# **UIStackView**

**UIStackView** - это контейнерный вид, который упрощает управление размещением и распределением элементов интерфейса. Он автоматически управляет расположением и размерами своих дочерних представлений в виде горизонтальной или вертикальной стопки.

**Основные свойства UIStackView:**

**- axis**: Определяет ось, вдоль которой располагаются представления (**.vertical** - вертикально, **.horizontal** - горизонтально).

**- distribution**: Управляет распределением представлений внутри **UIStackView** (**.fill**, **.fillEqually**, **.fillProportionally**, **.equalSpacing**, **.equalCentering**).

**- alignment**: Определяет выравнивание представлений вдоль поперечной оси (**.fill**, **.leading**, **.center**, **.trailing**, **.top**, **.bottom**).

**- spacing**: Определяет расстояние между представлениями в стеке.

**- isLayoutMarginsRelativeArrangement**: Булево значение, указывающее, следует ли использовать отступы макета для определения расположения представлений.

**Пример реализации программно:**

// Создание UIStackView

let stackView = UIStackView()

// Настройка основных параметров

stackView.axis = .vertical // Устанавливаем вертикальную ориентацию для UIStackView

stackView.distribution = .fill // Распределение представлений внутри UIStackView будет равномерным

stackView.alignment = .fill // Выравнивание представлений будет по ширине/высоте

stackView.spacing = 8 // Устанавливаем расстояние между представлениями в 8 точек

stackView.isLayoutMarginsRelativeArrangement = true // Отступы макета будут использоваться для определения расположения представлений

// Добавление представлений в UIStackView

let view1 = UIView()

let view2 = UIView()

// Настройка представлений...

stackView.addArrangedSubview(view1) // Добавляем первое представление в UIStackView

stackView.addArrangedSubview(view2) // Добавляем второе представление в UIStackView

// Добавление UIStackView в иерархию представлений

view.addSubview(stackView) // Добавляем UIStackView на главное представление

// Настройка констрейнтов (автодизайн)

stackView.translatesAutoresizingMaskIntoConstraints = false // Отключаем автоматически сгенерированные констрейнты

stackView.topAnchor.constraint(equalTo: view.topAnchor, constant: 20).isActive = true // Устанавливаем верхний констрейнт с отступом 20 точек

stackView.leadingAnchor.constraint(equalTo: view.leadingAnchor, constant: 20).isActive = true // Устанавливаем левый констрейнт с отступом 20 точек

stackView.trailingAnchor.constraint(equalTo: view.trailingAnchor, constant: -20).isActive = true // Устанавливаем правый констрейнт с отступом 20 точек

// Можно добавить констрейнты по высоте/ширине или другим правилам размещения

// Обновление интерфейса

view.layoutIfNeeded() // Принудительное обновление интерфейса для применения изменений