Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

# Статья № 1

ЗАО ”Сайенс”

**Работу выполнил:**
Поздняков А.А.
Группа:
3530904/00104
**Руководитель:**
Касилов В.А.

Санкт-Петербург
2023

# Содержание

**1. Введение** **3**

**2. Требования** **4**
2.1. Спецификация интерфейса . . . . . 4
2.1.1. Шаблон класса polylru::LRU< Key, Value > . . . . . 4
2.2. Спецификация поведения . . . . . 5

**3. Список литературы** **7**

# 1. Введение

Цель работы - разработать многопоточный кэш с методом вытеснения LRU, сравнить собственную реализацию с реализациями в библиотеках с открытым исходным кодом.

Для достижения поставленной цели в работе были поставленны следующие задачи:

1. Изучить подходы к реализации LRU кэша
2. Разработать спецификацию интерфейса разрабатываемого класса
3. Разработать тесты, описывающие поведение разрабатываемого класса
4. Реализовать класс кэша
5. Сравнить собственную реализацию с реализациями в свободно доступных библиотеках

# 2. Требования

## 2.1. Спецификация интерфейса

### 2.1.1. Шаблон класса polylru::LRU< Key, Value >

Данный шаблон класса реализует кэш с методом вытесенения LRU(Least recently used).
`#include <lru.hpp>`

**Открытые типы**

- typedef std::pair< Key, Value > **value_type**
- typedef Key **key_type**
- typedef Value **mapped_type**

**Открытые члены**

- **LRU** (size_t capacity)
- Value get (Key key)

  *Возвращает значение, установленное для указанного ключа.*
- void set (Key key, Value value)

  *Устанавлияет в кэше значение с указанным ключом.*
- bool contains (Key key) const noexcept

  *Проверяет, сохранён ли в кэше элемент с указанным ключом.*
- size_t size () const noexcept

  *Возвращает текущее количество элементов в кэше.*
- size_t capacity () const noexcept

  *Возвращает текущую ёмкость кэша.*
- void resize (size_t new_cap)

  *Изменяет ёмкость кэша.*

**Подробное описание**

**template<typename Key, typename Value>**
**class polylru::LRU< Key, Value >**

Данный шаблон класса реализует кэш с методом вытесенения LRU(Least recently used).

Элементы хранятся ассоциативно, в формате ключ-значение, ёмкость кэша задаётся при создании объекта. Размером кэша считается количество элементов, сохранённое в контейнере. Если размер кэша равен его ёмкости, он считается **заполненным**. Возраст элементов определяется количеством обращений к кэшу, в течение которого они не были использованы. Добавление элемента с новым ключом в заполненный кэш влечёт за собой вытесенение из кэша по методу LRU, самый старый элемент удаляется.

Функция resize(size_t) позволяет динамически изменить размер контейнера.

**Методы**

**get()** `template<typename Key , typename Value >`
`Value polylru::LRU< Key, Value >::get (`
`Key` *`key`* `)`

Возвращает значение, установленное для указанного ключа.

В случае, если элемент с указанным ключом присутствует в кэше, возвращает значение элемента. Попытка доступа по неизвестному ключу влечёт неопределенное поведение.

**resize()** `template<typename Key , typename Value >`
`void polylru::LRU< Key, Value >::resize (`
`size_t` *`new_cap`* `)`

Изменяет ёмкость кэша.

Метод уменьшает размер кэша в случае, если переданный размер меньше текущего, при этом происходит удаление самых старых элементов. В случае, если переданный размер больше текущего, кэш увеличивается до указанного размера. Если переданный размер равен текущему, изменений не происходит.

**set()** `template<typename Key , typename Value >`
`void polylru::LRU< Key, Value >::set (`
`Key` *`key,`*
`Value` *`value`* `)`

Устанавлияет в кэше значение с указанным ключом.

В случае, если элемент с указанным ключом сохранён в кэше, устанавливает новое значение для данного ключа. В случае, если элемент с указанным ключом отсутствует в кэше, добавляет элемент в кэш. Добавление элемента в заполненный кэш приводит к вытеснению самого старого элемента.

**size()** `template<typename Key , typename Value >`
`size_t polylru::LRU< Key, Value >::size [noexcept]`

Возвращает текущее количество элементов в кэше.

В случае, если размер кэша равен его ёмкости, добавление элемента с новым ключом привёдет к вытеснению самого старого элемента.

Объявления и описания членов класса находятся в файле:

- lib/src/lru/lru.hpp

## 2.2. Спецификация поведения

- ЕСЛИ конструктор вызывается с отрицательным параметром size, ТО выбрасывается исключение std::length_error
- ЕСЛИ конструктор вызывается с параметром size равным 0, ТО выбрасывается исключение std::length_error
- ЕСЛИ конструктор вызывается с положительным параметром size, ТО создаётся новый объект
- ЕСЛИ ёмкость кэша равна 1 И элемент с ключом k1 не присутствует в кэше И метод contains вызывается с ключом k1, ТО метод возвращает false
- ЕСЛИ ёмкость кэша равна 1 И элемент с ключом k1 присутствует в кэше И метод contains вызывается с ключом k1 ТО метод возвращает TRUE

- ЕСЛИ метод get вызывается с неизвестным ключом, ТО поведение не определено
- ЕСЛИ ёмкость кэша равна 1 И элемент с ключом k1 и значением v1 присутствует в кэше И метод get вызывается с ключом k1, ТО метод возвращает v1
- ЕСЛИ ёмкость кэша равна 1 И кэш заполнен И метод set вызывается с новым ключом ТО из кэша вытесняется самый старый элемент
- ЕСЛИ ёмкость кэша равна 1 И в кэше присутствует элемент с ключом k1 и значением v1 И вызывается метод set с ключом k1 и значением v2, ТО v2 перезаписывает v1
- ЕСЛИ ёмкость кэша >1 И кэш не заполнен И вызывается метод set с новым ключом, ТО элемент сохраняется в кэше
- ЕСЛИ ёмкость кэша >1 И в кэше присутствует элемент с ключом k1 и значением v1 И вызывается метод set с ключом k1 и значением v2, ТО v2 перезаписывает v1
- ЕСЛИ ёмкость кэша >1 И кэш заполнен И элемент с ключом k1 не присутсвует в кэше И вызывается метод set с ключом k1 и значением v1, ТО новый элемент вымещает самый старый элемент
- ЕСЛИ объект был инициализирован с параметром capacity равном x И вызывается метод capacity, ТО метод возвращает x
- ЕСЛИ в кэш было добавлено n элементов с разными ключами И n <= capacity И вызывается метод size, ТО метод возвращает n
- ЕСЛИ в кэш было добавлено n элементов с разными ключами И n > capacity ИИ вызывается метод size, ТО метод возвращает значение, равное capacity
- ЕСЛИ x элементов находится в кэше с ёмкостью y И вызывается метод resize с параметром z > y, ТО все элементы копируются в кэш с новым размером И значение capacity меняется на z
- ЕСЛИ x элементов находится в кэше с ёмкостью y И вызывается метод resize с параметром z < y, ТО y - z старших элементов вымещаются И z младших элементов копируются в кэш с новым размером И значение capacity меняется на z

# 3. Список литературы

*Rocca M. L.* Advanced Algorithms and Data Structures. — 1-е изд. — Manning Publications, 2021. — ISBN 1617295485,9781617295485. — URL: http://gen.lib.rus.ec/book/index.php?md5=47B7D19325B663DC8AE94A1FA17933D8.