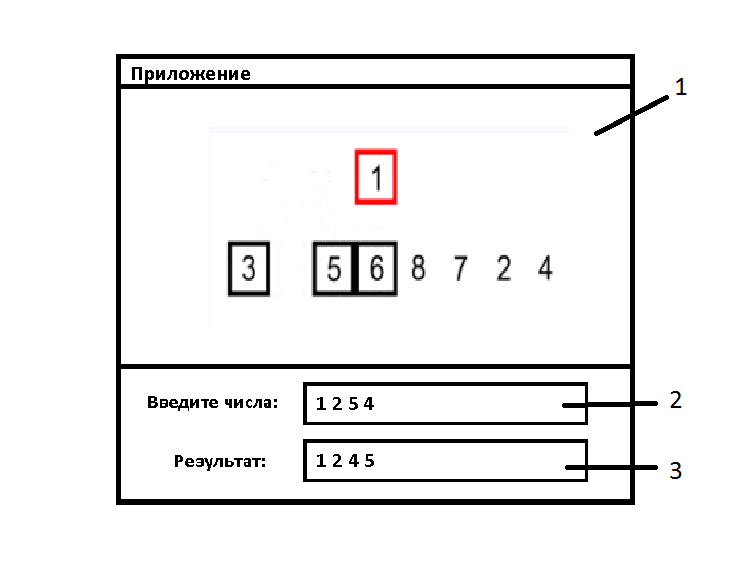
**Пользовательский интерфейс**

Рисунок 1 - Эскиз главного окна программы



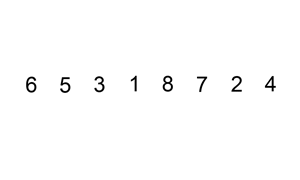
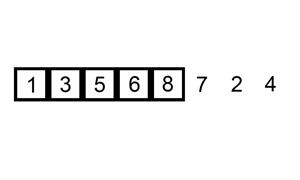
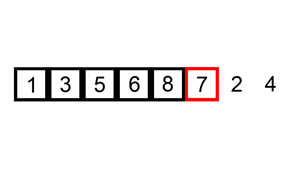
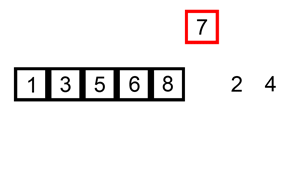
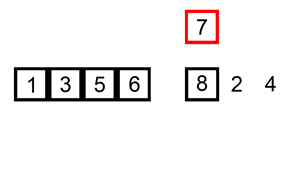
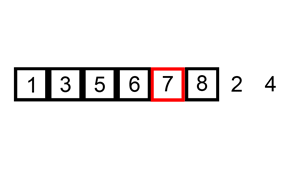
На рисунке 1 представлен эскиз пользовательского интерфейса.

Блок 1 окна программы предназначен для визуализации алгоритма сортировки вставками.

В текстовое поле 2 пользователь вводит исходный массив чисел, для которого будет продемонстрирована сортировка. Формат входных данных – список целых чисел, меньших 100 по модулю, разделенных пробелами. Максимум 10 чисел.

В текстовом поле 3 будет выведен отсортированный массив чисел или сообщение об ошибке, если пользователь ввел некорректные данные. Пользователь не может изменять данные в поле 3.

**Анимация**

Рисунок 2 – Этапы анимации визуализации алгоритма сортировки вставками

В начале сортировки числа исходного массива выписаны в ряд. В процессе сортировки обрабатывается каждое число в массиве, т.е происходит n подобных этапов.

В начале каждого этапа все предыдущие n-1 числа отсортированы и заключены в черный квадрат. Обрабатываемое число заключается в красный квадрат и плавно поднимается вверх на 1.5 квадрата. Пустая ячейка постепенно двигается влево, меняясь местами с черными квадратами, пока число слева от пустой ячейки не станет меньшим или равным активному. Тогда красный квадрат занимает пустое место, окрашивается в черный и этап завершается.

Сортировка заканчивается, когда все числа обработаны.

**Архитектура**

Программа будет написана на языке Java, с использованием библиотеки Swing.

**Описание классов программы.**

**NumberBlock**

Класс для представления элемента анимации – числа, заключенного в квадрат.

Псевдокод публичного интерфейса:

*NumberBlock(int number);*  // конструктор

*void setNumber(int number);* // устанавливает содержимое блока

*int getNumber();* // возвращает число, содержащееся в блоке

*void setColor(Color color);* // устанавливает цвет границы

*Color getColor();* // возвращает цвет границы

*void setEdgeVisibility(bool isVisible);* // устанавливает видимость границы

*void setPosition(Vector newPosition);* // перемещает блок в позицию newPosition

*Vector getPosition();* // возвращает координаты левого верхнего угла блока.

**InsertionSortAnimation**

Класс для управление анимацией сортировки.

Псевдокод публичного интерфейса:

*InsertionSortAnimation(int[] array, Field field);*

*void setField(Field field);* // Установить поле, в котором воспроизводится анимация

*void setArray(int[] array*); // Устанавливает сортируемый массив

*int[] getInitialArray();* // Возвращает исходный массив

*int[] getSortedArray();* // Возвращает отсортированный массив

*void setAnimationActivity(bool isActive);* // Активация/деактивация анимации

*tick(double dt);* // изменение состояния анимации на время *dt*

+классы из библиотеки Swing

**Описание работы программы.**

При нажатии Enter в поле ввода массива будет вызвана функция-обработчик, которая попробует распарсить пользовательский ввод и перевести его в массив *int[]*. При неудачной обработке будет выведено сообщение об ошибке. При удачной обработке массив будет передан объекту *InsertionSortAnimation* и будет вызван метод *setAnimationActivity(True)*.

На прерывание по таймеру будет установлена функция обработчик, которая будет вызывать метод *tick()* у объекта *InsertionSortAnimation* меняющий состояние анимации на состояние в момент времени *t+dt*.

**План разработки**

Прототип – Завершенный пользовательский интерфейс, программа никак не реагирует на ввод. Окно анимации пусто.

Версия 1 – Программа корректно обрабатывает пользовательский ввод, выводится отсортированный массив. Выводятся числа исходного массива в поле анимации, но сама анимация не проигрывается.

Версия 2 – Программа полностью готова и соответствует требованиям спецификации.

**Распределение обязанностей**

Гаськов Максим – Frontend Software Engineer

Создание пользовательского интерфейса на Swing, написание класса *NumberBlock,* написание обработчиков прерываний.

Кобылянский Алексей - Project manager

Разработка архитектуры, интерфейсов взаимодействия компонентов программы. Написание спецификации и отчетов, работа с заказчиками.

Розенкинд Евгений - Backend Software Engineer

Написание класса управления анимацией сортировки *InsertionSortAnimation.*