**Спецификация программного продукта**

Название: SwiftSorting

Описание: SwiftSorting - это программный продукт на языке Swift, который содержит 4 алгоритма сортировки: сортировка пузырьком, сортировка выбором, сортировка вставками и быстрая сортировка. Каждый алгоритм имеет свои особенности и может использоваться в зависимости от требований и данных, которые нужно отсортировать.

Алгоритм 1: Сортировка пузырьком

Описание: Сортировка пузырьком - это простой алгоритм сортировки, который проходит по списку несколько раз, сравнивая пары элементов и меняя их местами, если они не отсортированы в нужном порядке.

Сигнатура функции:

func bubbleSort<T: Comparable>(\_ arr: inout [T])

Аргументы функции:

* arr: массив элементов, который нужно отсортировать.

Возвращаемое значение:

* Отсортированный массив элементов

Алгоритм 2: Сортировка выбором

Описание: Сортировка выбором - это алгоритм сортировки, который ищет наименьший элемент в списке и перемещает его на первое место, затем ищет следующий наименьший элемент и перемещает его на второе место, и так далее, пока весь список не будет отсортирован.

Сигнатура функции:

func selectionSort<T: Comparable>(\_ arr: inout [T])

Аргументы функции:

* arr: массив элементов, который нужно отсортировать.

Возвращаемое значение:

* Отсортированный массив элементов

Алгоритм 3: Сортировка вставками

Описание: Сортировка вставками - это алгоритм сортировки, который проходит по списку и вставляет каждый элемент в отсортированную часть списка в нужном порядке.

Сигнатура функции:

func insertionSort<T: Comparable>(\_ arr: inout [T])

Аргументы функции:

* arr: массив элементов, который нужно отсортировать.

Возвращаемое значение:

* Отсортированный массив элементов

Алгоритм 4: Быстрая сортировка

Описание: Быстрая сортировка - это алгоритм сортировки, который разделяет список на две части, затем рекурсивно сортирует каждую часть, используя один из элементов как опорный, затем объединяет отсортированные части.

Сигнатура функции:

func quickSort<T: Comparable>(\_ arr: inout [T], low: Int, high: Int)

Аргументы функции:

* arr: массив элементов, который нужно отсортировать.
* low: индекс начала массива.
* high: индекс конца массива.

Возвращаемое значение:

* Отсортированный массив элементов

Пример использования:

var arr = [5, 3, 8, 4, 2]

bubbleSort(&arr)

print(arr) // [2, 3, 4, 5, 8]

arr = [5, 3, 8, 4, 2]

selectionSort(&arr)

print(arr) // [2, 3, 4, 5, 8]

arr = [5, 3, 8, 4, 2]

insertionSort(&arr)

print(arr) // [2, 3, 4, 5, 8]

arr = [5, 3, 8, 4, 2]

quickSort(&arr, low: 0, high: arr.count - 1)

print(arr) // [2, 3, 4, 5, 8]