

المحاضرة 03 · رسم منطق الحل

03

المخططات الانسيابية: ارسم القرار والتكرار

متى يصبح الرسم أسرع من قراءة خمسين سطرًا؟

تطبيق عملي

أمثلة C++17

رسم توضيحي

فهم قبل الحفظ

مستوحاة من برمجة 1 · صفحات المصدر 15-30 · إعداد تربوي: م.براءة علي رحيمو

++C

01 · المشكلة أولاً

قبل التعريف... لماذا نحتاج هذه الفكرة؟

متى يصبح الرسم أسرع من قراءة خمسين سطرًا؟

مشهد من الواقع

في نظام خصومات متجر: طلبية كبيرة تحصل على 30%، والمتوسطة 20%، والباقي بلا حسم. الرسم يُظهر التفرعات والعودة بسرعة.

أين المشكلة في الطريقة الساذجة؟

عند كثرة IF والتكرارات قد نفقد صورة المسار الكامل؛ يصعب تمييز بداية الدورة ونهايتها وحالات القرار.

كيف تساعدنا الفكرة؟

نحوّل الخوارزمية إلى رموز: يضاوي للبداية والنهاية، مستطيل للمعالجة، مُعين للقرار، ومتوازي أضلاع للدخل/الخروج.

شَبَّهها بما تعرفه: خريطة المترو لا تحكي كل التفاصيل، لكنها توضح أين تتفرع الطرق وكيف تصل إلى النهاية.

لا تحفظ كلمة جديدة قبل أن تعرف المشكلة التي وُجدت لتساعد في حلها.

02 · النموذج العقلي

اجعل المفهوم مرئيًا

ثلاث أفكار رئيسية ثم مسار واضح يربط بينها.

◇ Decision

شرط نتيجه نعم/لا ويحدد مسارًا.

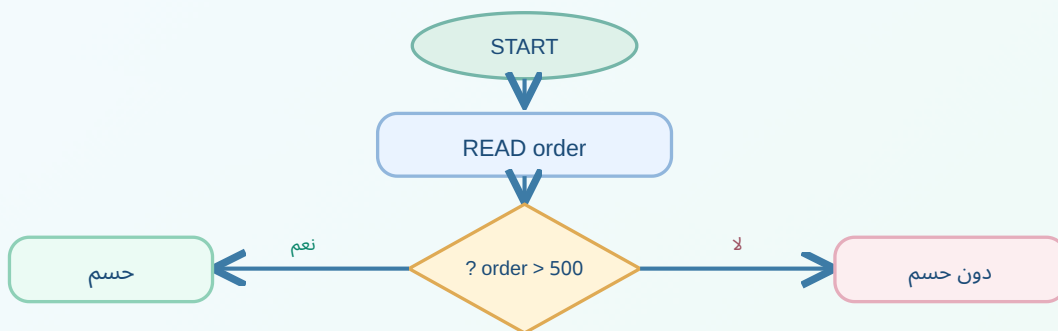
▢ Process

تنفيذ عملية حسابية أو إسناد.

🔄 Loop

سهم رجوع يجعل خطوات معينة تتكرر بشرط.

الرسم التوضيحي: كيف تعمل الفكرة؟



السؤال الأهم: متى يصبح الرسم أسرع من قراءة خمسين سطرًا؟

03 · مثال عملي أول

شغل وافهم كل سطر

خصم يعتمد على قيمة الطلبية

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    double order = 750;
    double discount = 0;
    if (order > 1000) discount = 0.30;
    else if (order > 500) discount = 0.20;
    cout << "Pay = " << order*(1-discount);
}
```

الخرج المتوقع Pay = 600

لماذا كتبنا البرنامج بهذا الشكل؟

ارسم مُعينًا لكل شرط، ومسارًا يصل بعد حساب الحسم إلى نقطة مشتركة للطباعة.

💡 قبل تجربة البرنامج: حاول توقع النتيجة على الورق، ثم قارن توقُّعك بالخرج الفعلي.

04 · مثال إضافي

نفس المبدأ، حالة استخدام مختلفة

رسم دورة لجمع 1 حتى N

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int n = 4, sum = 0, i = 1;
    while (i <= n) {
        sum += i;
        ++i;
    }
    cout << sum;
}
```

الخرج المتوقع 10

ما الذي نتعلمه من المثال الثاني؟

في المخطط نتحقق من $i \leq n$ ؛ إن تحقق الشرط نجمع ونزيد ثم نعود، وإلا نطبع.

جرب تغيير قيمة واحدة فقط، ثم توقّع أثرها: هل تغيّر مسار البرنامج أم الناتج فقط؟

05 · مختبر الأخطاء

الخطأ ليس للحفظ... اكتشف سببه

قارن الكود الخاطئ بالصحيح ولا تكتفِ بمعرفة الإجابة.

الكود الذي يحتاج انتباهًا

ثم طريق «لا» بلا استمرار $x > 0$: قرار

تصحيح وتفسير

نعم/لا، وكلاهما يصل لنقطة متابعة $x > 0 \rightarrow$: قرار

لماذا يُعد هذا خطأ؟

كل فرع يجب أن يصل إلى نهاية واضحة أو حلقة مضبوطة.

جرب بنفسك

تحدي 01

ارسم مخططًا لفحص زوجية عدد.

تحدي 02

ارسم دورة لحساب مجموع درجات 5 طلاب.

تحدي 03

راجع حدود حسم المتجر عند 500 و1000 تحديدًا.

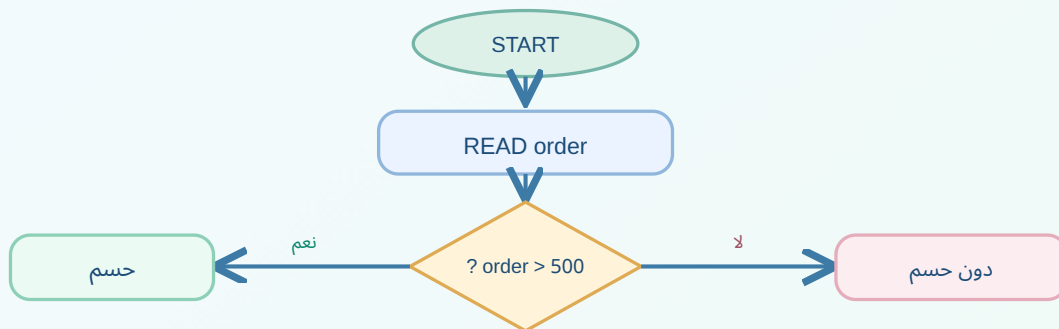
06 · تثبيت الفهم

ماذا يجب أن يبقى في ذاكرتك؟

لا تقرأ القواعد فقط؛ اربط كل قاعدة بخطأ تمنعه أو مشكلة تحلها.

- ✓ الرسم يساعدنا على رؤية كل المسارات.
- ✓ مُعين القرار ينتج فرعين أو أكثر بحسب المسألة.
- ✓ أخطر خطأ في المخطط: مسار لا ينتهي أو لا يلتقي بالنهاية.

الرسم التوضيحي: كيف تعمل الفكرة؟



تحضير الكويز التفاعلي

قبل فتح اختبار هذه المحاضرة، حاول شرح المثال الأول لشخص آخر دون النظر إلى الشيفرة. إذا فهم لماذا تعمل الخطوات، فأنت جاهز.

الجسر إلى المحاضرة التالية

بعد معرفة كيف نخطط، سنعرف لماذا نستخدم لغات مثل ++C لترجمة الخطوات.

المرجع: «برمجة 1» لم. بسمه سراقبي، الصفحات 15–30 من الملف المرفق. الأمثلة والشروح الإضافية صيغت للأغراض التعليمية وتستخدم ++C.