Министр науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет информационных технологий и программирования

Лабораторная работа №6

Алгоритмы

Выполнил студент группы № M3119
Самигуллин Руслан Рустамович
Подпись:

Проверил:
Повышев Владислав Вячеславович

Санкт-Петербург
2024

Требуется реализовать следующие обобщенные алгоритмы.
Каждый алгоритм должен быть выполнен в виде шаблонной функции, позволяющей взаимодействовать со стандартными контейнерами STL с помощью итераторов. Предикаты, условия, операторы сравнения должны быть параметризованы.
При сдаче работы требуется продемонстрировать работу алгоритмов как на стандартных, так и на пользовательских типах данных, например CPoint, CRational, далее работает ваша индивидуальная фантазия.
1. Для вектора целых чисел выполняются проверки:
- Проверка, все ли элементы положительные с помощью all_of.
- Проверка, отсортированы ли элементы по возрастанию с помощью is_sorted.
- Вывод последнего элемента вектора.

2. Для вектора структур CPoint:
- Проверка, все ли точки находятся в положительном квадранте с помощью all_of.
- Проверка, отсортированы ли точки по координате x с помощью is_sorted.
- Вывод последней точки.

3. Для вектора структур CTriangle:
- Проверка, все ли вершины всех треугольников находятся в положительном квадранте с помощью all_of.
- Проверка, отсортированы ли вершины треугольников по координате x с помощью is_sorted.
- Вывод координат последнего треугольника.

4. Для вектора структур CRational:
- Проверка, все ли рациональные числа положительные с помощью all_of.
- Проверка, отсортированы ли рациональные числа по возрастанию с помощью is_sorted.
- Вывод последнего рационального числа.

Код также содержит класс исключения ComparisonError, который используется для обработки ошибок сравнения элементов.

Результат работы:
Введите количество элементов (0-20): 5
Введите 5 элементов: 1 2 3 4 5
Все элементы положительные.
Элементы сортированы по возрастанию
Последний элемент: 5
Введите количество координат (0-3): 2
Введите 1-ые координаты (x, y): 0 0
Введите 2-ые координаты (x, y): 1 1
Не все точки находятся в положительном квадранте.
Точки отсортированы по координате x.
Последняя точка: (1, 1)
Введите количество треугольников (0-3): 2
Введите координаты для 1-го треугольника (x1, y1), (x2, y2), (x3, y3): 0 0 1 1 1 0
Введите координаты для 2-го треугольника (x1, y1), (x2, y2), (x3, y3): 1 1 2 2 2 1
Не все вершины всех треугольников находятся в положительном квадранте.
Вершины треугольника отсортированы по координате x.
Вершины треугольника отсортированы по координате x.
Координаты вершин последнего треугольника: (1, 1), (2, 2), (2, 1)
Введите количество рациональных чисел (0-3): 2
Введите числитель и знаменатель для 1-го числа (числитель, знаменатель): 1 2
Введите числитель и знаменатель для 2-го числа (числитель, знаменатель): 1 3
Все рациональные числа положительные.

Рациональные числа не отсортированы по возрастанию.
Последнее рациональное число: 1/3
Вопросы:
1. Что такое предикат? Как реализованы предикаты?
Ответ: это функция или функциональный объект, который принимает один или несколько аргументов и возвращает булевское значение (true или false), в зависимости от выполнения какого-то условия. Предикаты реализованы в виде функций и лямбда-выражений, которые принимают аргументы и возвращают булевское значение в соответствии с определенными условиями.

Пример:

```
// Предикат для проверки, все ли точки находятся в положительном квадранте
bool is_positive_point(const CPoint& p) {
   return p.x > 0 && p.y > 0;
}
```

2. Что такое функтор? Как реализованы функторы?
Ответ: это объект, который ведет себя как функция благодаря оператору вызова функции.
В моем коде отсутствуют функторы.

3. Что такое лямбда-выражение? Как они компилируются и как задаются?
Ответ: это анонимная функция, представленная в виде выражения. Они компилируются в объекты функций, которые могут быть вызваны в любом месте программы, где они определены. В коде лямбда-выражения используются для определения предикатов в вызовах функций all_of и is_sorted.

4. Как реализован каждый обобщённый алгоритм? Что принимает и что возвращает?

1. all_of: принимает два итератора и предикат, который определяет условие для каждого элемента. Возвращает true, если все элементы в диапазоне удовлетворяют условию, заданному предикатом.

2. sorted_of: принимает два итератора и компаратор, который определяет порядок сортировки. Возвращает true, если все элементы отсортированы с порядком, заданным компаратором.

3. find_backward: принимает два итератора, значение для поиска и предикат, который определяет условие для поиска. Возвращает итератор, указывающий на последний элемент в диапазоне.