

المحاضرة 07 · دورة التنفيذ

07

أول برنامج C++: من main إلى cin و cout

كيف تتحول فكرة بسيطة إلى برنامج يقرأ ويكتب؟

تطبيق عملي

أمثلة C++17

رسم توضيحي

فهم قبل الحفظ

مستوحاة من برمجة 1 · صفحات المصدر 60-72 · إعداد تربوي: م.براءة علي رحيمو

C++

01 · المشكلة أولاً

قبل التعريف... لماذا نحتاج هذه الفكرة؟

كيف تتحول فكرة بسيطة إلى برنامج يقرأ ويكتب؟

مشهد من الواقع

تريد برنامجًا يسأل عن طول وعرض غرفة ثم يعرض المساحة. من أين يبدأ التنفيذ؟ ومن أين تأتي الأرقام؟

أين المشكلة في الطريقة الساذجة؟

من دون هيكل واضح لا يعرف المترجم أين يبدأ البرنامج، ومن دون إدخال/إخراج لا يتواصل البرنامج مع المستخدم.

كيف تساعدنا الفكرة؟

نكتب include المناسب، main، ثم نقرأ باستخدام cin ونظهر النتيجة باستخدام cout، مع تعليقات مفيدة.

شبهها بما تعرفه: main مثل باب الدخول إلى ورشة، cin صندوق استقبال، cout لوحة عرض النتائج.

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.

<include <iostream#

تضمنين ترويسة أدوات الإدخال والإخراج القياسية.

()int main

نقطة بداية التنفيذ ويعيد عددًا صحيحًا إلى بيئة التشغيل.

cin / cout

دفق إدخال عبر << ودفق إخراج عبر >>.

خريطة المفهوم: من الفكرة إلى التنفيذ**السؤال الأهم:** كيف تتحول فكرة بسيطة إلى برنامج يقرأ ويكتب؟

03 · مثال عملي أول

شغل وافهم كل سطر

أول برنامج مستقل

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    int x = 4, y = 6;
    int sum = x + y;
    cout << "Sum = " << sum << '\n';
    return 0;
}
```

الخرج المتوقع Sum = 10

لماذا كتبنا البرنامج بهذا الشكل؟

التنفيذ يدخل إلى main؛ يخصص قيمًا ويحسبها ثم يرسل النص والنتيجة إلى الشاشة.

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

04 · مثال إضافي

نفس المبدأ، حالة استخدام مختلفة

قراءة مدخلات فعلية

C++17 / source.cpp

```
#include <iostream>
using namespace std;
int main() {
    double width, height;
    cout << "Width and height: ";
    cin >> width >> height;
    cout << "Area = " << width * height << '\n';
}
```

الخرج المتوقع Area = 15 عند إدخال 5 3 تظهر

ما الذي نتعلمه من المثال الثاني؟

المعامل << يستقبل مدخلات، بينما >> يرسل بيانات إلى مجرى الإخراج.

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

```
#include <iostream.h>
void main() {}
```

تصحيح وتفسير

```
#include <iostream>
int main() { return 0; }
```

لماذا يُعد هذا خطأ؟

المصدر يحتوي أمثلة قديمة لبيئات ++Turbo C; الأمثلة الحديثة تستخدم الترويسة القياسية وint main.

جرب بنفسك

تحدي 01

بدل الإدخال ليتلقى نصف قطر دائرة واحسب المساحة.

تحدي 02

جرب نسيان ; وحدد الخطأ الذي يحدث.

تحدي 03

استخدم \n لفصل سطرين في الخرج.

06 · تثبيت الفهم

ماذا يجب أن يبقى في ذاكرتك؟

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

✓ main هي نقطة بدء التنفيذ القياسية.

✓ cin لإحضار البيانات؛ cout لإظهار النتائج.

✓ اكتب C++17 وليس أوامر بيئات قديمة مثل conio.h.

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

سنتعلم بعد ذلك كيف يختار البرنامج مسارًا بناءً على شروط.

المرجع: «برمجة 1» لم. بسمه سراقبي، الصفحات 60-72 من الملف المرفق، الأمثلة والشروح الإضافية صيغت للأغراض التعليمية وتستخدم C++17.