

Спецкурс Си++. Лабораторная работа 2

Хэш-таблица

Введение:

Реализовать потокобезопасное подмножество [std::unordered_map](#)

API:

```
template <class Key, class T, class TAllocator>
class HashMap {
public:
    HashMap();

    explicit HashMap(const node_allocator_type& alloc);
    HashMap(std::initializer_list<value_type> il, const TAllocator& alloc);

    HashMap(const HashMap<Key, T, TAllocator>& other);
    HashMap(HashMap<Key, T, TAllocator>&& other);

    virtual ~HashMap();

    HashMap& operator=(const HashMap<Key, T, TAllocator>& right);
    HashMap& operator=(HashMap<Key, T, TAllocator>&& right);

    void swap(HashMap<Key, T, TAllocator>& other);

    template <class E, class P>
    iterator insert(E&& k, P&& value);

    std::optional<T> get(const Key& key);

    template <class... Args>
    std::pair<iterator, bool> emplace(Args&&... args);

    T& operator[](const Key& key);
    T& operator[](Key&& key);

    node_type extract(const_iterator position);
    node_type extract(const key_type& x);

    iterator find(const Key& key);
    const_iterator find(const Key& key) const;

    size_type size() const noexcept;
    bool empty() const noexcept;

    void clear();
```

```
// Iterators and such
iterator begin();
iterator end();
const_iterator begin() const;
const_iterator end() const;
const_iterator cbegin() const;
const_iterator cend() const;
};
```

Требования к реализации:

1. Все операции должны работать с алгоритмической сложностью, аналогичной [unordered_map](#).
2. Операции должны быть потокобезопасными
3. Код должен быть покрыт модульными тестами, нужно продемонстрировать тестовое покрытие кода класса (см. gcov/lcov).