضية مفادها أنّ تعلم التفكير الحوسبي في علوم الكمبيوتر يتحول إلى قدرة على حل المشكلات في مجالات أخرى. تتنبأ الفرضية بأن الشخص الذي أتقن حلّ المشكلات في علوم الكمبيوتر سيكون قادرًا على حل المشكلات في الفيزياء بالخبرة نفسها. ولا يدعم هذه الفرضية سوى قليل من الأدلة التجريبية.

قيمتا الصدق: هما القيمتان المسموح بهما للمتغير المنطقي وهما true و false. وعند تمثيل هاتين القيمتين رقميًا، عادة ما يُفسّر ٠ على أنه false و ١ أو أي قيمة غير صفرية على أنها true.

اختبار تورينج: اختبار اقترحه آلان تورينج في عام ١٩٥٠ لتسوية مسألة ما إذا كان يمكن للآلة التفكير. يُجري مراقب بشري مُحادثتين نصيّتين، واحدة عبر اتصال بجهاز كمبيوتر، والأخرى عبر اتصال بإنسانٍ آخر. ولا يعرف المراقب أيهما الكمبيوتر وأيهما الإنسان. إذا لم يتمكن المراقب لفترة طويلة من تحديد الإنسان (أو الآلة) بشكل قاطع، فستُعد الآلة ذكية.

ملاحظات

الفصل الثاني: أساليب الحوسبة

- (1) Davis (2012).

الفصل الرابع: علوم الكمبيوتر

- (1) Mahoney (2011).
- (2) Newell, Perlis, and Simon (1967).
- (3) Simon (1969).
- (4) Knuth (1974, 1985).
- (5) Dijkstra (1974).
- (6) Forsythe (1968).
- (7) Knuth (1985).
- (8) Guzdial (2014).
- (9) Arden (1980).

(10) In his talk *A Logical Revolution*, Moshe Vardi describes the changing role and perceptions of logic in the field of computing, including the 1980s gloominess over what computers cannot do.

الفصل الخامس: هندسة البرمجيات

(1) Niklaus Wirth, software pioneer and the designer of the popular language Pascal, gives an excellent account of the development of programming practices and their supporting languages (Wirth 2008).

(2) Stokes (1997).

(3) Wilkes, in Metropolis, Howlett, and Rota (1980).

(4) Wirth (2008).

(5) Dijkstra (1980).

(6) Saltzer and Schroeder (1975).

(7) Alexander (1979).

(8) Gamma et al. (1994).

(9) Lampson (1983).

(10) The levels principle was first used by Edsger Dijkstra in 1968 to organize the software of an operating system. It facilitated a correctness proof of the system because each level depended only on its components and the correctness of the lower levels, but not the higher levels. The discipline of designing a system as levels leads to much smaller and more easily verified systems.

الفصل السادس: التصميم لتلبية احتياجات الإنسان

(1) Forsythe (1966).

(2) Grudin (1990).

(3) Leveson (1995).

(4) Parnas and Denning (2018).

(5) Winograd (1983).

(6) Denning (2016).

الفصل السابع: العلوم الحوسبيّة

(1) Baltimore (2001).

(2) Wilson (1989).

(3) Baltimore (2001).

(4) For the more mathematically inclined, the Mandelbrot set is the points in the complex plane at which the series of values of a function converges. A complex number is represented as $a + bi$, where $i = \sqrt{-1}$ and $i^2 = -1$. The equation of the series is $z(n + 1) = z^2(n) + c$ where $z(n)$ and c are complex numbers. Having chosen a value of c , compute a series of $z(n)$ – values starting with $z(0) = c$. (You may need to go to an algebra refresher for algorithms to multiply complex numbers.) If the $z(n)$ sequence converges (stays within a short radius of c for all n), color the chosen value of c black. If it diverges color c blue or yellow. Now repeat this for all c points on a grid.

الفصل الثامن: تعليم التفكير الحوسبي للجميع

(1) Wing (2006).

(2) Tedre, Simon, and Malmi (2018).

(3) Minsky (1970).

(4) Knuth (1974).

(5) Bolter (1984).

(6) Abelson and Sussman (1996).

(7) Guzdial (2015).

(8) Denning (2017).

(9) See <http://csfieldguide.org.nz> and <http://csunplugged.org>.

الفصل التاسع: مستقبل الحوسبة

(1) Denning and Lewis (2017).

(2) McGeoch (2014).

(3) See Walter Tichy's interview with Catherine McGeoch, *Ubiquity* July 2017, for a worked example of an Ising equation and its encoding into a form for the D-wave machine to solve, <https://ubiquity.acm.org/article.cfm?id=3084688>.

(4) Adleman (1994).

(5) Kurzweil (2006).

(6) Wolfram (2002).

(7) In April 2016, *Scientific American* magazine reported on a symposium of physicists and philosophers discussing the whole-world-is-computer hypothesis, giving the impression that they take more delight in entertaining themselves with the hypothesis than in the hypothesis itself. See <https://www.scientificamerican.com/article/are-we-living-in-a-computer-simulation/>.

مراجع وقراءات إضافية

الفصل الثاني: أساليب الحوسبة

- Davis, Martin. (2012). *The Universal Computer: The Road from Leibniz to Turing*. CRC Press.
- Grier, David A. (2005). *When Computers Were Human*. Princeton University Press.
- Hodges, Andrew. (1983). *Alan Turing: The Enigma*. Vintage Books.
- Priestley, Mark. (2011). *A Science of Operations: Machines, Logic and the Invention of Programming*. Springer-Verlag.
- Rapaport, William J. (2018). *Philosophy of Computer Science*. An online book draft, <https://cse.buffalo.edu/~rapaport/Papers/phics.pdf>.
- Williams, Michael R. (1997). *A History of Computing Technology*. 2nd edition. IEEE Computer Society Press.

الفصل الثالث: آلات الحوسبة

- Aspray, William, ed. (1990). *Computing Before Computers*. Iowa State University Press.
- Campbell-Kelly, Martin, and William Aspray. (2004). *Computer: A History of the Information Machine*. 2nd edition. Westview Press.

- Ceruzzi, Paul E. (2003). *A History of Modern Computing*. 2nd edition. MIT Press.
- Cortada, J. W. (1993). *Before the Computer: IBM, NCR, Burroughs, and Remington Rand and the Industry They Created, 1865-1956*. Princeton University Press.
- Williams, Michael R. (1997). *A History of Computing Technology*. 2nd edition. IEEE Computer Society Press.

الفصل الرابع: علوم الكمبيوتر

- Arden, Bruce W., ed. (1980). *What Can Be Automated? Computer Science and Engineering Research Study*. MIT Press.
- Daylight, Edgar G. (2012). *The Dawn of Software Engineering: From Turing to Dijkstra*. Lonely Scholar.
- Dijkstra, Edsger. W. (1974). Programming as a discipline of mathematical nature. *American Mathematical Monthly* 81 (6): 608–612.
- Knuth, Donald E. (1974). Computer science and its relation to mathematics. *American Mathematical Monthly* 81 (April): 323–343.
- Knuth, Donald E. (1985). Algorithmic thinking and mathematical thinking. *American Mathematical Monthly* 92 (March): 170–181.
- Mahoney, Michael Sean. (2011). *Histories of Computing*. Harvard University Press.
- Metropolis, N., J. Howlett, and Gian-Carlo Rota, eds. (1980). *A History of Computing in the Twentieth Century: A Collection of Essays with Introductory Essay and Indexes*. Academic Press.
- Newell, Alan, Alan J. Perlis, and Herbert A. Simon. (1967). Computer science. *Science* 157 (3795): 1373–1374.
- Simon, Herbert A. (1969). *Sciences of the Artificial*. MIT Press.
- Smith, Brian C. (1998). *On the Origin of Objects*. MIT Press.

الفصل الخامس: هندسة البرمجيات

- Alexander, Christopher. (1979). *The Timeless Way of Building*. Oxford University Press.
- Brooks, Frederick P. Jr. (1975). *The Mythical Man-Month*. (20th anniversary edition, 1995). Addison-Wesley.
- Brooks, Frederick P. Jr. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE Computer* 20 (4): 10–19.
- Campbell-Kelly, Martin. (2003). *From Airline Reservations to Sonic the Hedgehog*. MIT Press.
- Denning, Peter. (2018). Interview with David Parnas. *Communications of ACM* 61 (6) (June).
- Ensmenger, Nathan L. (2010). *The Computer Boys Take Over: Computers, Programmers, and the Politics of Technical Expertise*. MIT Press.
- Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- Koen, Billy V. (2003). *Discussion of the Method: Conducting the Engineer's Approach to Problem Solving*. Oxford University Press.
- Lampson, Butler. (1983). Hints for computer system design. *Proc. ACM Symposium on Operating Systems Principles*, 33–48.
- Metropolis, N., J. Howlett, and Gian-Carlo Rota, eds. (1980). *A History of Computing in the Twentieth Century: A Collection of Essays with Introductory Essay and Indexes*. Academic Press.
- Mitcham, Carl. (1994). *Thinking Through Technology: The Path Between Engineering and Philosophy*. University of Chicago Press.
- Saltzer, Jerome H., and Michael D. Schroeder. (1975). Protection of information computer systems. *Proceedings of the IEEE* 63 (9) (September): 1278–1308.

- Stokes, Donald E. (1997). *Pasteur's Quadrant—Basic Science and Technological Innovation*. Brookings Institution Press.
- Wirth, Niklaus. (2008). A brief history of software engineering. *IEEE Annals of the History of Computing*, 30 (3): 32–39.

الفصل السادس: التصميم لتلبية احتياجات الإنسان

- Brooks, Frederick P. Jr. (1975). *The Mythical Man-Month*. (20th anniversary edition, 1995). Addison-Wesley.
- Denning. Peter. (2016). Software quality. *Communications of ACM* 59 (9) (September): 23–25.
- Forsythe, George E. (1966). *A University's Educational Program in Computer Science*. Technical Report No. CS39, May 18, 1966. Stanford University: Computer Science Department, School of Humanities and Sciences.
- Grudin, Jonathan. (1990). The computer reaches out: The historical continuity of interface design. In *CHI '90: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 261–268. ACM.
- Landwehr, Carl, et al. 2017. Software Systems Engineering Programmes: A Capability Approach. *Journal of Systems and Software* 125: 354–364.
- Leveson, Nancy. (1995). *SafeWare: System Safety and Computers*. Addison-Wesley.
- Norman, Donald A. (1993). *Things That Make Us Smart*. Basic Books.
- Norman, Donald A. (2013). *The Design of Everyday Things*. First edition 1983. Basic Books.
- Parnas, Dave, and Peter Denning. (2018). An interview with Dave Parnas. *Communications of ACM* 61 (6).
- Winograd, Terry, and Flores, F. (1987). *Understanding Computers and Cognition*. Addison-Wesley.

الفصل السابع: العلوم الحوسبيّة

- Aho, Al. (2011). Computation and computational thinking.
- Akera, Atshushi. (2007). *Calculating a Natural World: Scientists, Engineers, and Computers During the Rise of U.S. Cold War Research*. MIT Press.
- Baltimore, David. (2001). How biology became an information science. In *The Invisible Future*. Peter Denning, ed., pp. 43–46. McGraw–Hill.
- Denning, Peter. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM* 60 (6) (June): 33–39.
- Wilson, Ken. (1989). Grand challenges to computational science. In *Future Generation Computer Systems*, pp. 33–35. Elsevier.
- Wolfram, Stephen. (2002). *A New Kind of Science*. Wolfram Media.

الفصل الثامن: تعليم التفكير الحوسبي للجميع

- Abelson, Harold, and Gerald J. Sussman. (1996). *Structure and Interpretation of Computer Programs*. 2nd edition. MIT Press.
- Bolter, J. David. (1984). *Turing's Man: Western Culture in the Computer Age*. University of North Carolina Press.
- Denning, Peter. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM* 60 (6) (June): 33–39.
- Guzdial, Mark. (2015). *Learner-Centered Design of Computing Education: Research on Computing for Everyone. Synthesis Lectures on Human-Centered Informatics*. Morgan & Claypool.
- Kestenbaum, David. (2005). The challenges of IDC: What have we learned from our past? *Communications of the ACM* 48 (1): 35–38. [A conversation with Seymour Papert, Marvin Minsky, Alan Kay].
- Knuth, Donald E. (1974). Computer science and its relation to mathematics. *American Mathematical Monthly* 81 (April): 323–343.

- Lockwood, James, and Aidan Mooney. (2017). *Computational Thinking in Education: Where Does It Fit? A Systematic Literary Review*. Technical report, National University of Ireland Maynooth.
- Minsky, Marvin. (1970). Form and content in computer science. *Journal of the ACM* 17 (2): 197–215.
- Tedre, Matti, Simon, and Lauri Malmi. (2018). Changing aims of computing education: a historical survey. *Computer Science Education*, June.
- Wing, Jeanette M. (2006). Computational thinking. *Communications of the ACM* 49 (3): 33–35.

الفصل التاسع: مستقبل الحوسبة

- Adleman, Leonard M. (1994). Molecular computation of solutions to combinatorial problems. *Science* 266 (5187): 1021–1024.
- Brynjolfsson, E., and McAfee, A. (2014). *The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies*. W. W. Norton & Company.
- Denning, Peter. J., and Ted G. Lewis. (2017). Exponential laws of computing growth. *Communications of ACM* 60 (1) (January): 54–65.
- Friedman, Thomas. (2016). *Thank You for Being Late*. Farrar, Straus and Giroux.
- Kelly, Kevin. (2017). *The Inevitable: Understanding the 12 Technological Forces That Will Shape Our Future*. Penguin Books.
- Kurzweil, Ray. (2006). *The Singularity Is Near*. Penguin Books.
- McGeoch, Catherine. (2014). *Adiabatic Quantum Computation and Quantum Annealing. Synthesis Series on Quantum Computing*. Morgan & Claypool.
- Wolfram, Stephen. (2002). *A New Kind of Science*. Wolfram Media.

