

Требования к программе

Программа должна выполнять сортировку элементов типа Integer (до 1000 элементов от 0 до `maxInt`) и String (до 1000 элементов по 128 латинских букв и цифр) поразрядным методом (Radix Sort) и осуществлять визуализацию работы данного алгоритма. Она должна обладать графическим интерфейсом пользователя (далее GUI) с возможностью ручного ввода элементов для сортировки и случайной генерации заданного количества элементов для автоматического ввода.

Для визуализации перемещения элементов по колонкам, текущий сортируемый элемент визуально выносится за пределы массива и имеет более крупный размер шрифта для отображения. На следующем шаге он переносится в нужную колонку, которая в свою очередь выделяется цветом. При этом, на его место встает следующий элемент рабочего массива.

Программа предполагает три режима работы:

- Автоматический: программа сразу сортирует введенные числа и выводит конечный результат;
- Пошаговый: в этом режиме сортировка разделяется на шаги (например, перенос одного числа из рабочего массива в специальный разрядный массив), которые выполняются строго по очереди и при нажатии соответствующей клавиши. Возможно выполнение шагов как вперед, так и назад. При этом, актуальное промежуточное состояние рабочих массивов отображается на экране;
- Плавный: шаги работы алгоритма выполняются автоматически через заданные промежутки времени (например, одну секунду).

Способы ввода данных:

- Ручной: ввод данных в поле ввода данных осуществляется с клавиатуры.
- Ввод из файла: данные поступают с файла в формате *.txt, ввод осуществляется по нажатию на соответствующую кнопку, при этом данные будут дублированы в поле ручного ввода, для возможности их редактирования.
- Генерация случайных данных: при нажатии на кнопку генерации производится заполнение поля ввода данных случайными числами по параметрам для генерации: количество чисел, диапазон данных.

Демонстрация текущего шага:

- Поле комментария: отдельное поле для демонстрации комментариев по текущим событиям, заполняется автоматически после выполнения «шага».
- Подсветка действия: выделение цветом разряда и колонки, с которыми производится работа на текущем действии.

Анимация движений:

- Анимация перемещения: плавное перемещение элемента на нужную позицию.

Анализ задачи

Данная программа будет состоять из двух базовых классов:

- класс, отвечающий за GUI и обеспечивающий работу основной логики программы;
- класс, реализующий разработку алгоритма поразрядной сортировки.

В самом начале оба класса будут разрабатываться отдельно, затем они будут связаны между собой при помощи предварительно описанных интерфейсов (под интерфейсом в данном случае подразумевается набор методов класса, разрешенных для использования извне).

Отвечающий за GUI класс будет содержать, в частности, методы, служащие для проецирования внутреннего состояния рабочих массивов на графические элементы главного окна программы.

Таким образом, логика работы каждого класса останется только в соответствующем классе и никак не будет зависеть от внешней среды.

Если понадобится, будут реализованы вспомогательные классы, упрощающие разработку некоторых частей программы.

Интерфейс

Макет представлен на рис. 1

Введите числа:

Чтобы заполнить случайными числами, введите необходимые данные:
 Генерировать числа от 0 до: Количество чисел:

32 543 754 349 2332 95 3 764

Текущий разряд сортировки: 1

540 110					15 645 115		7		99
0	1	2	3	4	5	6	7	8	9

Комментарии к выполнению:

Рис. 1

Уточнения

В последнюю версию программы были внесены уточнения:

1. Необходимость добавить генерацию по интервалу, задаваемого пользователем.
2. Разделение главных итераций (разрядов) в поле комментариев для большей наглядности.
3. Сделать окно пошагового вывода адаптивным от размера входных данных.

План разработки

Разработка программы будет вестись с использованием языка программирования Java и его мощных возможностей по созданию GUI. Для упрощения написания программы используется интегрированная среда разработки (далее IDE) **IntelliJ IDEA**.

Для обеспечения доступа к исходному коду всех участников проекта и более гибкой работы с ним используется система контроля версий **Git**.

Репозиторий с исходным кодом и данной спецификацией расположен на крупнейшем веб-сервисе для хостинга IT-проектов **GitHub** по адресу:

<https://github.com/EgorYackovleff/RadixSort.git>.

План разработки программы разбит на следующие шаги:

1. Создание прототипа: к этому шагу будет создан класс, отвечающий за GUI, и реализован некоторый интерфейс к нему. Результатом работы на данном шаге будет являться макет GUI с неработающими элементами, служащий только для демонстрации.
2. Первая версия: на этом шаге будет реализован класс, отвечающий за работу самого сортировочного алгоритма. Затем он будет соединен с классом GUI, используя интерфейсы обоих классов. В результате получится рабочая программа, которая, тем не менее, может содержать некоторые недоделанные компоненты или баги, также будет реализован ввод из файла, добавление окна с комментариями на текущий шаг.
3. Вторая версия: реализация пошаговой работы программы.
4. Третья (конечная) версия: добавление возможности генерация входных данных, анимация движения, подсветка текущего действия. Во время последней итерации будет вестись работа над исправлением ошибок, неочевидными моментами в работе программы и небольшими улучшениями.

Распределение ответственности

Яковлев Егор: реализация алгоритма сортировки внутри класса и методов взаимодействия с ним; написание отчета.

Иванкова Валерия: написание спецификации программы; проектирование системы классов и их взаимодействия; связывание готовых классов программы; написание отчета.

Клименко Кирилл: реализация GUI и методов взаимодействия с ним; написание отчета.