InsideC++



InsideC++26481

InsideC++26481

Last time we left of with:

The Contract Drama

Potentially the biggest time sink ever

Today's program

WG21 Buzios report

- Topics other than contracts
- Contracts
- What does it mean for C++26

Did ISO National Bodies prevent WG21 from shipping an incomplete feature?

InsideC++ FAQ

InsideC++ FAQ

- Show notes

InsideC++ FAQ

- Show notes
- Did the SwedenCpp channel format change?

Topics other than contracts

- 26 fixes
- 29 work

Contracts, a strange first day

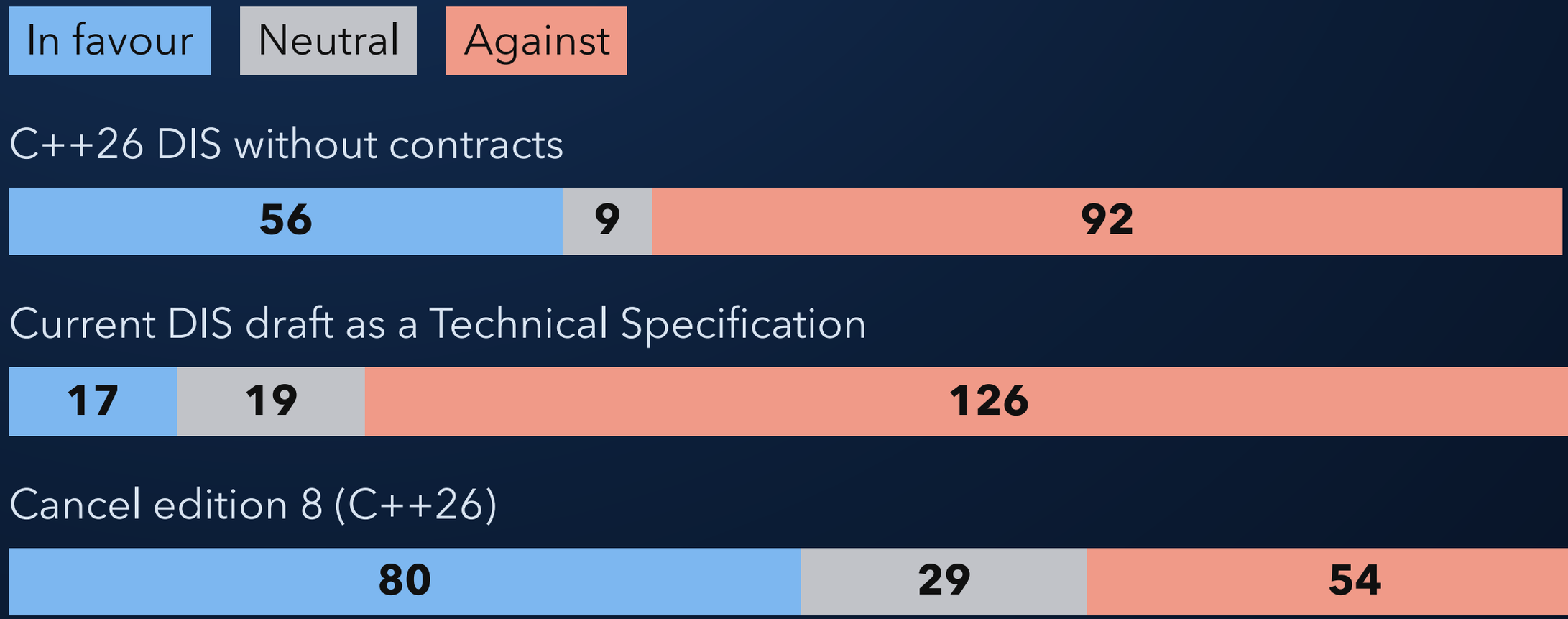
- Remove contracts
- Put C++ in TS state
  meaning: there is no standard anymore 🤯
- No C++26

Contracts, a strange first day

- Remove contracts
- Put C++ in TS state
  meaning: there is no standard anymore 🤯
- No C++26

Not an option, the Swedish suggestion
→ ship 26 with no contracts + Contracts TS.

Contracts, a strange first day



Separate polls. All bars use the same scale: 0–170 votes.

Contracts questions

Since removal was not an option

- Constification
- Virtual functions
- Always enforce

Contracts

- Always enforce
  - Quick Fix: P4451
  - Fix the Fix: D4489R0

The missed opportunity

Full adoption of D4489R0

The missed opportunity

Full adoption of D4489R0

```
void fun(int x) pre( x < 22 ) {
    //....
}
```

What should be the default behavior?

Wide or narrow contract?

Can we wait until 29 to find out?

The missed opportunity

Full adoption of D4489R0

```
#include <contracts>

void f(int x)
  pre<cc::enforced>(x > 0)        // OK, terminating
  pre<cc::impldefined>(x < 100) // OK, as in the DIS
  pre(x != 7);                    // error: no label
```

Can we wait until 29 to find out?

The missed opportunity

Full adoption of D4489R0

Wait until 29 to find out was NOT an option!

My favorite arguments 🤯

- Easier to teach compiler flags and implementation defined than whats between angle brackets.
- Better a wrong default than no default

Summary

Some companies stressed the need for contracts that are always enforced.

This was also a key concern raised by National Bodies.

Summary

Thanks to the ISO National Bodies for challenging an incomplete design.

Their objections prevented C++26 from shipping contracts without this basic guarantee and secured at least one potentially useful addition.

Summary

Is C++26 good to ship?

⇒ Still an open question ⇐

Thank you for today



InsideC++26481