

Спецификация

Постановка задачи:

Разработка приложения, визуализирующего алгоритм сортировки слабой кучей.

Возможности приложения:

Приложение должно выполнять следующие функции:

1. Задание входных данных с помощью текстового поля.
2. Задание входных данных с помощью элементов графического интерфейса путём нажатия на кнопки, которые будут отражать массив элементов, и выбора соответствующей функции в контекстном меню, добавление элемента через отдельную кнопку с символом “+”.
3. Удаление и обмен местами элементов, также с помощью контекстного меню кнопок.
4. Применение алгоритма сортировки к введённым данным путём нажатия соответствующей кнопки.
5. Визуализация элементов массива набором кнопок, пошаговая визуализация самого алгоритма. На каждом шаге будет отражаться текущее состояние массива и бинарного дерева. Отсортированная часть массива будет выделена специальным цветом. На первом этапе, построении кучи визуализируется каждое перемещение элементов, при этом элементы подсвечиваются специальным цветом. На втором этапе, непосредственно сортировки визуализируются операции удаления и перемещения элементов, которые также окрашиваются в специальный цвет. На экране рядом с деревом будет текст при построении кучи "Этап 1. Построение кучи" а при непосредственно сортировке "Этап 2. Сортировка".
6. Считывание входных данных из файла и запись результатов в файл.
7. Взаимодействие с пользователем посредством текстовых сообщений, поясняющих управление приложением и шаги работы алгоритма, указывающие, какие элементы удаляются, перемещаются и сравниваются на каждой операции.

Пример работы визуализации алгоритма.

Пусть на текущем шаге минимальный элемент (34) находится в корне кучи. Следующим шагом должна быть смена элемента в корне (34) и последнего элемента в куче (42).

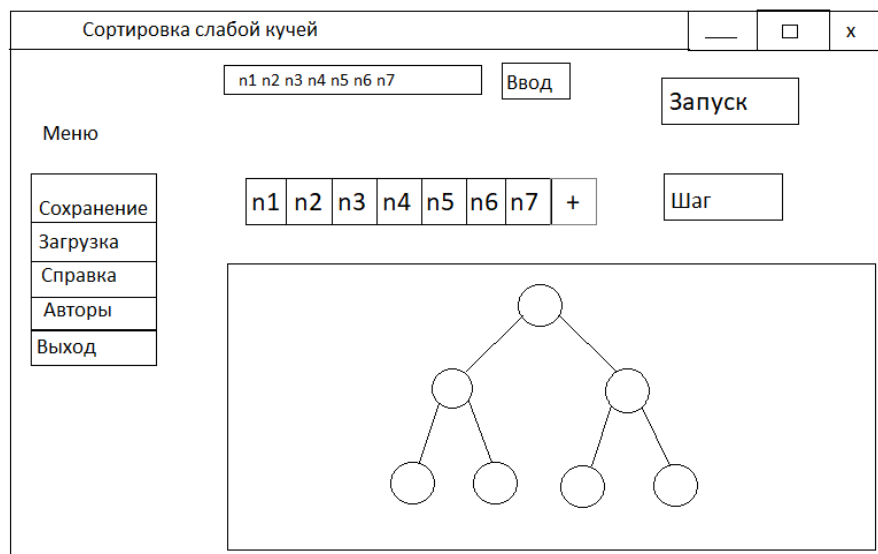
Красим эти два элемента (например, в красный) в куче и массиве, после чего производим следующий шаг (обмен):

В куче последний элемент 42 становится корнем, а элемент 34 исчезает из кучи; в массиве элементы 34 и 42 меняются местами. Элемент 34 становится частью отсортированного куска массива и красится в цвет, соответствующий отсортированному куску (например, серый).

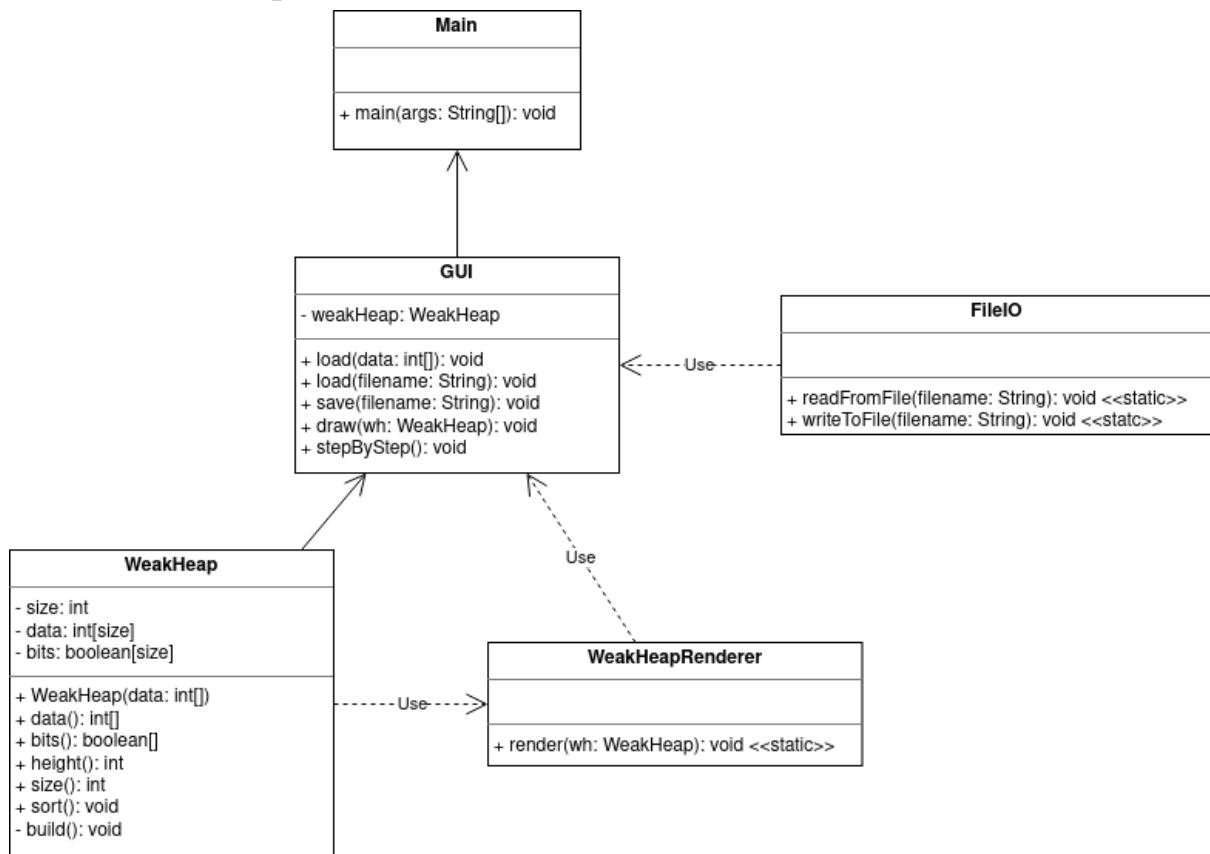
В качестве текстового пояснения данного шага пользователю выводится сообщение по типу:

"В корне находится минимальный элемент 34. Меняем его местами с последним элементом неотсортированной части массива 42. Удаляем элемент 34 из корня. Теперь элемент 34 принадлежит отсортированной части массива."

Эскиз интерфейса.



UML диаграмма классов.



Распределение ролей в бригаде.

1. Птичкин Сергей - реализация графического интерфейса.
2. Ноздрин Василий - реализация алгоритма сортировки, считывание данных, работа с файлами.
3. Прибылов Никита - тестирование, сборка проекта, визуализация алгоритма.

План разработки:

- 1) 5 июля - согласование спецификации и плана.
- 2) 7 июля - прототип (интерфейс)
- 3) 9 июля - 1 версия (реализация алгоритма, некоторые функции графического интерфейса, визуализация результата работы алгоритма)
- 4) 11 июля - 2 версия (загрузка, сохранение данных в файловой системе, пошаговая визуализация работы алгоритма, добавление текстового описания шагов алгоритма, защита приложения от некорректных действий пользователя)

- 5) 13 июля - финальная версия (обработка программой исключительных ситуаций, доработка уже реализованных возможностей приложения)