

Лабораторная работа I Шаблоны.

Обобщённая сортировка

Используя инструментарий шаблонов C++, напишите быструю сортировку Хоара, позволяющую сортировать массив из одинаковых элементов произвольного типа, допускающего копирование.

В качестве зависимости для сравнения используйте идею "объекта компаратора": абстрактный класс, содержащий единственный метод, устанавливающий порядок элементов.

```
1 template <typename T>
2 struct Comparator {
3     virtual bool operator()(T const &, T const &) const = 0;
4 };
5
6 struct IntComparator final: Comparator<int> {
7     bool operator()(int const &lha, int const &rha) const override {
8         return lha < rha;
9     }
10 };
```

Напишите набор тестов для проверки корректности работы сортировки.

Очередь с приоритетом

Используя инструментарий шаблонов C++, напишите реализацию очереди с приоритетом на основе бинарной кучи для обработки объектов произвольного одинакового типа, допускающего копирование. Очередь должна быть представлена классом **PriorityQueue** со следующими методами:

- конструктор по умолчанию, создающий пустую очередь с приоритетом;
- метод **push** для добавления нового значения в очередь;
- метод **peek** для чтения верхнего элемента очереди, но не удаления элемента;
- метод **poll** для удаления верхнего элемента очереди;

- метод **is_empty**, который возвращает **true**, если очередь пуста, иначе - **false**.

Для определение приоритета используйте идею компаратора из предыдущего пункта. Компаратор должен внедряться в очередь в конструкторе (сохраняется ссылка или указатель). Так как очередь будет выделять память под новые элементы, дополните методы методом **void free()**, который освобождает всю выделенную ранее память.

Напишите набор тестов для проверки корректности работы очереди.