# **UIPickerView**

* **UIPickerView** - это элемент интерфейса, который позволяет пользователю выбирать одну или несколько значений из списка с помощью крутящегося селектора.
* **UIPickerView** используется для предоставления пользователю выбора из ограниченного набора значений или опций. Некоторые примеры использования:

**- Выбор даты и времени**: Позволяет пользователю выбирать дату и/или время.

- **Выбор из списка опций**: Используется для выбора элемента из списка опций.

- **Настройка параметров или категорий**: Пользователь выбирает параметры или категории из представленного списка.

* **Пример кода с характеристиками свойств:**

// Создаем экземпляр UIPickerView

let pickerView = UIPickerView(frame: CGRect(x: 50, y: 100, width: 200, height: 150))

// Настраиваем делегата и источник данных

pickerView.delegate = self

pickerView.dataSource = self

// Добавляем UIPickerView на экран

self.view.addSubview(pickerView)

// Количество компонентов (столбцов) в UIPickerView

func numberOfComponents(in pickerView: UIPickerView) -> Int {

return 1 // Одна компонента для списка опций

}

// Количество строк в UIPickerView

func pickerView(\_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {

return data.count // Количество строк равно количеству опций в списке

}

// Настраиваем отображение опций в UIPickerView

func pickerView(\_ pickerView: UIPickerView, titleForRow row: Int, forComponent component: Int) -> String? {

return data[row] // Отображаем каждую опцию из массива

}

// Обработка выбора опции в UIPickerView

func pickerView(\_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component: Int) {

let selectedOption = data[row]

print("Выбранная опция: \(selectedOption)")

}

* // ассинхронная прорисовка линий textField'a на главном потоке
* /\* DispatchQueue - это класс в Grand Central Dispatch (технология для управления параллельным выполнением кода в многозадачной среде), который управляет выполнением задач в многозадачной среде.
* .main указывает на главную очередь. Главная очередь является основной очередью диспетчера в GCD и обычно используется для выполнения кода, который взаимодействует с пользовательским интерфейсом.
* .async указывает на то, что задача должна быть выполнена асинхронно, т.е., код, находящийся в блоке, не будет ждать завершения этой задачи. \*/
* DispatchQueue.main.async { [weak self] in // [weak self] создает слабую ссылку на текущий объект - LoginUserProfileView
* self?.settingLineLogin()
* self?.settingLinePassword()

}