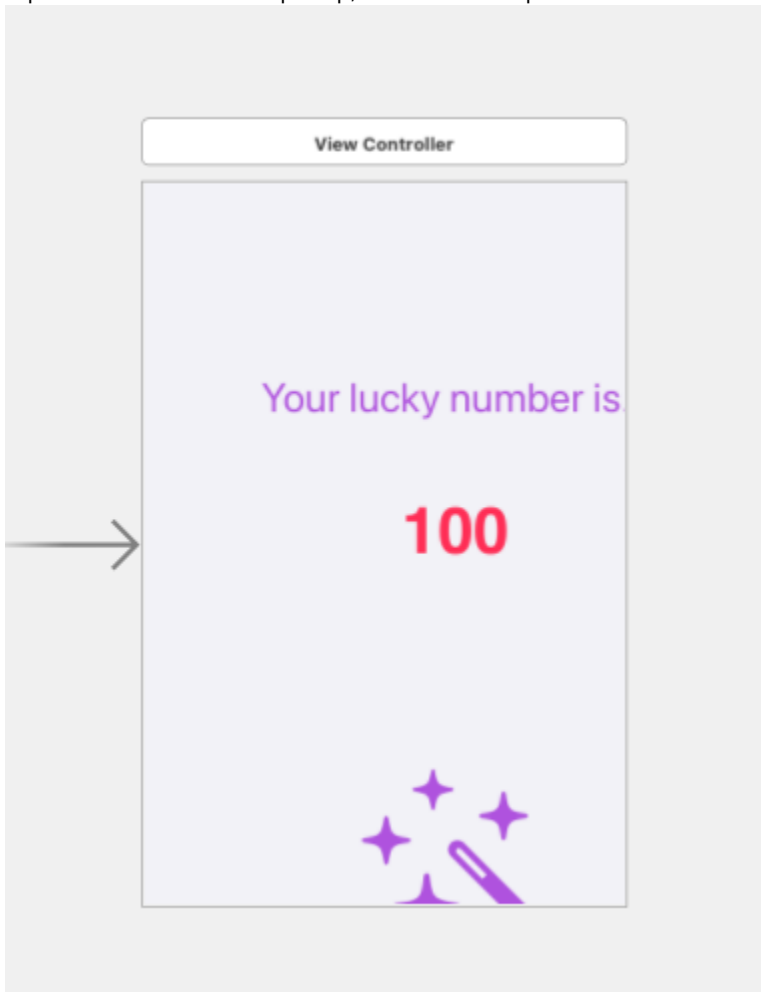


Часть 3. Вёрстка с помощью констрейнтов, контейнеров и стеков.

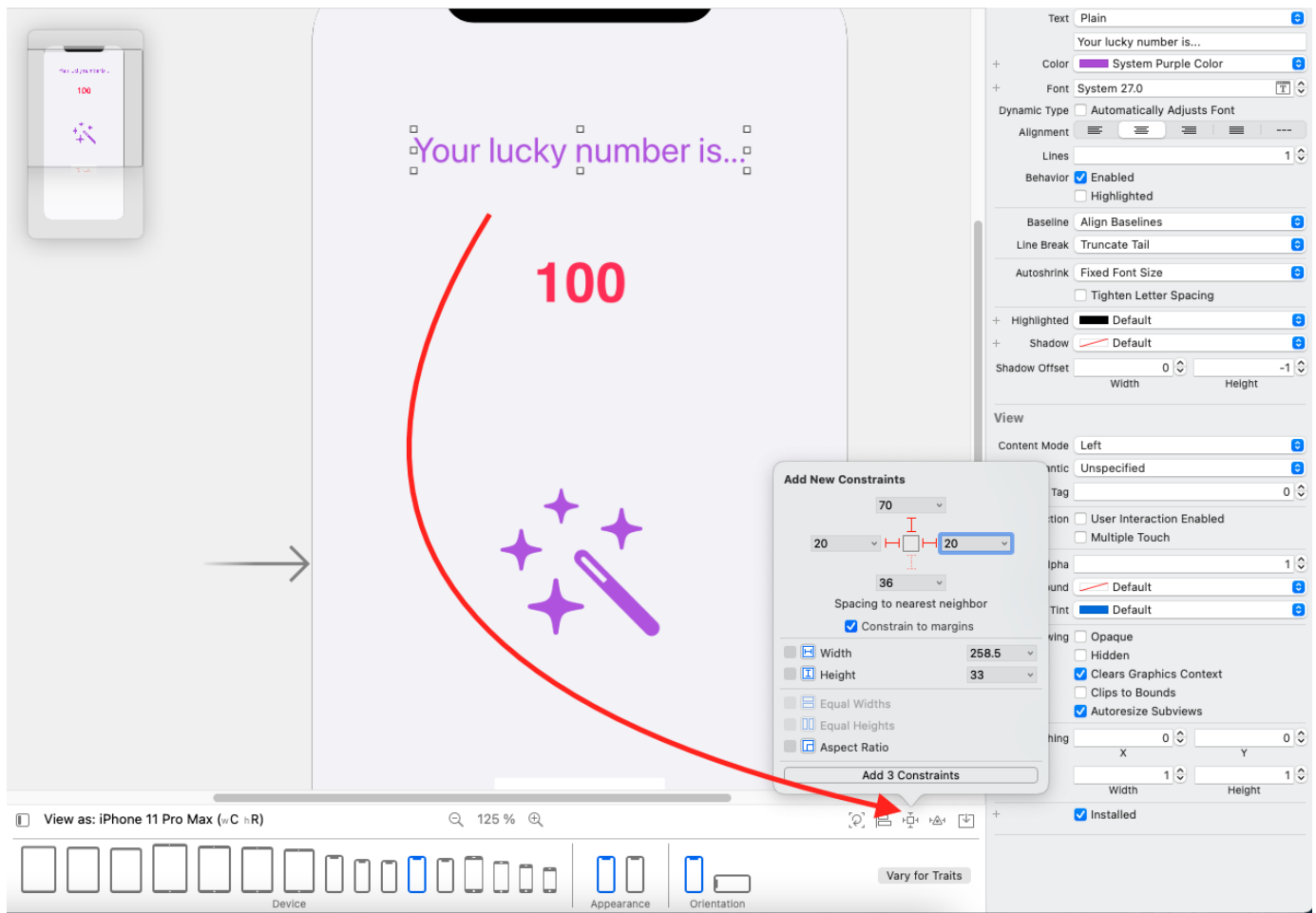
Наш генератор счастливых чисел отлично смотрится на экране 11 айфона, но что будет, если мы выберем модель с экраном поменьше? Например, на 4s это смотрелось бы вот так



Все элементы “уехали” вниз и вправо. Догадаетесь, почему?) Всё дело в том, что расположение каждого элемента рассчитывается от верхней левой точки экрана

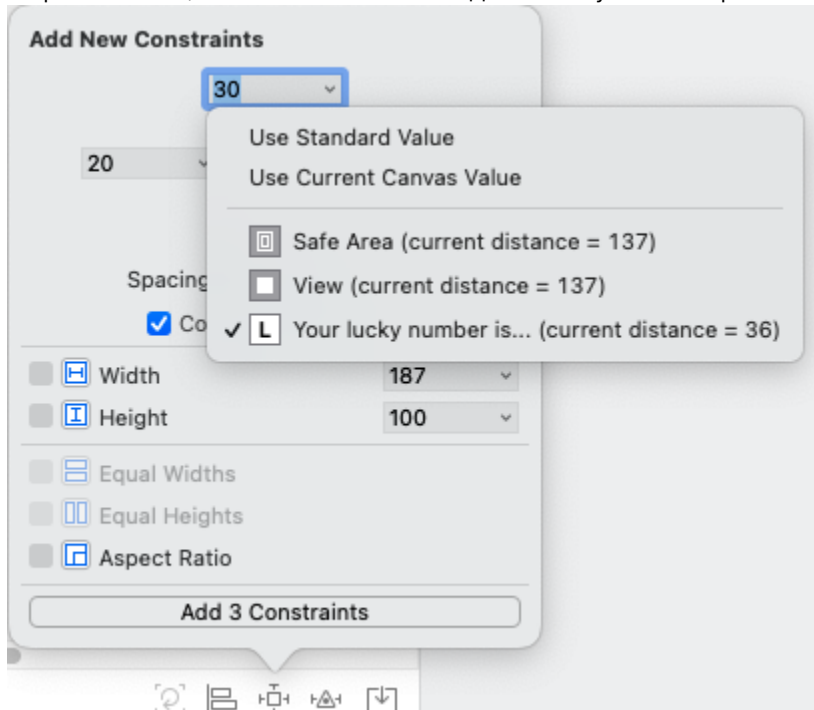
Для того, чтобы наш интерфейс целиком отображался на любых экранах, существует **Auto Layout**. Это механизм, который сам вычислит X, Y, Width и Height для элемента, а мы всего лишь зададим ему несколько ограничительных условий (они называются **constraints** или просто констрейнты).

Выберем наш верхний Label и нажмем на кнопку *Add New Constraints*. А теперь выставим 3 ограничения - 70 точек сверху (не забудьте убедиться, что красная полоска возле поля ввода стала подсвеченной), и по 20 точек слева и справа



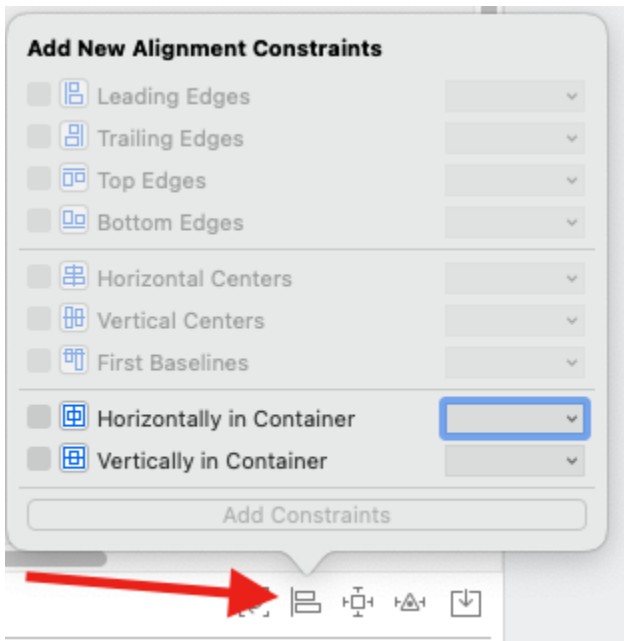
Теперь можем переключить модель айфона в сториборде и убедиться, что на любом экране этот Label отобразится по центру, с одинаковыми отступами от верхней и боковых границ

Повторим то же самое для нашего второго Label (с числом). Нажмем *Add New Constraints*, установим 20 по бокам и 30 сверху. Но теперь выберем другой объект для отступа. По нажатию на стрелку рядом с верхним констрейнтом откроется меню, и мы сможем сказать: "Делай отступ не от верхнего края экрана, а от другого Label"



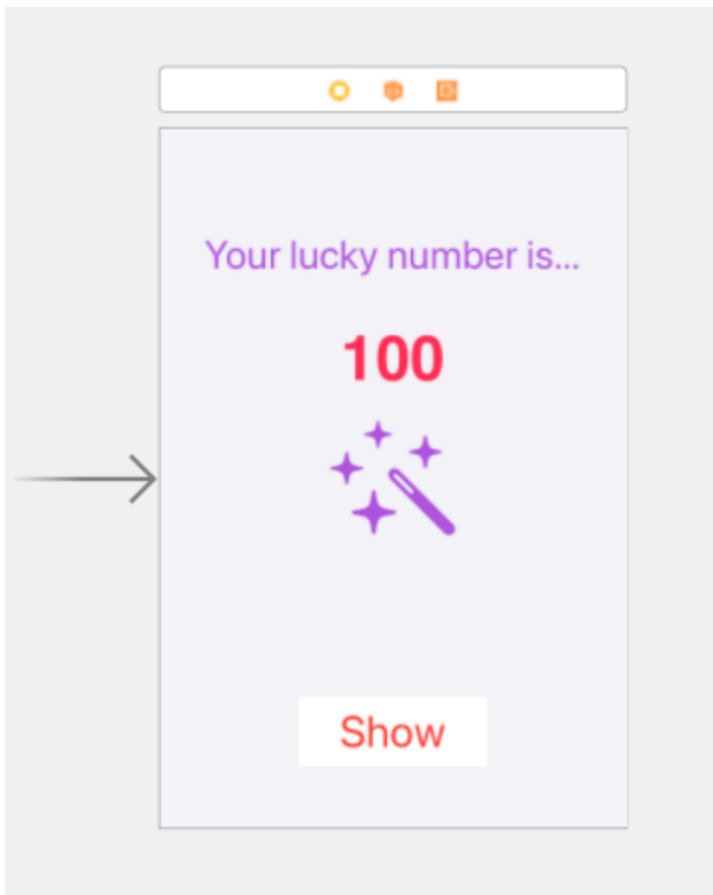
По сути, мы ставим математическую задачку для Xcode с условием: “Найди для этого элемента такие X, Y, Width и Height, чтобы он располагался на 20 точек левее одного края, на 20 точек правее другого, и на 30 точек ниже соседнего элемента”. Самостоятельно решая это уравнение, Xcode располагает наш элемент на каждом экране.

Для нашего UIImageView мы попробуем кое-что новое. Вместо констрейнтов выберем выравнивание и нажмем кнопку *Align*.

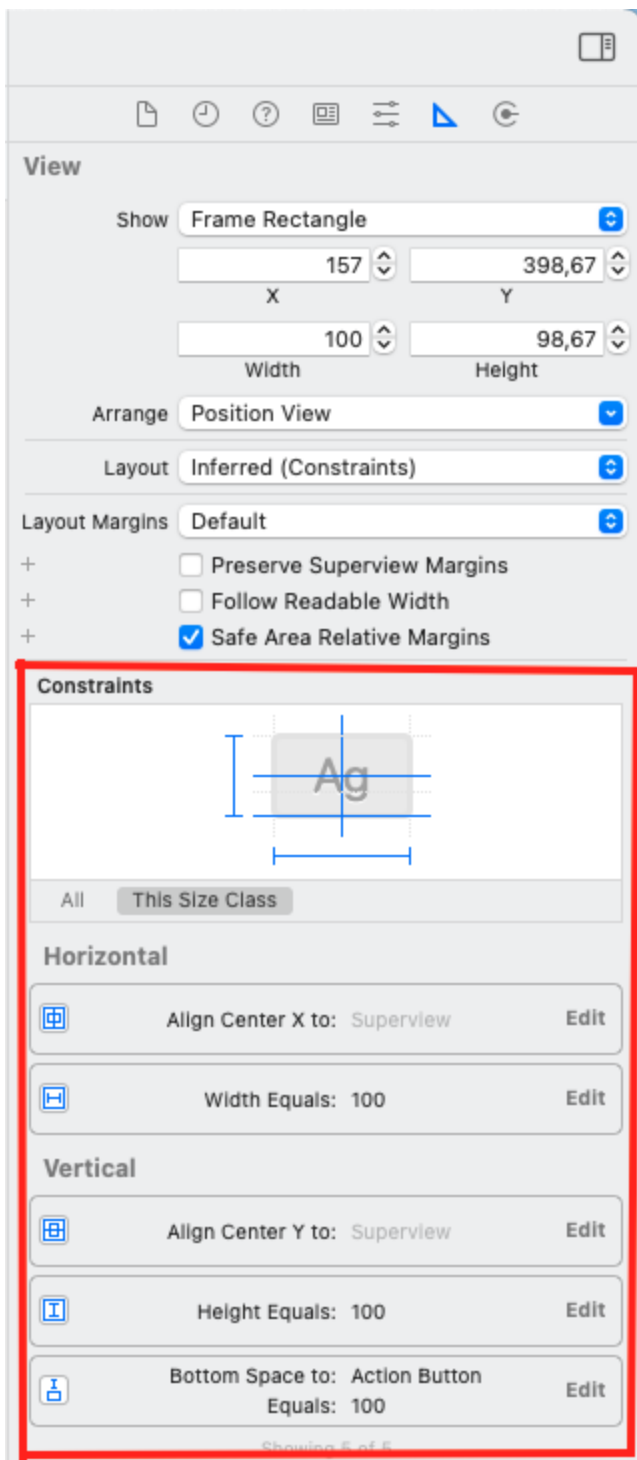


Если поставить галочки в пунктах *Horizontally in Container* и *Vertically in Container*, картинка получит выравнивание по горизонтали и вертикали одновременно, то есть окажется прямо по центру...но сильно уменьшится в размерах(Исправим это, нажав *Add New Constraints* и зафиксировав Width и Height (по 100 точек на каждый)

А для последнего элемента - нашей кнопки - смешаем оба подхода. Чтобы расположить её по вертикали, установим констрейнт сверху, размером 100 точек (и не забудем указать, что отступ должен быть от предыдущего элемента, то есть, Image View). А для расположения по горизонтали, нажмем *Align*, и затем *Horizontally in Container*. Осталось только исправить ширину (констрейнт Width), и наш контроллер шикарно выглядит даже на маленьком экране!

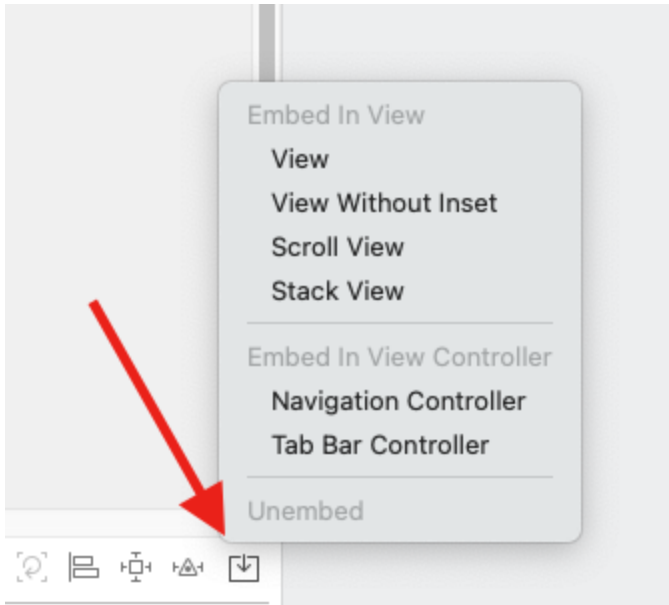


Если в какой-то момент нам захочется изменить значение констрейнта (или удалить его), мы можем найти все констрейнты элемента в Size Inspector'e



Перед тем как наш интерфейс станет еще более универсальным, поговорим про **контейнеры**. Контейнер - это элемент, внутри которого находятся другие элементы (чаще всего он имеет тип *UIView* или просто *View*). В нашем сториборде уже есть такой - это View контроллера, на котором мы располагали другие элементы (тот, который занимает всю площадь экрана).

В иерархии видов элемент, который вмещает в себя другие, называется **superview**, а помещенный внутрь него - **subview**. Чтобы разобраться в этих понятиях, добавим новый контейнер для нашей кнопки. Выделим её, нажмём *Embed In* и выберем *View*



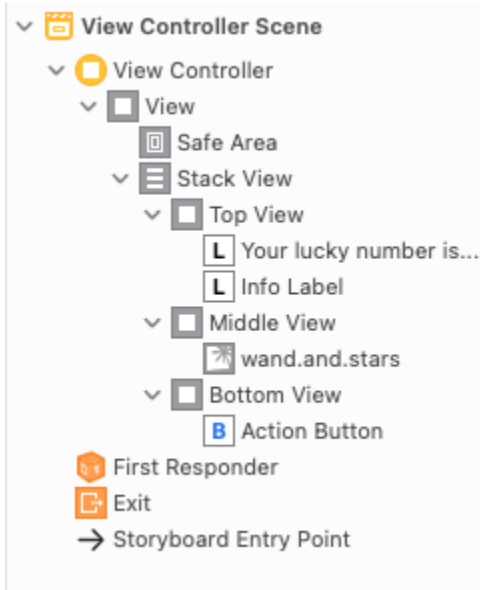
Кнопка попала в контейнер. Переименуем его в дереве файлов и назовем "*Bottom View*". Если посмотреть на дерево файлов, можно увидеть, что некоторые из них вложены друг в друга как в папки. Так мы можем проследить нашу иерархию:

View это *superview* для Bottom View
 Bottom View это *subview* для View
 Action Button это *subview* для Bottom View
 Bottom View это *superview* для Action Button

Надеюсь, никто не запутался в бесконечных view view view... Главное, что нужно понять - одни элементы могут быть вложены в другие, и зависеть от них (например, выстраивать констрейнты для элемента лучше всего относительно его *superview*)

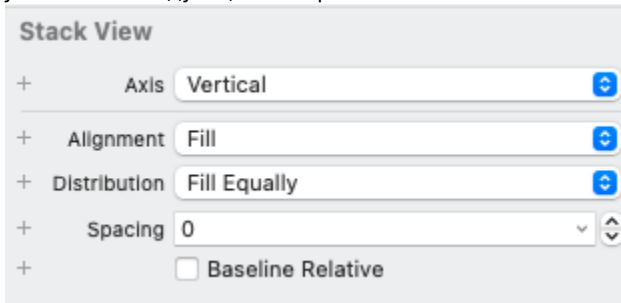
Повторим историю с контейнерами еще дважды - встроим во View нашу картинку и назовем контейнер Middle View. А затем выделим оба Label'a (можно зажать *cmd*) и встроим их в свой контейнер с именем Top View. Таким образом мы поделили экран на три блока. Попробуем расположить их вплотную, заставив подстраиваться под любую высоту экрана. Это позволит нам запускать приложение в альбомной ориентации, сохранив верстку и интерфейс! Удалим все констрейнты и встречаем... Stack View!

Стек в буквальном переводе - это стопка (представьте стопку книг, например). Точно так же мы сделаем стопку из наших контейнеров. Для этого выделим все три, нажмем *Embed In* и выберем *Stack View*.



Иерархия снова поменялась! View - это контейнер для Stack View, а он в свою очередь вмещает Top View, Middle View и Bottom View. Сделаем так, что бы стек занимал всю площадь экрана - добавим ему констрейнты (0 с каждой стороны) относительно View

Главная фишка Stack View - элементы внутри него не нуждаются в констрейнтах! Стек будет сам подстраивать свои subview под нужный размер, а мы можем лишь изменять параметры. Для нашего стека в инспекторе атрибутов установим следующие настройки:

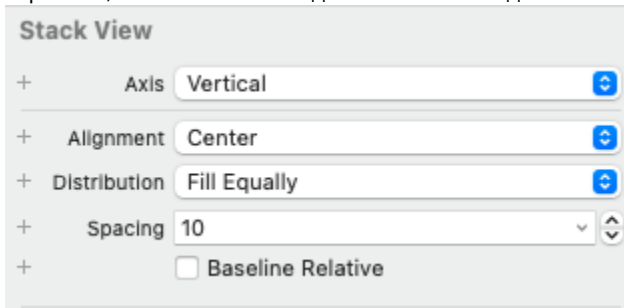


Таким образом три наших контейнера всегда будут равномерно заполнять площадь стека и вплотную прижиматься друг к другу. А их subview снова расположим при помощи Auto Layout

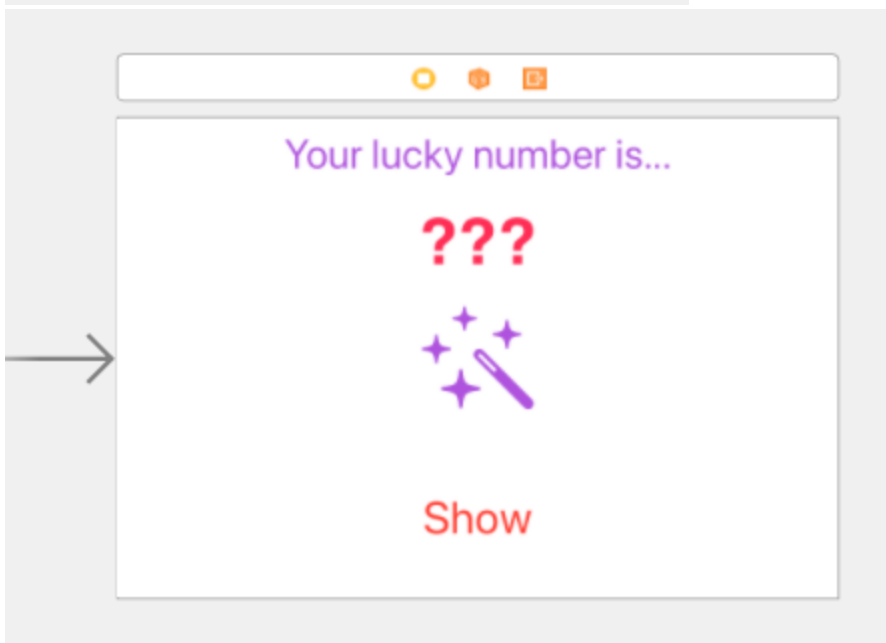
Начнем с нижнего контейнера. Кнопку внутри него выровняем по центру с помощью Align.

Для Middle View “отцентрируем” картинку по оси X, а сверху и снизу добавим констрейнты по 10 точек каждый. И чтобы картинка перестала сжиматься, установим Aspect Ratio 1:1.

Чуть сложнее обстоят дела с Top View - в нем содержатся два элемента. И чтобы не перегружать проект сложной версткой, их мы тоже объединим в стек! Выделяем оба Label'a и выбираем Embed In → Stack View



Наши лейблы автоматически выровнены по центру, равномерно заполняют выделенное пространство, и соблюдают дистанцию в 10 точек между собой. Финальный шаг - “прикрепить” только что созданный стек к контейнеру. Установим нулевые контрейнты со всех сторон для Stack View



...и наше приложение отлично вписывается в любой экран, в любом положении!