

المحاضرة 08 · منطق الاختيار

08

القرارات: if و else و switch

كيف تجعل برنامجك يتصرف بشكل مختلف حسب الحالة؟

تطبيق عملي

أمثلة C++17

رسم توضيحي

فهم قبل الحفظ

مستوحاة من برمجة 1 · صفحات المصدر 73-89 · إعداد تربوي: م.براءة علي رحيمو

++C

01 · المشكلة أولاً

قبل التعريف... لماذا نحتاج هذه الفكرة؟

كيف تجعل برنامجك يتصرف بشكل مختلف حسب الحالة؟

مشهد من الواقع

تطبيق تسجيل الدخول يسمح بالدخول إن كان الرمز صحيحًا، وإلا يعرض رسالة. وبطاقة تقييم تحتاج فئات عديدة.

أين المشكلة في الطريقة الساذجة؟

برنامج ينفذ الأسطر دائمًا بالترتيب غير كافٍ حين تتعدد الاحتمالات؛ نحتاج شروطًا واضحة لا تتداخل بطريق الخطأ.

كيف تساعدنا الفكرة؟

نستخدم if لقرار، و else if لشرائح مرتبة، و switch لمقارنة قيمة بفروع ثابتة متعددة.

شبهها بما تعرفه: بوابة التفتيش: تُراجع شرطًا ثم تسمح بالمرور أو توجّه المستخدم لمسار آخر.

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.

if / else

تنفيذ مسار عند تحقق الشرط ومسار بديل عند عدم تحققه.

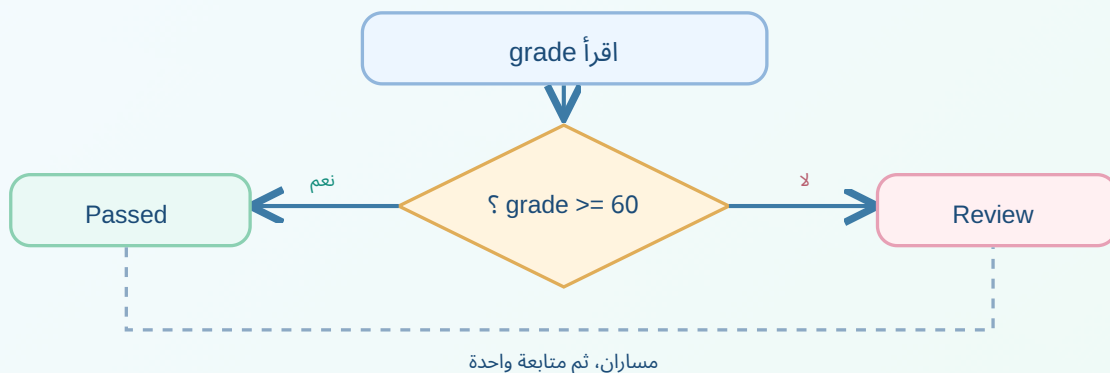
else if

تسلسل قرارات؛ أول شرط صحيح ينفذ في السلسلة.

switch / case

اختيار فرع من قيمة ثابتة النوع المناسب، مع break لمنع السقوط للفروع التالية.

الرسم التوضيحي: كيف تعمل الفكرة؟



السؤال الأهم: كيف تجعل برنامجك يتصرف بشكل مختلف حسب الحالة؟

03 · مثال عملي أول

شغل وافهم كل سطر

قرار النجاح برسالة مختلفة

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int grade = 72;
    if (grade >= 90) cout << "Excellent";
    else if (grade >= 60) cout << "Passed";
    else cout << "Try again";
}
```

الخرج المتوقع Passed

لماذا كتبنا البرنامج بهذا الشكل؟

فحص 90 يأتي أولاً. إذا تحقق 60 بعد فشل 90، ينفذ فرع Passed فقط.

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

04 · مثال إضافي

نفس المبدأ، حالة استخدام مختلفة

اختيار قائمة بواسطة switch

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int choice = 2;
    switch (choice) {
        case 1: cout << "Start"; break;
        case 2: cout << "Settings"; break;
        default: cout << "Unknown";
    }
}
```

الخرج المتوقع Settings

ما الذي نتعلمه من المثال الثاني؟

عند مطابقة case 2 يطبع Settings؛ يتوقف break عن النزول إلى فروع أخرى.

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 (grade >= 60) cout << "Pass";  
if (grade >= 90) cout << "Excellent";
```

تصحيح وتفسير

```
if (grade >= 90) cout << "Excellent";  
else if (grade >= 60) cout << "Pass";
```

لماذا يُعد هذا خطأ؟

قد تطبع الصيغة الأولى رسالتين للطالب المتفوق؛ السلسلة المتبادلة تجعل التصنيف واحدًا.

جرب بنفسك

تحدي 01

أنشئ برنامجًا لاختيار الأكبر بين ثلاثة أعداد.

تحدي 02

تحقق من حدود 59 و60 و89 و90 في التصنيف.

تحدي 03

جرب حذف break من switch واطرح ما يحدث.

06 · تثبيت الفهم

ماذا يجب أن يبقى في ذاكرتك؟

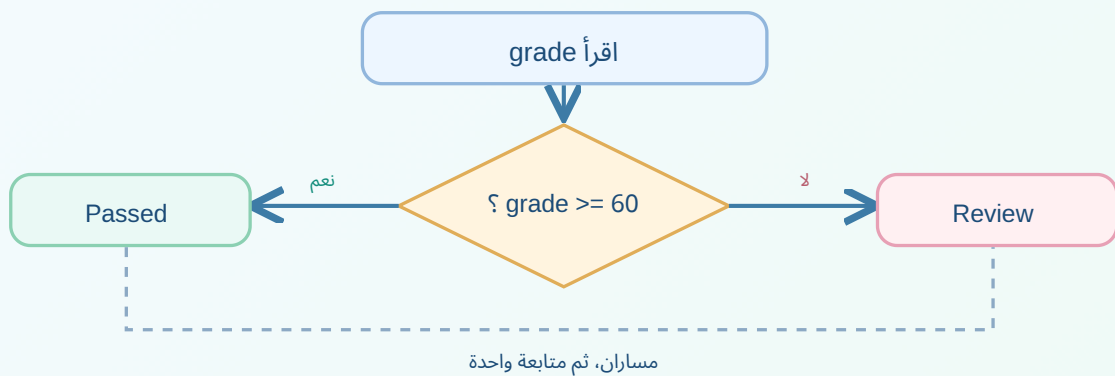
لا تقرأ القواعد فقط؛ اربط كل قاعدة بخطأ تمنعه أو مشكلة تحلها.

✓ ترتيب الشروط يغير النتائج.

✓ استخدم else إذا كانت الحالات متبادلة الاستبعاد.

✓ أضف default لمعالجة القيم غير المعروفة.

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

سنتعرف بعدها إلى تنفيذ المهمة نفسها مرات متعددة باستخدام الحلقات.

المرجع: «برمجة 1» لم. براءة سراقبي، الصفحات 73–89 من الملف المرفق. الأمثلة والشروح الإضافية صيغت للأغراض التعليمية وتستخدم C++17.