# Лабораторная работа №1

Тихомирова Наталия Б19-511

# По курсу «Языки программирования и методы программирования» (информатика, 3 семестр)

**Техническое задание**

## Введение

На основе реализации класса Sequence<T>, полученной в рамках курса за 2-й семестр, была написана программа на С++, способная производить сортировку исходных данных с помощью одного из 3 алгоритмов сортировки, а также сравнивать скорость выполнения алгоритмов.

## Алгоритмы сортировки

В рамках лабораторной работы были реализованы следующие алгоритмы:

1. “Bubble Sort” (BubbleSort()). Сортировка пузырьком – простейший способ отсортировать исходные данные. Алгоритм проходит несколько раз по массиву, каждый раз меняя соседние элементы, если они стоят в неправильном порядке.
2. “Shell Sort” (ShellSort()). Сортировка Шелла - идея метода заключается в сравнение разделенных на группы элементов последовательности, находящихся друг от друга на некотором расстоянии. Изначально это расстояние равно d или N/2, где N — общее число элементов. На первом шаге каждая группа включает в себя два элемента расположенных друг от друга на расстоянии N/2; они сравниваются между собой, и, в случае необходимости, меняются местами. На последующих шагах также происходят проверка и обмен, но расстояние d сокращается на d/2, и количество групп, соответственно, уменьшается. Постепенно расстояние между элементами уменьшается, и на d=1 проход по массиву происходит в последний раз.
3. “Heap Sort” (HeapSort()). Метод пирамидальной сортировки является улучшением традиционных сортировок с помощью дерева. Общая идея пирамидальной сортировки заключается в том, что сначала строится пирамида из элементов исходного массива, а затем осуществляется сортировка элементов. Выполнение алгоритма разбивается на два этапа.

## Проверка работы алгоритмов

Все алгоритмы покрыты тестами, проверяющими их работу с последовательностями нулевой, единичной и произвольной длины, последовательностями, содержащими равные или различные значения.

Также присутствует консольный интерфейс, позволяющий проверить корректность работы алгоритмов на последовательностях, любой длины и содержания. Присутствует возможность создания произвольной последовательности фиксированной длины и отсортированной последовательности фиксированной длины.

Консольное приложение также предоставляет возможность узнать время работы алгоритмов на тех или иных входных данных. В том числе возможно сравнение времени работы алгоритмов на одинаковых входных данных.