**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Прикладная математика и информатика»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Руководитель проекта,  Яндекс, Старший программист  \_\_\_\_\_\_\_\_\_\_\_ / А.А. Сапрыкин /  «22» марта 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Прикладная математика и информатика»  доцент, канд. физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.С. Конушин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

**АССИСТЕНТ ДЛЯ ВОДИТЕЛЯ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 81 01-1-ЛУ**

Исполнитель:

студент группы БПМИ175

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / С.В. Булгаков /

«22» марта 2019 г.

**Москва 2019**

УТВЕРЖДЕН

RU.17701729.04.01-01 81 01-1-ЛУ

**АССИСТЕНТ ДЛЯ ВОДИТЕЛЯ**

**Пояснительная записка**

**RU.17701729.04.01-01 81 01-1**

**Листов 12**

Москва 2019

Содержание

[Введение 3](#_Toc4424987)

[Документы, на основании которых ведется разработка 3](#_Toc4424988)

[Наименование темы разработки 3](#_Toc4424989)

[Назначение и область применения 4](#_Toc4424990)

[Технические характеристики 5](#_Toc4424991)

[Постановка задачи на разработку программы 5](#_Toc4424992)

[Описание и обоснование алгоритма и функционирования программы 5](#_Toc4424993)

[Функционирование системы контроля версий 5](#_Toc4424994)

[Алгоритм и функционирование сборочной системы 5](#_Toc4424995)

[Алгоритм и функционирование модуля приложения 6](#_Toc4424996)

[Алгоритм и функционирование модуля распознавателей 7](#_Toc4424997)

[Функционирование модуля абстрактного кода 7](#_Toc4424998)

[Описание и обоснование выбора метода организации входных и выходных данных 9](#_Toc4424999)

[Описание и обоснование выбора состава технических и программных средств 9](#_Toc4425000)

[Технико-экономические показатели 10](#_Toc4425001)

[Предполагаемая потребность 10](#_Toc4425002)

[Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 10](#_Toc4425003)

[Список использованных источников 11](#_Toc4425004)

# Введение

## Документы, на основании которых ведется разработка

1. Приказ Национального исследовательского университета "Высшая школа экономики" (номер, дата: 2.3-02/1501-03, 15.01.2019).

## Наименование темы разработки

Наименование темы разработки – «Ассистент для водителя».

Условное обозначение темы разработки – «Driver assistant».

Программа выполняется в рамках темы программного проекта в соответствии с учебным планом подготовки бакалавров по направлению 01.03.02 «Прикладная математика и информатика» Национального исследовательского университета «Высшая школа экономики», факультет компьютерных наук.

# Назначение и область применения

«Ассистент для водителя» – программа, предназначенная для запуска на смартфоне или планшете водителя, следящая за ситуацией на дороге. Программа выполняет функцию видеорегистратора – ведет непрерывный захват видео потока, а также предоставляет функцию распознавания изображения. Мобильное устройство требуется расположить так, чтобы его камера обозревала направление движения транспортного средства, например, непосредственно на лобовое стекло. Программа может помочь не только начинающим водителям. Имеется множество применений:

* Запись истории возникновения ДТП с полным набором собираемых данных и последующее использование этих данных для определения виновника ДТП.
* Запись истории возникновения ДТП с полным набором собираемых данных и последующее использование этих данных в качестве обучающего материала в автошколах.
* Сбор информации о линиях дорожной разметки и других распознаваемых объектах для последующего объединения этой информации в единую базу данных.

# Технические характеристики

## Постановка задачи на разработку программы

Разрабатываемая программа должна:

1. Отображать видеопоток на главном экране.
2. Иметь возможность распознавать изображение и предоставлять в качестве примера следующие распознаватели:
   1. Случайный распознаватель.
   2. Распознаватель линий дорожной разметки.
3. Иметь возможность показывать распознанные объекты поверх видеопотока на главном экране.
4. Иметь возможность захватывать видеопоток.
5. Иметь возможность сохранять лог распознанных объектов.

## Описание и обоснование алгоритма и функционирования программы

### Функционирование системы контроля версий

Использование системы контроля версий Git позволяет вести именованную историю изменений, а также дает возможность разработки приложения несколькими людьми, в том числе и для удобства добавления пользовательских распознавателей сообществом.

### Алгоритм и функционирование сборочной системы

Программа собирается с помощью сборочной системы Gradle из нескольких модулей:

* App – модуль с кодом приложения.
* Recognizer – модуль с кодом распознавателей.
* Util – модуль с абстрактным кодом.

Зависимость модулей:

App

Recognizer

Util

Благодаря разделению приложения на модули, есть возможность относительно быстрой инкрементальной компиляции, когда компилируются не вся программа, а только измененные с последней компиляции модули. Более того, в силу независимости модулей Reconizer и Util их компиляцию сборочная система имеет возможность выполнять параллельно (при необходимости компиляции их обоих).

### Алгоритм и функционирование модуля приложения

Основное окно приложения работает на базе конечного автомата и использует несколько предопределенных состояний. Это связано со следующими причинами:

* Экран Android приложения всегда находится в некотором состоянии и может его менять.
* Пользователь с помощью элементов управления может подать команду на изменение текущего состояния, но система Android не всегда выполняет смену состояния сразу же. Это верно в основном для операций с камерой.

Модуль приложения содержит код доступных внутри приложения переходов между состояниями, а также код самих состояний.

Действия, которые могут менять состояния:

* ActivityResumed/ActivityPaused – изменение состояния экрана приложения.
* SurfaceTextureAvailable – активация компонента для отображения видеопотока.
* CameraOpened/CameraClosed – активация камеры, выводящей видеопоток на компонент.
* RecognizerImageButtonClicked/RecordSwitch – нажатие пользователем соответствующих кнопок.
* PreviewCaptureSessionConfigured/RecordingCaptureSessionConfigured – активация записывающей сессии камеры при отсутствии/осуществлении записи видеопотока соответственно.
* ImageShot – получение очередной картинки для работы распознавателем.

Дополнительно модуль приложения содержит код пользовательского интерфейса:

* Главный экран приложения отображает картинку с камеры и имеет переключатели видеозаписи и распознавания. Также на главном экране находится нестандартный компонент отображения распознанных объектов.
* Нестандартный компонент отображения распознанных объектов позволяет отрисовать переданный список объектов.
* Возможные для отрисовки элементарные объекты, которые создаются из данных о распознанных объектов.

В заключение, модуль приложения содержит код для логирования распознанных объектов в память устройства.

### Алгоритм и функционирование модуля распознавателей

Модуль распознавателей предоставляет интерфейсы и классы, которые может использовать любое Android приложение.

Основной интерфейс распознавателя имеет единственный метод recognize, принимающий подготовленную картинку и возвращающий список распознанных объектов.

Подготовленная картинка является набором сырых байт JPEG кодировки изображения.

Дополнительно модуль предоставляет примеры реализаций двух распознавателей:

* Случайный распознаватель. Не использует данные с камеры, но рисует случайные 10 красных стрелок.
* Распознаватель линий дорожной разметки. Применяет к изображению алгоритм Кэнни и преобразование Хафа с фильтрацией результатов для получения высококонтрастных линий на изображении.

### Функционирование модуля абстрактного кода

Модуль предоставляет следующее.

1) Код абстрактного конечного автомата:

**open class** StateMachine<**in** ActionType : Action, **out** StateType : State<ActionType, StateType>>(initialState: StateType) {  
 **private var currentState**: StateType = initialState  
  
 **fun** make(action: ActionType) {  
 *synchronized*(**this**) **{  
 try** {  
 **val** nextState = **currentState**.consume(action)  
  
 **if** (nextState !== **currentState**) {  
 Log.i(**TAG**, **"${currentState**.*javaClass*.*simpleName***} -> ${**nextState.*javaClass*.*simpleName***}"**)  
 }  
  
 **currentState** = nextState  
 } **catch** (t: Throwable) {  
 **throw** StateSwitchException(**"Can't switch to the new state"**, t)  
 }  
 **}** }  
}

Позволяет переключать состояния в реалиях многопоточного Android кода.

2) Код абстрактного действия:

**abstract class** State<**in** ActionType : Action, **out** StateType : State<ActionType, StateType>> {  
 **open fun** consume(action: ActionType): StateType {  
 **throw** IllegalStateException(**"Invalid action ($**action**) have been passed to this state ($this)"**)  
 }  
}

Предполагает, что классы будут переопределять метод, обрабатывать нужные действия, а для недоступных действий вызывать метод обработки действия суперкласса. Таким образом, позволяет создавать иерархию наследования состояний.

3) Классы для хранения данных о камере нужны, чтобы разделять данные о камере, которые получаются на разных стадиях ее инициализации. Реализованы три класса: настроенная камера, снимающая без записи камера, записывающая камера.

4) Методы для работы с файловой системой Android: создание папки и файла для видео.

5) Перечисление разрешений для работы с системой разрешений Android.

6) Методы для пользовательского интерфейса Android: выполнение функции на UI-потоке и перевод приложения в полный экран.

## Описание и обоснование выбора метода организации входных и выходных данных

Для возможности гибкой настройки камеры и полного управления ею было решено использовать Camera2 API. В силу распространенности формата JSON было решено использовать его для сохранения логов работы программы.

## Описание и обоснование выбора состава технических и программных средств

1. Устройство, оснащенное операционной системой Android версии не ниже 5.0 для возможности использования Camera2 API.
2. Не менее 250 МБ доступного для записи места на встроенном накопителе для хранения самого приложения. Около 200 МБ занимает библиотека OpenCV для распознавания изображения. 50 МБ с запасом хватит для кода приложения. Для функции захвата видео потока требуется дополнительное место, вычисляемое как произведение выбранных пользователем битрейта и длительности сохраняемого фрагмента. Для функции сохранения лога требуется дополнительное место из расчета не более одного МБ на кадр.
3. 256 МБ свободной в момент запуска приложения оперативной памяти или больше для возможности полной загрузки приложения в оперативную память.
4. Дисплей с диагональю не меньше пяти дюймов с сенсорным экраном для крупности картинки и возможности управления.
5. Камера и процессор, позволяющие снимать видео в разрешении не менее разрешения дисплея.

# Технико-экономические показатели

## Предполагаемая потребность

Программа может помочь не только начинающим водителям. Имеется множество применений:

* Запись истории возникновения ДТП с полным набором собираемых данных и последующее использование этих данных для определения виновника ДТП.
* Запись истории возникновения ДТП с полным набором собираемых данных и последующее использование этих данных в качестве обучающего материала в автошколах.
* Сбор информации о линиях дорожной разметки и других распознаваемых объектах для последующего объединения этой информации в единую базу данных.

## Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Быстрый поиск в сети Интернет на момент создания приложения не выявил аналогов данной программы.

Данное приложение:

1. распространяется бесплатно и открыто;
2. не требует вложения денежных средств во время использования;
3. имеет неограниченный срок службы, так как работает автономно;
4. приглашает независимых разработчиков внести вклад, реализовав уникальные распознаватели с помощью предоставленного интерфейса.

# Список использованных источников

1. android camera2 api video [Электронный ресурс] // URL: <https://www.youtube.com/playlist?list=PL9jCwTXYWjDIHNEGtsRdCTk79I9-95TbJ> (Дата обращения: 22.03.2019, режим доступа: свободный).
2. Android Studio Tutorial - Camera 2 API FULL [Электронный ресурс] // URL: <https://www.youtube.com/watch?v=oPu42I0HSi4> (Дата обращения: 22.03.2019, режим доступа: свободный).
3. Android Camera2Basic Sample [Электронный ресурс] // URL: <https://github.com/googlesamples/android-Camera2Basic> (Дата обращения: 22.03.2019, режим доступа: свободный).
4. Android Camera2Video Sample [Электронный ресурс] // URL: <https://github.com/googlesamples/android-Camera2Video> (Дата обращения: 22.03.2019, режим доступа: свободный).
5. Практический пример создания собственного View-компонента [Электронный ресурс] // URL: <https://habr.com/ru/post/433782/> (Дата обращения: 22.03.2019, режим доступа: свободный).
6. Custom View Components: Custom Drawing [Электронный ресурс] // URL: <https://developer.android.com/training/custom-views/custom-drawing> (Дата обращения: 22.03.2019, режим доступа: свободный).
7. Error inflating when extending a class [Электронный ресурс] // URL: <https://stackoverflow.com/questions/3739661/error-inflating-when-extending-a-class> (Дата обращения: 22.03.2019, режим доступа: свободный).
8. Рисование. Доступ к Canvas [Электронный ресурс] // URL: <https://startandroid.ru/ru/uroki/vse-uroki-spiskom/311-urok-141-risovanie-dostup-k-canvas.html> (Дата обращения: 22.03.2019, режим доступа: свободный).
9. Finite State Machines + Android + Kotlin = Good Times [Электронный ресурс] // URL: <https://robots.thoughtbot.com/finite-state-machines-android-kotlin-good-times> (Дата обращения: 22.03.2019, режим доступа: свободный).
10. How to define common android properties for all modules using gradle [Электронный ресурс] // URL: <https://stackoverflow.com/questions/20404476/how-to-define-common-android-properties-for-all-modules-using-gradle> (Дата обращения: 22.03.2019, режим доступа: свободный).
11. Android studio doesn't recognise source folders [Электронный ресурс] // URL: <https://stackoverflow.com/questions/22068731/android-studio-doesnt-recognise-source-folders> (Дата обращения: 22.03.2019, режим доступа: свободный).
12. Репозиторий проекта [Электронный ресурс] // URL: <https://github.com/driver-assistant/driver-assistant> (Дата обращения: 22.03.2019, режим доступа: свободный).