

Константин Игоревич Вадимов, Syntacore

Лаборатория RISC-V со среды

От C к C++

LRU - cache - last recent used



Реализация на C

Тема: как писать список + тесты, реализация + тесты, грайпер + тесты

Действия: „Свой собственный проект“

Стиль: два проекта на модуль: header + src file

! Необходимо сразу написать переиспользуемые абстракции (не std списки например)

Реализация на C++

- Условно можно считать в C
- Одна из главных особенностей: объединение данных и методов их обработки
- Другой тон: писать явно this (конечу 2-го семестра)
- Обобщение данных и методов: template <typename T> (в C: макросы)
- При этом очень важной особенностью является
- БENCHMARK: quick bench
- В cpp quicksort быстрее: компилятор умнее (т.к. qsort не предкомпилирован: он интерпретируется).
- Также подход к std позволяет писать кучу обобщенных контейнеров (cpp reference)
- В LRU не надо писать ни hashtable, ни list
- Создавать модификаторы максимально близко к т. использованию
- Семантика значений: better ceda как int

Д/З: срабне ARC/20/LF4/LERS с игеаюном конем (знаеі дыггынзе)

Сраггепі C++ 20

Сраггепып 4-th edition

human C++ primer

лекция 3.

Объявление и определение

`int x` - определение (и объявление!)

`struct S;` - объявление

`extern void foo();` - определение

`extern S *ps;` - объявление

`int bar();` - объявление

`struct T {int x;};` - определение

`extern int y=0;` - определение

Ключевые слова

`static` - единичная приватная для данного ед. трансляции

`extern` - указание linkage specifier (указывает всем)

Примеры:

`static` в ф-ии - модальное пер-аз с од-го вычисления ф-ии

`static` в struct - ссылка на всех объектов пер-аз

Удобно для дефайнции, он гарантирует наличие одного экземпляра во всех единичных трансляциях, в к-рые попадает.

`inline` ф-ии - одна на все ед. трансляции - может быть только в header или ф-ии

`static` ф-ии - по одному на каждую ед. трансляции - вызывает размер про-м

`static inline` \equiv `static` & этот символ

