

# РОБОТ ДЛЯ РАЗПОЗНАВАНИЯ БЛАНКОВ ОТВЕТОВ

Выполнил: Колмачихин Александр

Преподавателям нужно проверять знания своих студентов

Самостоятельно



30–50 минут

С помощью робота



3–5 минут

Целью работы является разработка приложения для распознавания бланков ответов и дальнейшее его использование в работе



Для достижения цели были поставлены задачи:

Выбор **платформы** и **языка** для разработки

Проектирование **бланка** для ответов

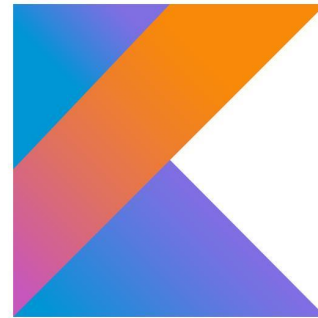
**Программирование** приложения и алгоритмов распознавания

**Рассмотрение** реализации разработанного приложения **в работе**

## Платформа и язык разработки приложения



Android Studio



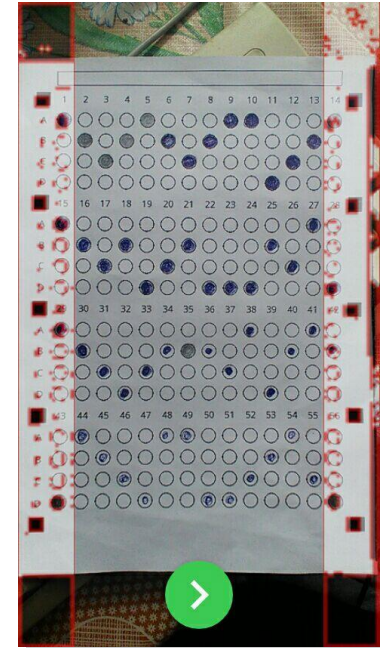
Kotlin



## Алгоритм поиска ответов на фотографии

### 1. Фотографирование бланка

Нужно привести камеру так, чтобы все 10 квадратов были в своих красных зонах.



## Алгоритм поиска ответов на фотографии

### 2. Перевод изображения в чёрно-белый формат

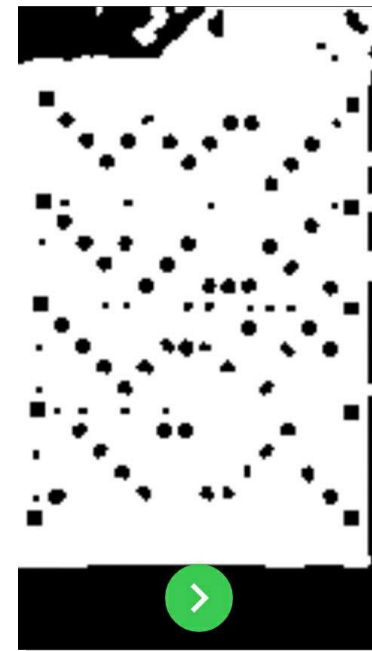
Пиксель считается черным если:

Красный  $\leq 130$

Зеленый  $\leq 130$

Синий  $\leq 255$

```
fun isBlack(r: Int, g: Int, b: Int) = r <= 130 && g <= 130 && b <= 255
```

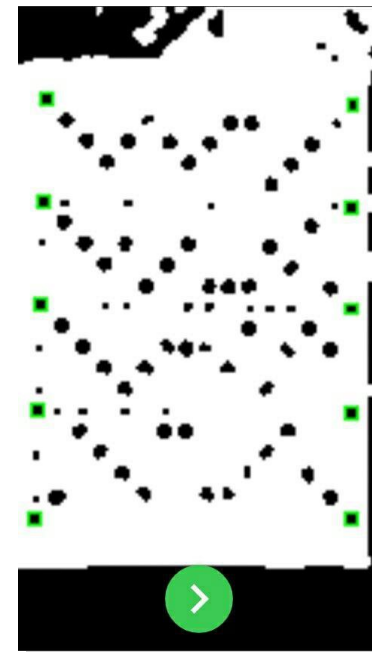




## Алгоритм поиска ответов на фотографии

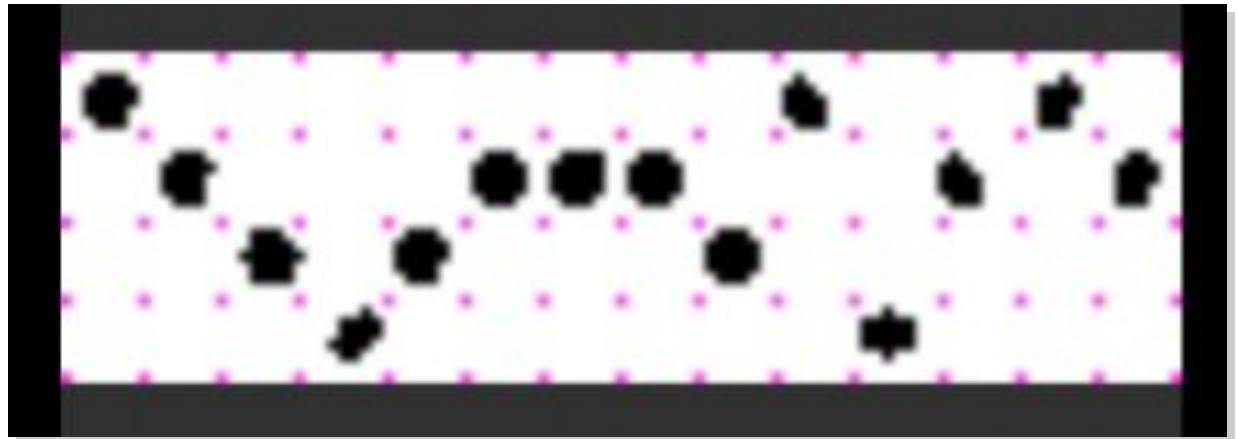
### 3. Поиск квадратов в красных зонах

Распознавание происходит по особым признакам характерных для квадратов с некоторой погрешностью.



## Алгоритм поиска ответов на фотографии

### 4. Поиск ответов между найденными квадратами



## Результат сканирования

Правильные ответы отмечены зеленым, а не правильные красным.

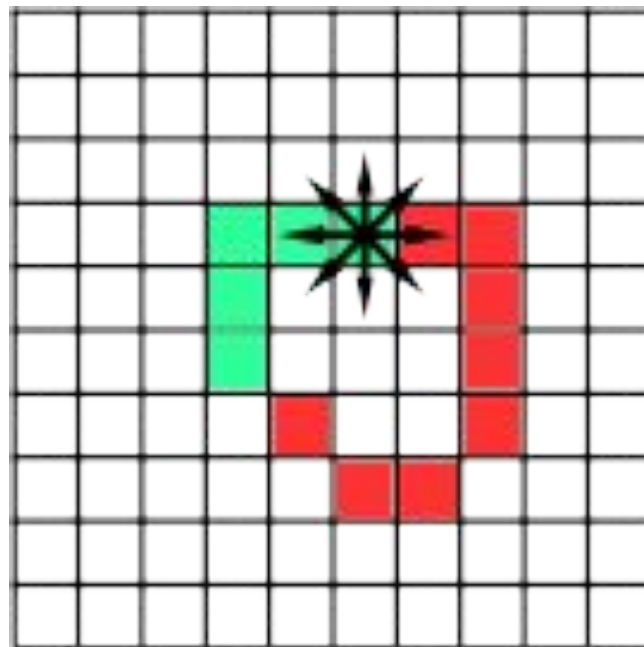


## Алгоритм поиска контуров

Алгоритм проходит по каждой точке контура и записывает ее координаты.

Цикл длится пока хотя бы одна из стрелок указывает на красный квадрат.

Если найден красный квадрат, то курсор перемещается в эту точку и записывает ее координаты в контур.



## Класс Contour

Все считанные точки хранятся как список объектов класса Point.

```
class Contour {  
    val list = ArrayList<Point>()  
  
    fun width() = maxX() - minX() + 1  
    fun height() = maxY() - minY() + 1  
  
    fun set(p: Point) {  
        var i = 0  
        list.forEach {  
            if (p == it) {  
                list[i] = p  
                return  
            }  
            i++  
        }  
        list.add(p)  
    }  
  
    operator fun get(x: Int, y: Int): Point? {  
        list.forEach {  
            if (x == it.x && y == it.y) {  
                return it  
            }  
        }  
        return null  
    }  
}
```

## Распознавание квадратов

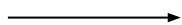
Метод `isMaybeSquare` принимает на вход готовый контур и проверяет его характеристики на сходство с квадратом.

```
fun isMaybeSquare(contour: Contour): Boolean {  
    val width = contour.width()  
    val height = contour.height()  
    return contour.size in 12..50  
        && contour.isFirstNearWithLast()  
        && getRectPercent(contour) >= 70  
        && abs(width - height) <= 3  
}
```

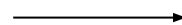
## Будущая реализация в работе



Бланки студентов



Робот-сканер



Сервер

Спасибо за внимание