**Разработка приложения для рисования**

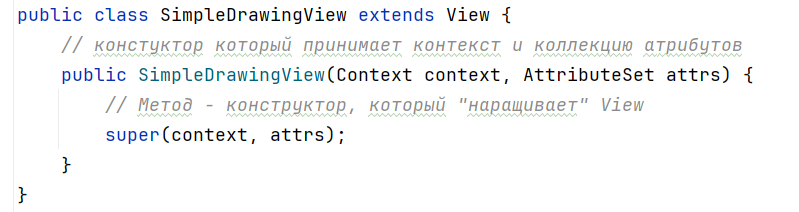
Разработайте приложения для рисования на экране. Пользователь может рисовать на экране, нажимая пальцем.

**Создание пользовательского элемента интерфейса**

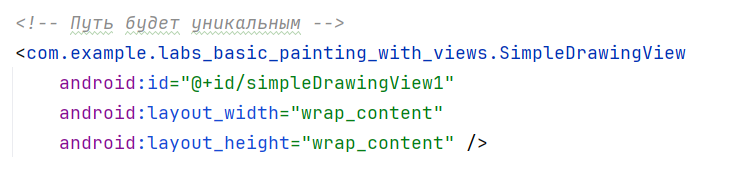
Для рисования элементов на экране используется класс Canvas позволяющий программно создавать Drawable файлы и отрисовывать их на пользовательском интерфейсе. Хотя уже предустановленные элементы интерфейса поддерживают взаимодействие с Canvas позволяя использовать их для рисования, вы разработаете собственный элемент установив в нем функции необходимые для выполнения задачи. Такой подход позволяет сократить количество используемого кода, используя и модифицируя данный класс многократно, а также персонализировать его функциональность.

Создайте простой класс, пусть он наследует стандартный View.

Установите в классе конструктор который принимает контекст текущего экрана и коллекцию необходимых атрибутов:



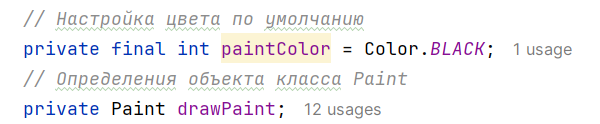
Добавьте ваш элемент в макет интерфейса, путь к элементу будет зависеть от расположения созданного класса:



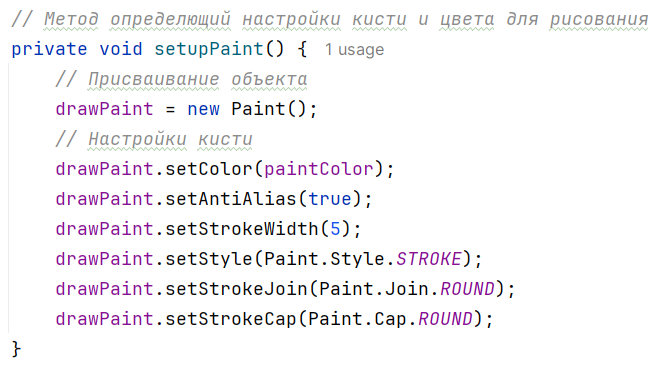
**Простой рисунок с помощью Canvas**

Для примера нарисуем несколько окружностей разного цвета используя класс Canvas. Для этого необходимо определить объект класса Paint который управляет стилем и цветом рисуемого.

Инициализируйте переменные для дальнейшей работы внутри класса, чтобы ими могли пользоваться все методы класса.

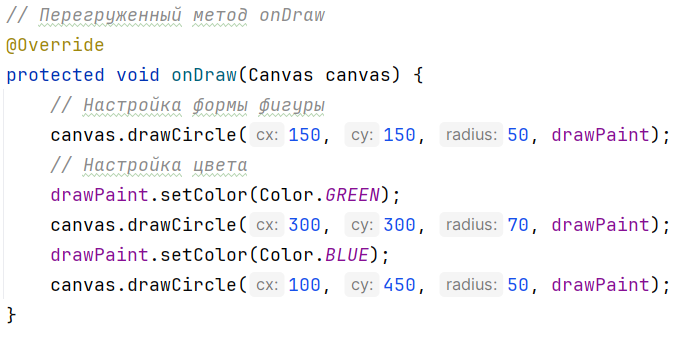


Далее отдельно настройте переменную-объект класса Paint: настройте цвет и тип кисти которая будет отвечать за рисование:



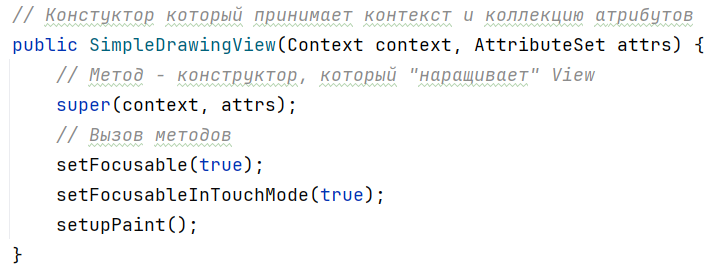
Теперь, когда у вас есть настроенная кисть создайте метод который будет рисовать фигуры, для примера нарисуем три круга разных цветов.

Функцию рисования необходимо установить в рамках метода onDraw() - это базовый метод вашего родительского класса View, который автоматически вызывается для отображения элемента на экране:

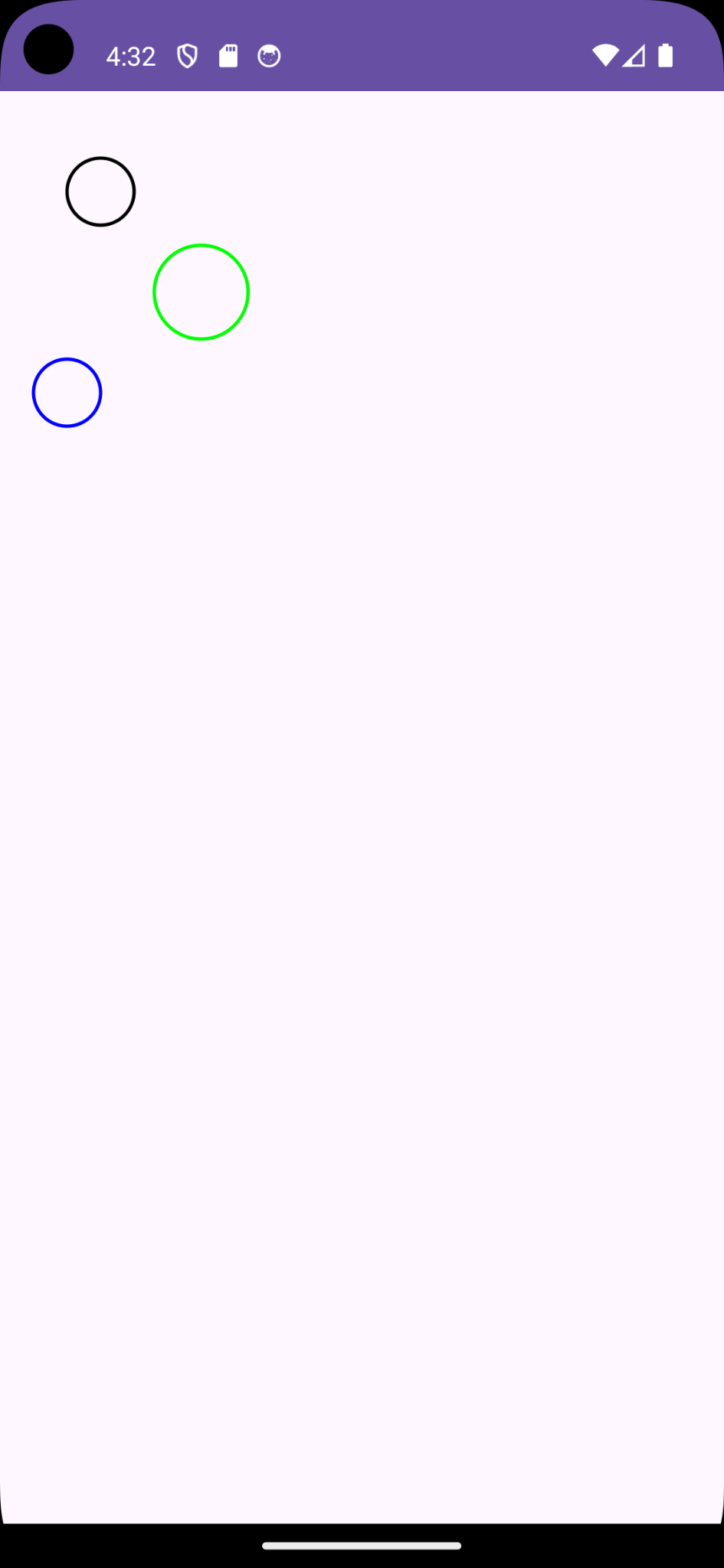


Обратите внимание на то что метод принимает объект класса Canvas, данному объекту для рисования присваиваем созданный и настроенный ранее объект класса Paint. Метод drawCircle() принимает значения x, y и радиус окружности в дополнение к Paint. Установите значения x, y и радиуса по своему усмотрению чтобы изучить работу метода drawCircle.

Наконец дополните конструктор вызовом созданных методов:



Итоговый результат:



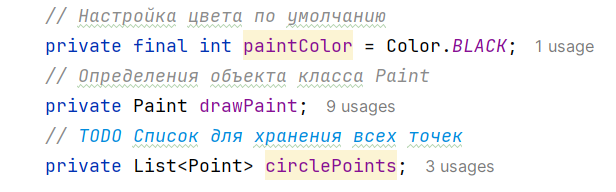
**Рисования фигур при сенсорном взаимодействии**

Для разработки приложения для рисования необходимо не только уметь отрисовывать элементы на экране но и обрабатывать нажатия пользователя на экран.

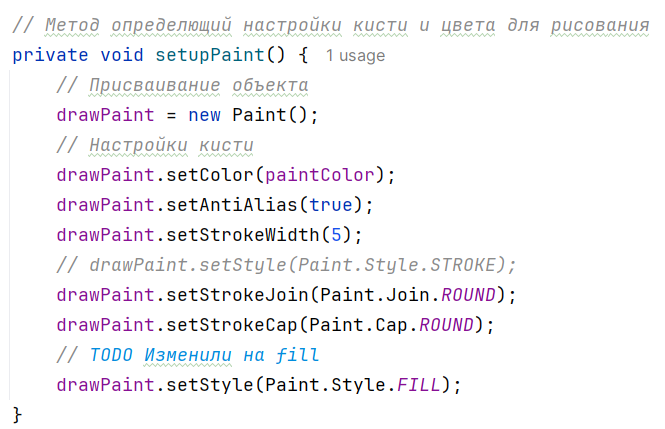
Каждый раз когда пользователь нажимает на экран, вы хотите рисовать на нажатом месте фигуру. Для данной задачи предпочтительно использовать фигуры-точки (Point). Будет необходимо чтобы на экране находилось множество точек, причем вы не можете определить не их местоположение, не их количество заранее, до нажатия пользователя. Это требует создания массива точек в котором будут записываться новые координаты для новой точки при каждом срабатывании метода onTouch().

Создайте новый класс и перенесите программный код из предыдущего для модификации.

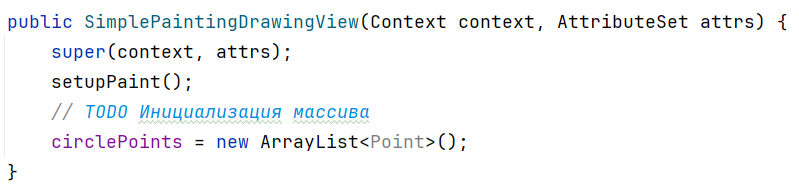
Создайте список для точек:



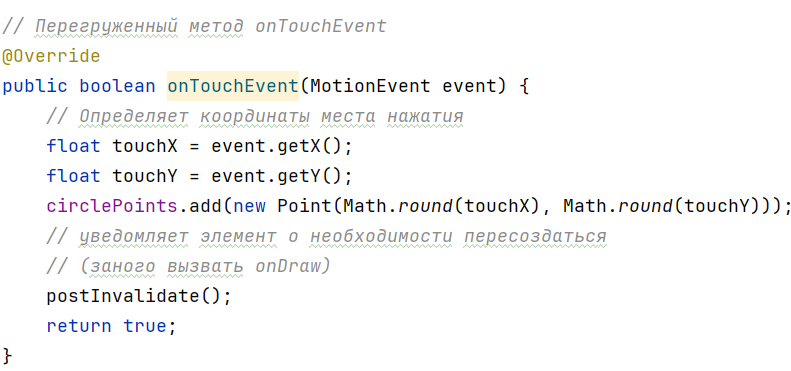
Немного поменяйте метод setupPaint():



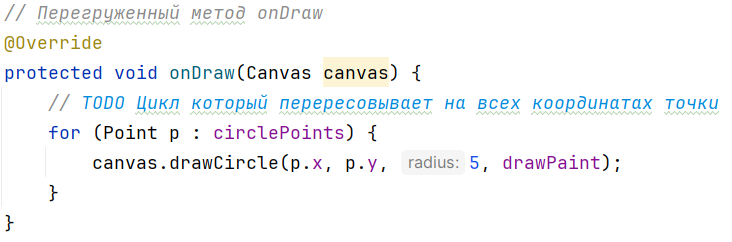
Инициализируйте список для хранения точек в конструкторе:



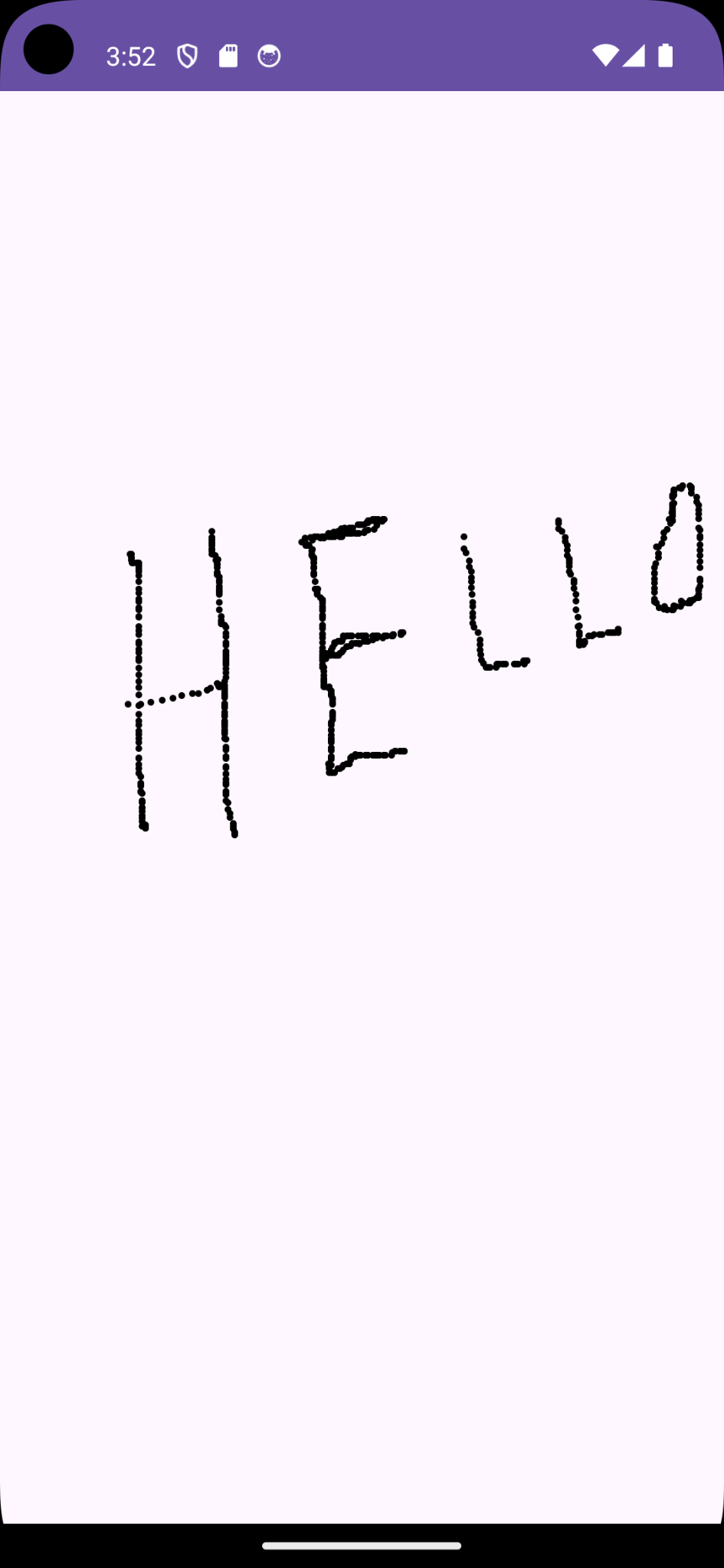
Создайте метод onTouchEvent() - базовый метод срабатывающий при нажатии на элемент. В методе реализуйте функции определения координат и сохранения их в массив. Вызовите метод postInvalidate() который уведомляет элемент о том что нужно перерисовать свой интерфейс:



Метод onDraw должен отрисовывать точки на всех координатах из списка:



Добавьте ваш элемент на макет интерфейса и запустите приложение. При нажатии на любое место внутри элемента на этом месте будет отрисовываться небольшая точка, которые в совокупности формируют рисунок:



**Рисование фигур с помощью Path**

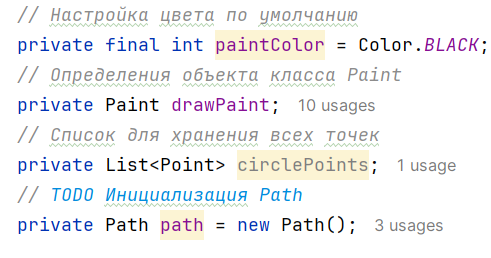
Использование класса Canvas приемлемо при рисовании простых фигур, но для рисования сложных фигур с нестандартной компоновкой этот класс нежелателен для использования.

На примере работы прошлого элемента вы видите присутствие пустых пространств между точками, это связано с тем что каждая точка это отдельная не связанная с другими фигура, рисовать пальцем используя маленькие точки довольно неудобно.

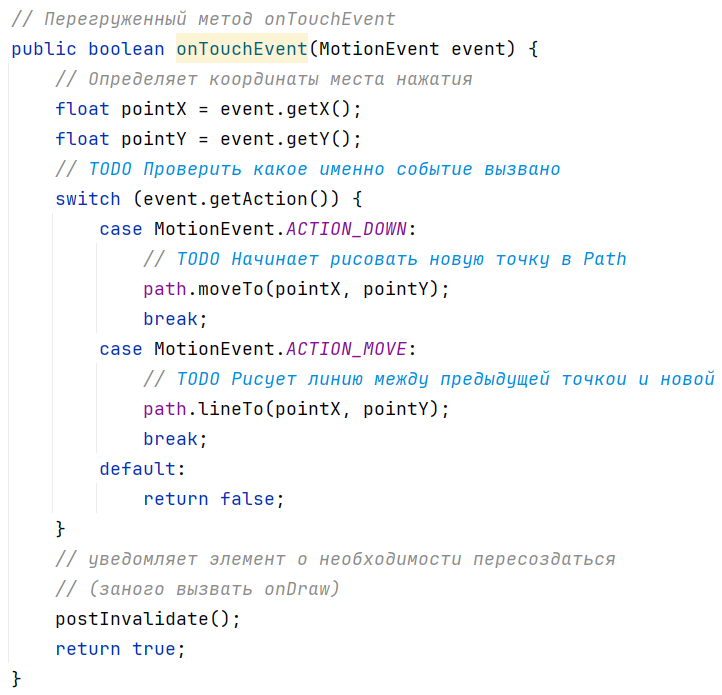
Для вашей задачи идеально подходит другой класс для рисования - Path. Path может содержать множество линий, контуров и фигур.

Создайте новый класс и перенесите программный код из предыдущего для модификации.

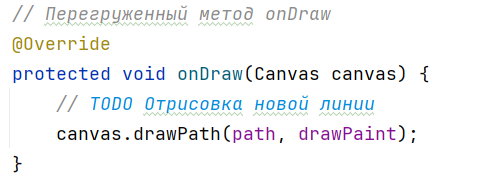
Инициализируйте объект класса Path



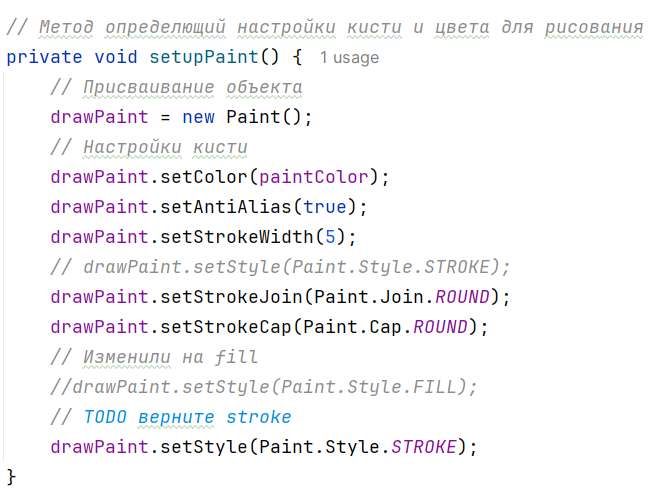
Когда пользователь нажимает на экран должна вызываться функция рисования точки, когда пользователь двигает пальцем точки формируются в единую линию. Для этого измените метод onTouchEvent(), разработав в нем две вышеназванные функции и используйте операторы выбора чтобы определять какое именно событие нажатия произошло:



Удалите из метода onDraw() функцию рисования точек и вместо этого отобразите линии:



Немного поменяйте метод setupPaint():



Добавьте ваш элемент на макет интерфейса и запустите приложение. При нажатии на любое место внутри элемента на этом месте будет отрисовываться линия:

