

المحاضرة 06 · لغة المنطق

06

العمليات والتعابير: لماذا تختلف نتائج الحساب؟

لماذا تعطي 7/2 أحياناً 3 وأحياناً 3.5؟

تطبيق عملي

أمثلة C++17

رسم توضيحي

فهم قبل الحفظ

مستوحاة من برمجة 1 · صفحات المصدر 48-59 · إعداد تربوي: م.براءة علي رحيمو

++C

01 · المشكلة أولاً

قبل التعريف... لماذا نحتاج هذه الفكرة؟

لماذا تعطي 7/2 أحياناً 3 وأحياناً 3.5؟

مشهد من الواقع

تخيل متجرًا يحسب خصمًا ونسبة ضريبة. نتيجة القسمة الخاطئة تغيّر الفاتورة، وخلط && مع || قد يمنح خصمًا لزبون غير مستحق.

أين المشكلة في الطريقة الساذجة؟

الكتابة الرياضية لا تُنسخ حرفيًا: في ++C توجد أنواع بيانات وأولويات وأشكال مختلفة للزيادة والمقارنة.

كيف تساعدنا الفكرة؟

نفهم العمليات قبل حفظ رموزها: / و %، الإسناد = مقابل المقارنة ==، و && و || و ! والتحويل الصريح.

شبهها بما تعرفه: قواعد الأولوية مثل ترتيب العمليات الحسابية الذي تعلمته في المدرسة؛ لكن نوع العدد هنا يؤثر أيضًا على الناتج.

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.

 $2 / 7.0 \neq 2 / 7$

القسمة بين int تزيل الكسر، بينما دخول double يغيّر نوع الحساب.

= مقابل ==

الأولى إسناد قيمة؛ الثانية سؤال مساواة يعيد true أو false.

&& و ||

تجميع شروط «كلها» مقابل «واحد على الأقل»، مع قصر التقييم Short-circuit.

خريطة المفهوم: من الفكرة إلى التنفيذ



السؤال الأهم: لماذا تعطي 7/2 أحياناً 3 وأحياناً 3.5؟

03 · مثال عملي أول

شغل وافهم كل سطر

قسمة صحيحة مقابل حقيقية

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int a = 7, b = 2;
    cout << a / b << '\n';
    cout << static_cast<double>(a) / b << '\n';
    cout << a % b;
}
```

الخرج المتوقع 3 | 3.5 | 1

لماذا كتبنا البرنامج بهذا الشكل؟

تغيير نوع معامل واحد إلى double يمنع ضياع الجزء الكسري. % يعطي الباقي للأعداد الصحيحة.

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

04 · مثال إضافي

نفس المبدأ، حالة استخدام مختلفة

حقّ الدخول عبر شرطين

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int age = 20;
    bool hasTicket = true;
    bool canEnter = age >= 18 && hasTicket;
    cout << (canEnter ? "Enter" : "Denied");
}
```

الخرج المتوقع Enter

ما الذي نتعلمه من المثال الثاني؟

لا تكفي السن وحدها إذا كانت سياسة الدخول تتطلب أيضًا وجود تذكرة.

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

```
if (score = 100) cout << "Top";
```

تصحيح وتفسير

```
if (score == 100) cout << "Top";
```

لماذا يُعد هذا خطأ؟

الإسناد داخل شرط قد يعيد قيمة حقيقية ويغيّر score؛ المقارنة هي المطلوبة.

جرب بنفسك

تحدي 01

احسب 10/4 و 10/4.0 و اشرح الاختلاف.

تحدي 02

أعد كتابة خصم للطلاب: `age<25 && hasCard`.

تحدي 03

ميّز `x++` عن `++x` في سطر `cout`.

06 · تثبيت الفهم

ماذا يجب أن يبقى في ذاكرتك؟

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

✓ استخدم الأقواس لتوضيح القصد بدل الاتكال على الذاكرة.

✓ نوع المعامل يحدد نوع القسمة.

✓ استخدم `===`؟ لا، المقارنة في C++ هي `==`.

خريطة المفهوم: من الفكرة إلى التنفيذ



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

في المحاضرة التالية نحول التعابير إلى برنامج كامل قابل للإدخال والإخراج.

المرجع: «برمجة 1» لم. براءة سراقبي، الصفحات 48–59 من الملف المرفق، الأمثلة والشروح الإضافية صيغت للأغراض التعليمية وتستخدم C++17.