

УДК 004.5

DOI: 10.46548/21vek-2020-0952-0005

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ НАВИГАЦИИ С ПРИМЕНЕНИЕМ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

© 2020

Мартышкин Алексей Иванович, кандидат технических наук, доцент,
доцент кафедры «Вычислительные машины и системы»

Пензенский государственный технологический университет
(440039, Россия, Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11, e-mail: alexey314@yandex.ru)

Пашенко Татьяна Юрьевна, кандидат экономических наук, доцент,
доцент кафедры «Менеджмент и экономическая безопасность»

Пензенский государственный университет
(440026, Россия, Пенза, улица Красная, 40, e-mail: tania.paschenko@gmail.com)

Бершадская Елена Григорьевна, кандидат технических наук, профессор,
профессор кафедры «Вычислительные машины и системы»

Пензенский государственный технологический университет
(440039, Россия, Пенза, проезд Байдукова/ул. Гагарина, д. 1а/11, e-mail: bereg.50@mail.ru)

Аннотация. В настоящей статье приводится описание процесса разработки программного обеспечения информационной системы указания пути с применением дополненной реальности. В начале статьи рассматриваются вопросы, связанные с постановкой задачи исследования. Определяются цель и решаемые задачи для ее достижения. Приводятся теоретические сведения и наработки в рассматриваемой предметной области. Во второй части рассматривается архитектура разрабатываемой программы, приведено обоснование выбора языка программирования. Третья часть содержит непосредственно решение вопросов разработки алгоритмов и программы. Следующий раздел посвящен описанию руководства оператора. В заключении подводится итог о проделанной работе, делаются соответствующие выводы о полученных результатах. Подчеркивается, что результатом работы является программное обеспечение для создания маршрута и передвижения по нему в режиме дополненной реальности. Отмечается, что в качестве расширения и доработки данного программного продукта можно рассматривать разработку программного обеспечения под другие мобильные операционные системы (например, iOS и Windows Phone).

Ключевые слова: дополненная реальность, карты, мобильные приложения, навигация, Android, Java.

MAP DIRECTIONS SYSTEM APPLICATION UTILIZING AUGMENTED REALITY TECHNOLOGY

© 2020

Martyshkin Alexey Ivanovich, candidate of technical sciences, docent,
associate Professor of sub-department «Computers and systems»

Penza state technological University
(440039, Russia, Penza, Baydukov Proyezd / Gagarin Street, 1a/11, e-mail: alexey314@yandex.ru)

Pashchenko Tatyana Yuryevna, candidate of economic Sciences,
associate Professor of sub-department «Management and economic security»

Penza State University
(440026, Russia, Penza, Krasnaya Street, 40, e-mail: tania.paschenko@gmail.com)

Bershadskaya Elena Grigorevna, candidate of technical sciences,
professor, professor of sub-department «Computers and systems»

Penza state technological University
(440039, Russia, Penza, Baydukov Proyezd / Gagarin Street, 1a/11, e-mail: bereg.50@mail.ru)

Abstract. This paper describes the development process for the map directions information system application utilizing augmented reality technology. The first part of this paper discusses issues related to the research objective formulation. The objective and tasks to be solved are determined. Theoretical information and developments in the subject area are discussed. The second part describes the developed application architecture, the choice of a programming language is explained. The third part expressively addresses the development of algorithms and the application itself. The next section contains instructions for end users. And finally, a summary of the concluded work is given and relevant conclusions are made about the obtained results. It is emphasized that the work leads to the creation of an application for route finding and navigation in augmented reality mode. It is noted that this application can be extended and enhanced by developing versions for other mobile operating systems (for example, iOS and Windows Phone).

Keywords: augmented reality, maps, mobile applications, navigation, Android, Java.

Введение. Популярность мобильных устройств возрастает ежедневно, и это не удивительно, ведь сегодня с их помощью можно просто найти любую

необходимую информацию или выполнить сложную операцию, не прикладывая больших усилий. Одним из самых главных факторов популярности телефонов

перед PC и планшетами служит доступность гаджета и его мобильность. Сегодня одним из наиболее перспективных направлений развития информационных технологий является применение дополненной реальности – среды с дополнением физического мира цифровыми данными, которые воспринимаются как элементы реальной жизни [1, 2]. Если пользователь отклонится от маршрута, приложение поможет ему найти направление к назначенному месту с помощью указателя в дополненной реальности (AR). Итак, настоящая статья направлена на разработку программного обеспечения для мобильного приложения «AR Навигатор», и основные из ее преимуществ – современность и актуальность.

Материалы и результаты исследования. Была поставлена задача разработать качественное и доступное приложение с элементами дополненной реальности, предоставляющее возможность пользователю построить удобный маршрут. Перед разработчиком мобильного приложения стоит задача в построении маршрута до пункта назначения с использованием элементов дополненной реальности.

В приложении должны быть реализованы следующие возможности: выбор объекта; формирование пути до объекта от текущей координаты; разбиение пути на координаты и привязка к ним разных типов указателей; прорисовка указателей в дополненной реальности.

Разрабатываемое мобильное приложение предназначено для отображения текущего местоположения и построения пути к месту, выбранному на карте, с использованием указателей, прорисованных в дополненной реальности. Выбор источника входных данных происходит на основании доступности более приоритетных источников, если доступа или данных нет, то система автоматически запрашивает данные у менее приоритетного источника. Пользователь получает информацию в виде графических элементов, таких как метки, стрелки, формы и т.д. Данные в отображаемых элементах зависят от того, какой набор действий произвел пользователь.

Разрабатываемое мобильное приложение предназначено для отображения текущего местоположения и построения пути к месту, выбранному на карте, с использованием указателей, прорисованных в дополненной реальности.

На данный момент времени ОС *Android* является самой популярной ОС в мире. По данным с официального сайта ОС *Android*, на данный момент ОС работает на таких устройствах как: смартфоны, *smart*-часы, планшеты, *smart-TV*, а также успешно интегрируется в автомобиль [3]. Существует множество книг, описаний, статей и *internet*-ресурсов, посвященных написанию приложений под ОС *Android*. Всю необходимую литературу можно найти как на английском, так и на русском языках. Проанализированы следующие литературные источники по теме работы [4 – 7] и [8 – 11]. В [12] рассмотрены некоторые вопросы разработки приложения: разработка программ для мобильных

устройств под управлением ОС *Android*, базовые сведения о платформе *Android*, программное обеспечение, необходимое для разработки *Android*-приложений. Источник [13] посвящен разработке программного обеспечения на базе *Android SDK*. Примеры, которые представлены в книге, отражают реальные задачи программирования с опорой на изученный теоретический материал. Книга [14] содержит объемный теоретический материал. Описание каждого подраздела сопровождается наглядными примерами в виде шестнадцати приложений для *Android*, существующих в реальном времени и широко использующихся.

Одной из главных особенностей статьи является использование датчиков, встроенных в мобильное устройство. На сайте [15] можно найти информацию о всех видах датчиков, которые могут использоваться в мобильном устройстве, а также примеры их использования.

Для работы приложения с картами и расчетом расстояний, площадей и определением направлений с помощью сферической геометрии используется *Google Maps Android API*. С данным интерфейсом можно подробнее ознакомиться на официальном сайте *Google Maps API* [16], где можно найти необходимые уроки по теме.

На начальном этапе синтеза мобильного приложения в статье прорабатываются некоторые вопросы, среди которых, например, выбор шаблона проектирования приложения. О существующих шаблонах и их способах реализации рассказано в [17, 18].

На сегодняшний день рынок мобильных приложений является одним из самых востребованных. По статистике аналитической компании *We Are Social* на апрель 2020 год смартфоны остаются главным инструментом для выхода в сеть и генерируют больше веб-трафика, чем все другие устройства вместе взятые [19].

Архитектурные шаблоны являются важнейшей частью ПО. Они помогают сохранить код в чистоте, сделать его расширяемым и тестируемым. Существует множество архитектурных шаблонов, которые используются при андроид разработке, рассмотрим шаблоны *MVVM*, *MVC*, *MVP* [17, 18].

MVVM (Model-View-View Model). Такой подход позволяет связывать элементы представления со свойствами и событиями *View*-модели. Можно утверждать, что каждый слой этого паттерна не знает о существовании другого слоя. *MVVM* рассчитан на разработку приложений в команде.

Архитектурный шаблон *MVC (Model-View-Controller)* создавался с целью разделения бизнес логики от представления. Принцип работы этого шаблона состоит в том, что пользователь взаимодействует с контроллером, контроллер запрашивает данные у модели и заполняет *View*, который отображается пользователю.

Одним из эффективных шаблонов проектирования является *MVP (Model-View-Presenter)* т.к. он позволяет отделить пользовательский интерфейс (*View*) от самих

данных (*Model*) и производить взаимодействие между ними через посредника (*Presenter*), что упрощает написание *Unit*-тестов и упрощает масштабирование приложения. Данный подход к разработке является молодым, но стремительно набирает свою популярность за счет простоты и эффективности решения многих задач, так же данный подход будет добавлен в новую версию языка *Java*. Суть данного подхода заключается в использовании асинхронных потоков данных. Такой подход позволяет сделать приложения более устойчивыми к сетевым ошибкам и упростить обработку различных событий. Поэтому с целью повышения стабильности работы приложения был использован данный подход к программированию.

В начале работы приложения открывается карта, на которой отображается текущая локация пользователя, получение данных осуществляется через интернет. Для того чтобы использовать карту в приложении, сначала необходимо зарегистрироваться на сайте *Google Maps* чтобы получить их *API* ключ.

Архитектура программного обеспечения. Разработка приложений под ОС *Android* не требует от разработчика использования специфической аппаратной и/или программной платформы, что упрощает процесс разработки ПО. Для проектирования используется интегрированная среда разработки (*IDE*) *Android Studio* от компании *Google*, которая включает в себя все необходимые инструменты для создания, отладки и тестирования программных приложений. Так же для работы с *IDE* и компиляции приложений нужна минимальная версия *Java 6*. *Android Studio* располагает богатым функционалом, который помогает ускорить разработку приложений. Данная *IDE* включает в себя ряд необходимых средств для разработки приложений [20].

Из рассмотренных вариантов (*Swift*, *C#*, *Java*), язык программирования *Java* является наиболее подходящим для выполнения поставленных задач так как включает в себя обширную библиотеку классов, фреймворков и т.д. Поскольку выбранным языком программирования является *Java*, разработка приложения будет производиться в среде разработки *Android Studio*. Разработка структуры данных является одной из важнейших задач на начальном этапе разработки, зачастую именно данные определяют возможный функционал разрабатываемого приложения и принципы его работы. Используемый в настоящей статье архитектурный шаблон проектирования *MVP* дает возможность отделить данные от их представления, что дает возможность хранить данные в любом удобном виде и изменять их без ущерба для основной логики программы. Входные данные поступают из датчиков на мобильном устройстве, а также служб *Google*. Для работы с аппаратными датчиками применяется класс *SensorManager*, вне зависимости от того сколько датчиков будет использовано в программе все они работают независимо друг от друга и с них легко снимать показания, однако при работе с датчиками необходимо учитывать, что они значительно увеличивают энерго-

потребление.

Обычно тестирование подобных приложений с использованием аппаратных датчиков происходит на реальном устройстве и прежде, чем это сделать, лучше всего будет убедиться в наличии датчиков на вашем смартфоне, это можно сделать, посмотрев характеристики вашего телефона или написать небольшую программу для вывода показаний со всех датчиков. После получения данных они представляются в виде потока данных, которые внутри себя содержат структуру классов. Все данные, поступившие с сервисов, хранятся в локальном хранилище, что позволяет, в случае отсутствия подключения к интернету или сетевой ошибки, получать данные из локального хранилища и продолжить работу с ним. Данные с датчиков не нуждаются в длительном хранении так как данные, получаемые из них, постоянно изменяются.

Разработка алгоритмов. При запуске приложения, пользователю необходимо принять соглашение для работы с его местоположением, после чего определяется его локация и камера центрируется по его местоположению, если не удалось определить локацию то из хранилища будут браться данные о последней локации пользователя, если ее не удалось получить то на экране появится сообщение с подсказкой. После чего можно будет построить маршрут, с ним можно выполнять операции по изменению и удалению, для дальнейшего использования приложения необходимо щелкнуть на кнопку с изображением камеры для перехода на экран с камерой. Если пользователь совершил какое-то не правильное действие, то ему выведется сообщение с подсказкой в нижней части экрана, подробнее об этом написано ниже по тексту статьи.

Пользователь свободно может переключаться между экраном с камерой и картой, однако если пользователь не прошел весь маршрут и вернулся на экран с картой, то ему придется строить маршрут заново т.к. маршрут не сохраняется. Схема экранных переходов представлена на рисунке 1.



Рисунок 1 – Схема экранных переходов

Когда пользователь осуществляет переход из экрана с картой на экран с камерой, происходит передача данных о маршруте с помощью метода *Intent*, после чего полученные данные записываются в заранее подготовленные переменные и используются в коде. В классах с камерой и картой используются данные полученные с датчиков, это осуществляется путем запуска классов созданных для получения показателей с датчиков, для их запуска необходимо вызвать их метод *start()* и имплементировать интерфейс помощью которого происходит передача данных. Сразу же после открытия экрана с камерой происходит обработка данных, они поступают из интернета и датчиков в гад-

жете, после чего происходит загрузка экрана, на котором отображается камера и указатель пути к первому маркеру маршрута. Для того чтобы создать указатель необходимо использовать датчик ориентации, связать его с указателем, после чего необходимо найти азимут по выражению (1). Азимут необходим для того, чтобы найти угол между пользователем и маркером на карте.

$$\tan_{\varphi_{AB}} = \left| \frac{y_{AB}}{x_{AB}} \right| \quad (1)$$

Для отображения объекта дополненной реальности необходимо знать как азимут, так и дистанцию, дистанция вычисляется по выражению (2).

$$d_{AB} = \sqrt{\Delta x^2 + \Delta y^2} \quad (2)$$

Чтобы приложением было проще пользоваться в программе есть константы для допустимых отклонений азимута и дистанции, они используются для того, чтобы объект было проще отобразить на камере. Когда пользователь находится в целевом диапазоне с учетом допустимых отклонений то выполняется условие отображения и на экране выводится объект дополненной реальности, после чего программа получает координаты о следующем маркере и проделывает эту же работу и для него. После того как будет пройден весь маршрут, произойдет переход на экран с картой, где можно будет построить другой маршрут.

Разработка программ. Разрабатываемое мобильное приложение «AR Навигатор» предназначено для формирования указателей пути к объекту с использованием карты и камеры в дополненной реальности. Приложение должно быть написано на языке программирования *Java* и иметь функционал, перечисленный в начале статьи. Работа приложения начинается с класса *MapActivity*, при инициализации данного класса у него вызывается метод *onCreate()* в котором приложение инициализирует карту подключается к службам *Google* и запрашивает доступ к текущему местоположению и вызывает срабатывание метода для работы с маркерами на карте. Для добавления карты во фрагмент используется метод *createMapView()*, подключаемый к службам *Google* и отображающий карту в случае успешного подключения к интернету, но если карту не удалось создать то выведется сообщение об ошибке. После добавления карты можно приступить к построению маршрута для этого необходимо знать координаты пользователя и добавить маркеры, соединенные линией, после чего необходимо получить широту и долготу устройства и присвоить их значение. В классе *CameraViewActivity* реализуется основная задумка программы, при переходе на это *Activity* срабатывает метод *OnCreate()* после чего происходит цепочка событий: получение данных о маршруте из класса *MapActivity*, вызов конструкторов класса *MyCurrentAzimuth* и *MyCurrentLocation* и выполнение их методов *start()*, инициализация элементов экрана и отображение камеры, создание экземпляра объекта дополненной реальности.

Обработка и вывод данных на экран происходит в классах *Activity*, после чего собранные данные передаются в необходимый класс посредством вызова методов из классов по сбору показаний. Для полноценного функционирования приложения нужны: смартфон с подключением к интернету и *GPS*; наличие необходимых датчиков в мобильном устройстве;

Если реальное устройство отсутствует, программу можно установить и проверить ее работоспособность на эмуляторе ОС *Android*. Для этого необходимо установить с официального сайта [16] базовый пакет инструментов *SDK*. После корректной установки запустить файл *SDK Manager.exe*. Далее необходимо проделать несложные манипуляции, чтобы запустить виртуальное устройство.

После того как устройство загрузится, можно приступить к установке программы. Файл разработанного приложения называется *RealityNavigator.apk*. После установки запуск приложения можно осуществить, выбрав иконку приложения из списка установленных приложений. Приложение «AR Навигатор» представляет из себя информационную систему указания пути с применением дополненной реальности. Перед запуском необходимо, чтобы устройство было подключено к сети интернет и *GPS*, для прорисовки карты и получения данных о местоположении устройства. Если возникла проблема с картой и она не отображается, то выведется сообщение «Ошибка при создании карты». При запуске приложения пользователь попадает на главный экран (рис. 2, а). На данном экране расположена встроенная *Google* карта, а также кнопки для управления приложением. При нажатии на кнопку «*My Location*» в верхнем правом углу идет загрузка данных о местоположении пользователя, камера центрирует карту по текущему местоположению. Если не удалось загрузить данные, то отобразится местоположение последней локации пользователя. На карте местоположение отображается в виде маленькой синей точки (рис. 2, б).

В нижней части экрана находятся 3 кнопки необходимые для использования приложения, а именно: построить маршрут, удалить маршрут, перейти в режим дополненной реальности. При нажатии на кнопку с картой «Построить маршрут», на карту добавляется 5 маркеров неподалеку от локации пользователя, соединенные между собой линией, они имеют нумерацию и их можно перемещать (рис. 2, в) если маршрут уже был добавлен, на экран выведется сообщение «Маршрут уже построен!». К каждому маркеру привязывается указатель в дополненной реальности. Если локация пользователя не была определена, на экран выведется сообщение «Подождите пока определится ваше местоположение». При нажатии на кнопку с изображением ластика «Удалить маршрут» происходит проверка условия, если на карте есть соединенные между собой маркеры, то она удаляются вместе с соединяющей их линией,

если на карте ничего нет, то выводится сообщение «Нельзя удалить маршрут».



Рисунок 2 – Главный экран приложения «AR Навигатор» (а); отображение местоположения на карте (б); форма с построением маршрута (в)

При нажатии на кнопку с изображением камеры «Перейти в режим дополненной реальности» осуществляется переход на новое окно, на котором размещена камера (рис. 3, а). Перед нажатием необходимо, чтобы уже был построен маршрут, если его

нет, на экран выведется сообщение «Проложите маршрут!». При переходе в «Режим дополненной реальности» на экране появляется круг со стрелкой, который показывает направление к первой из пяти точек маршрута. Внизу экрана есть информация о том сколько пользователь прошел контрольных точек, дистанция до следующей точки и кнопка для перехода на экран с картой. Когда пользователь приходит к точке назначения на экране появляется изображение метки в дополненной реальности (рис. 3, б).

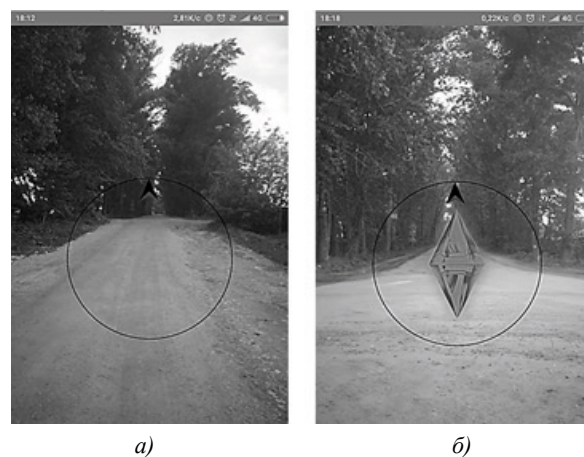


Рисунок 3 – Режим дополненной реальности (а);
Отображение AR объекта (б)

После отображения объекта дополненной реальности, информация внизу экрана обновляется и прокладывается маршрут к следующей точке, после этого AR объект остается на экране некоторое время, после чего пропадает. Когда пользователь дойдет до последней точки, работа программы выполнится и произойдет автоматический переход на экран с картой, где можно будет построить новый маршрут.

Основными входными данными для программы являются данные, получаемые через интернет и GPS, а также показания, собранные с датчиков. Для того чтобы работать с предлагаемой программой, необходимо обладать базовыми возможностями работы с сотовыми телефонами, у которых имеется сенсорный дисплей. Все взаимодействие с устройством происходит при помощи сенсорного дисплея.

Используемые подходы при разработке приложения убирают ряд действий, необходимых для технического обслуживания приложения. Но для поддержания стабильной работы необходимо периодически очищать оперативную и постоянную память устройства, проверять работу и доступность подключения к сети интернет и GPS. Программа была разработана и проверена на устройстве Xiaomi Redmi 4.

Заключение. По итогам проведенных исследований предложена система, реализующая определенные в цели функции. Программа позволяет с помощью мобильного устройства быстро и эффективно построить маршрут с применением дополненной реальности.

К достоинствам системы можно отнести использование перспективных мобильных технологий,

повсеместно набирающих популярность, лаконичность системы, простота их использования на обычном мобильном телефоне, интуитивно понятный пользовательский интерфейс.

К особенностям системы можно отнести установленную на мобильном устройстве ОС *Android*. Не каждый мобильный телефон работает под управлением данной ОС, но именно эта платформа в данный момент является самой популярной. Плюсом данной ОС является и то, что выпускаются телефоны разной ценовой категории под ее управлением.

СПИСОК ЛИТЕРАТУРЫ:

1. Цветков В.Я. Дополненная реальность // Международный журнал прикладных и фундаментальных исследований. – 2017. – № 6-2. – С. 211-212.
2. Яковлев Б.С., Пустов С.И. Классификация и перспективные направления использования технологии дополненной реальности // Известия Тульского государственного университета. Технические науки. – 2013. – № 3. – С. 484-492.
3. Android [Электронный ресурс] – Android | Платформа, расширяющая возможности. – Режим доступа: https://www.android.com/intl/ru_ru/ свободный. Яз. русский. (дата обращения: 30.07.2020).
4. Azuma R. T. (1997) A Survey of Augmented Reality // Presence: Teleoperators and Virtual Environments. Vol. 6, № 4. PP. 355–385.
5. Бойченко И.В., Лежанкин А.В. Дополненная реальность: состояние, проблемы и пути решения // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2010. – № 1-2 (21). – С. 161-165.
6. Иванова А. Технологии виртуальной и дополненной реальности: возможности и препятствия применения // Стратегические решения и риск-менеджмент. – 2018. – № 3 (108) – С. 88-107.
7. Егоров А.А. Одна из современных проблем дополненной реальности в портативных устройствах // Молодежь XXI века: образование, наука, инновации: Материалы VIII Всероссийской студенческой научно-практической конференции с международным участием. Под редакцией Т.А. Бирюковой. 2019. – С. 13-14.
8. Кураев Н.И. Цифровизация и приложения дополненной реальности // Молодежь XXI века: образование, наука, инновации: Материалы VIII Всероссийской студенческой научно-практической конференции с международным участием. Под редакцией Т.А. Бирюковой. 2019. – С. 15-16.
9. Чиркин А.Н., Пимонов А.Г. Представление возможностей компании с использованием приложения дополненной реальности // Инновации в информационных технологиях, машиностроении и автотранспорте Сборник материалов III Международной научно-практической конференции. Редколлегия: Д.М. Дубинкин [и др.]. 2019. – С. 101-102.
10. Honcharova N. Technology of Augmented Reality in Textbooks of New Generation // Проблемы сучасного підручника. – 2019. – № 22. – С. 46-56.
11. Павлова К.Т., Фалеева Е.В., Павлов Н.Г. Особенности способов взаимодействия пользователя и интерфейсов дополненной реальности // Фундаментальные и прикладные научные исследования: актуальные вопросы, достижения и инновации сборник статей XXXI Международной научно-практической конференции. 2020. – С. 92-94.
12. Голощапов А. Л. Google Android: программирование для мобильных устройств. БХВ-Петербург, 2011. – 804 с.
13. Майер Р. Android 2. Программирование приложений для планшетных компьютеров и смартфонов. Эксмо, 2011. – 671 с.
14. Дейтел П., Дейтел Х., Дейтел Э., Морган М. Android для программистов. Создаем приложения. Питер, 2012. – 560 с.
15. Сайт А. Климова [Электронный ресурс] – Android: собираем показания датчиков – Режим доступа: <http://developer.alexanderklimov.ru/android/sensors.php> свободный. Яз. русский. (дата обращения: 30.07.2020)
16. Google Maps API [Электронный ресурс] – Maps SDK for Android Utility Library | Google Developers – Режим доступа: <https://developers.google.com/maps/documentation/android-api/utility/?hl=ru> свободный. Яз. русский. (дата обращения: 30.07.2020).
17. Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Питер, 2016. – 432 с.
18. Джейсон Мак-Колм Смит. Элементарные шаблоны проектирования. Питер, 2013. – 533 с.
19. Wearesocial [Электронный ресурс] – Digital around the world in April 2020 - We Are Social. – Режим доступа: <https://wearesocial.com/blog/2020/04/digital-around-the-world-in-april-2020> свободный. Яз. русский. (дата обращения: 05.05.2020).
20. AndroidDeveloper [Электронный ресурс] – Android Studio features | Для разработчиков Android. – Режим доступа: <https://developer.android.com/studio/features.html> свободный. Яз. англ. (дата обращения: 30.07.2020).

Статья поступила в редакцию 28.07.2020

Статья принята к публикации 14.09.2020