МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

КАЛИНИНГРАДСКОЙ ОБЛАСТИ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СТРОИТЕЛЬСТВА»

(ГБУ КО ПОО «КИТиС»)

|  |  |  |
| --- | --- | --- |
| Курсовой проект допущен к защите  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата)  Заместитель директора по УМР Павленко Г. Я.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |  | Курсовой проект защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Пояснительная записка к курсовому проекту

по дисциплине: МДК 01.01 Разработка программных модулей

на тему: Разработка мобильного приложения под Android «Генератор идей»

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
| Выполнила: | студентка 3 курса,  группы ИСп 19-2к  Козьмина Екатерина Евгеньевна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель: | Большакова-Стрекалова Анна Викторовна, преподаватель ГБУ КО ПОО КИТиС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Калининград

2022

Содержание

Введение……………………………………………………………………….…..3

1 Описание предметной области…………………………….…………..….……4

1.1 Аналоги разрабатываемого приложения………………………………….4

* 1. Техническое задание…………………………………………...…….……..8
  2. Описание структуры приложения……………………………………........9

1. Описание разработки приложения…………………….………….…….......12

2.1 Обоснование средств разработки……………………...………….……...12

* 1. Разработка интерфейса…………………………………………..…..……15
  2. Разработка логики работы приложения. Схема взаимодействия компонентов проекта…………………………………………….…….….17
  3. Описание переменных, компонентов, классов и подпрограмм……...…19

1. Тестирование и установка приложения……………………………………..21

Заключение……………………………………………………………….………25

Список использованной литературы……………………………………………26

Приложение А. Код программы…………………………………………......….27

Введение

Телефоны являются компактными компьютерами, которые удобно брать с собой. В современном мире без телефона и специализированных приложений не обходится практически ни один человек.

Для начала нужно разобраться что такое мобильное приложение и для чего оно нужно.

Мобильное приложение – программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы (Android, Windows Phone, IOS и т. д.).

Назначение приложений может быть разным: сервисы, игры, магазины, развлечения, онлайн-помощники и другое.

Разработка мобильных приложений – это процесс, основанный на разработке программного обеспечения.

Цель нашего курсового проекта – разработать программный продукт, который будет функционировать на внешнем устройстве.

Наша основная задача – это создать приложение развлекательного характера. Для начала нужно найти и проанализировать похожие приложения. Выбрать среду разработки, удобный для нас язык программирования.

1 Описание предметной области

Наше приложение будет относиться к категории развлечений. Функциональное составляющее выглядит следующим образом:

1. Кнопка.
2. Область, для показа какого-либо текста.

При нажатии на кнопку будет показана фраза (идея) из возможных 50.

Данная программа будет полезна для художников, которые ищут новые идеи для рисования.

* 1. Аналоги разрабатываемого приложения

Рассмотрим аналоги разрабатываемого приложения. Если в Google Play задать запрос «Идеи для рисования», то самыми первыми можно увидеть такие приложения как: «Virink Что Нарисовать» (Рисунок 1) и «Идеи для рисования» (Рисунок 2). На момент запроса одно из них довольно популярно (100 тыс.+ скачиваний), а второе, наоборот, не сильно популярно (1 тыс.+ скачиваний). [1]

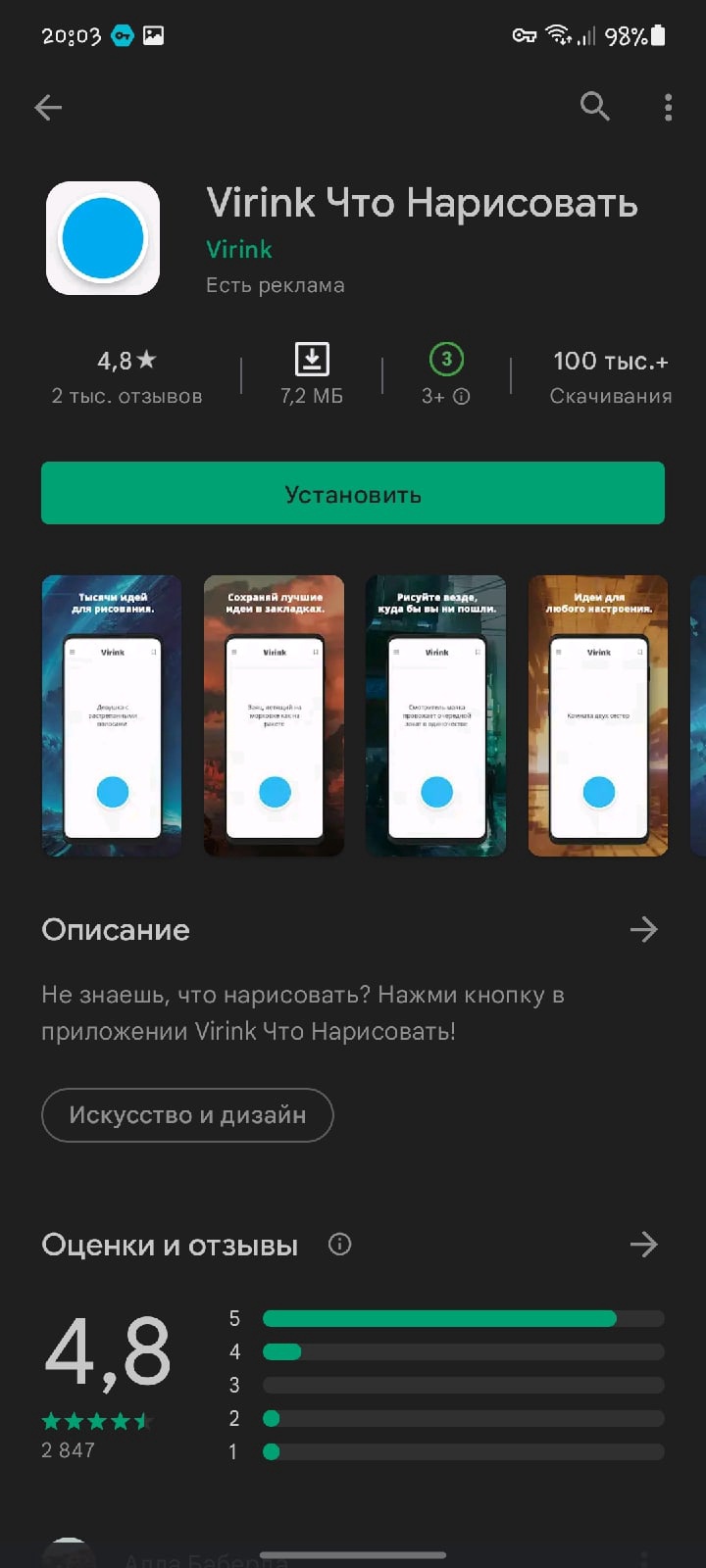


Рисунок 1 – Приложение «Virink Что Нарисовать»

Для удобства анализа можно скачать данные приложения. При открытии приложений, как можно увидеть, появляется активность. Наблюдаем кнопку, при нажатии на которую появляется текст.

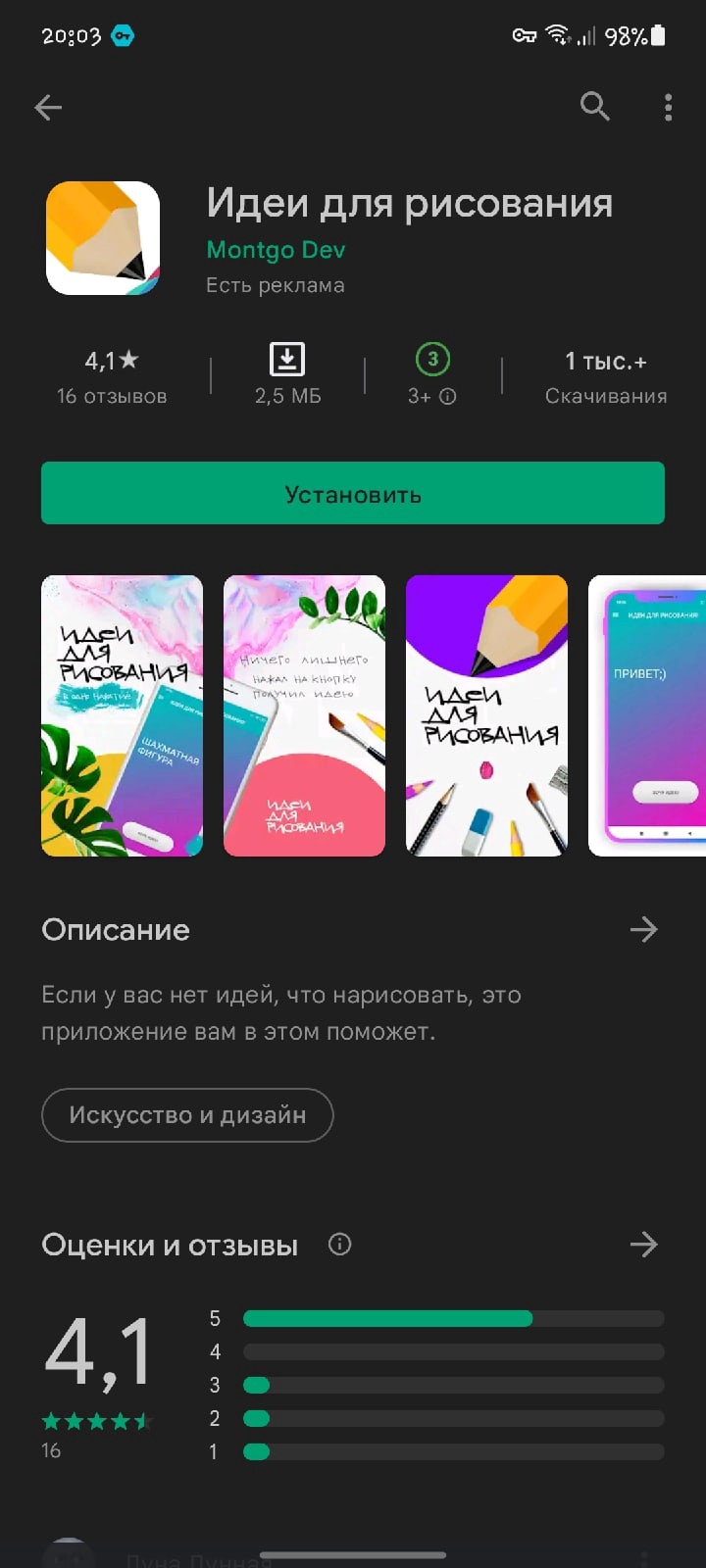


Рисунок 2 – Приложение «Идеи для рисования»

В верхнем левом углу каждого из приложений есть меню. Меню первого приложения содержит (Рисунок 3):

1. Закладки (понравившиеся пользователю идеи).
2. Смену темы приложения (темная/светлая).
3. Смену языка.
4. Ссылку на приложение, чтобы делится им с друзьями.
5. Функцию для предложения своих идей.
6. Стандартную информацию о конфиденциальности и о самом приложении.

Меню второго приложения содержит (Рисунок 4):

1. Информацию о приложении.
2. Ссылку на сайт с идеями.
3. Оценку приложения.
4. Информация о том, как пользоваться приложением.
5. Функцию для предложения своих идей.
6. Ссылку на приложение, чтобы делится им с друзьями.
7. Кнопку выхода из приложения.

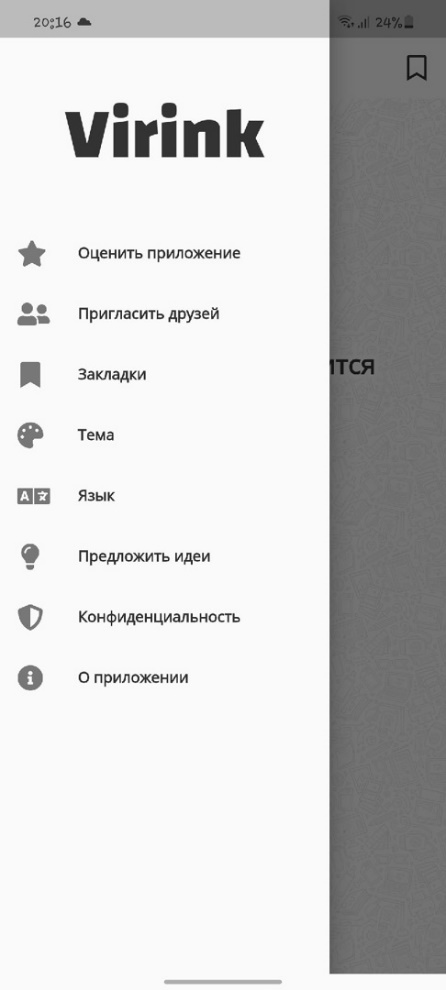


Рисунок 3 – Меню приложения «Virink Что Нарисовать»

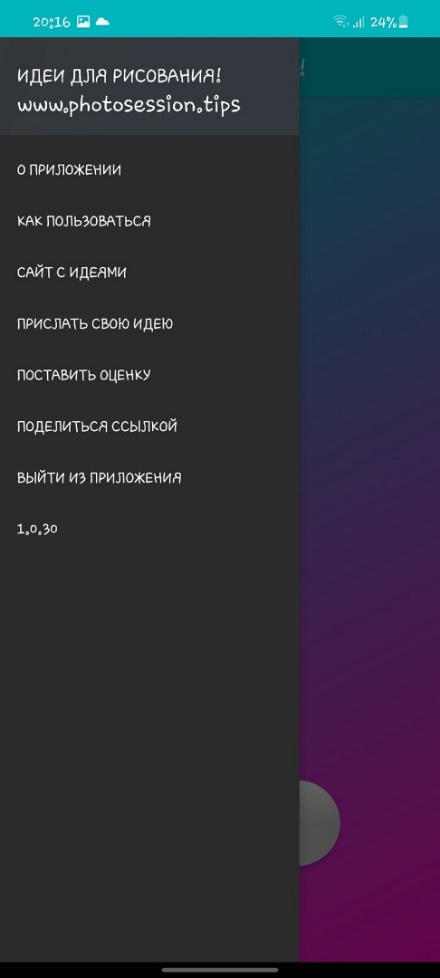


Рисунок 4 – Меню приложения «Идеи для рисования»

Многие из этих функций нашему приложению будут не нужны, т. к. запускать его в массовое пользование мы не будем. Однако данные функции могут быть дополнением (улучшением), например, при обновлениях приложения, если им будут активно пользоваться.

Интерфейс главной активности каждого из приложений, которые мы рассматриваем представлен на рисунке 5 (первое приложение «Virink Что Нарисовать») и на рисунке 6 (второе приложение «Идеи для рисования»).

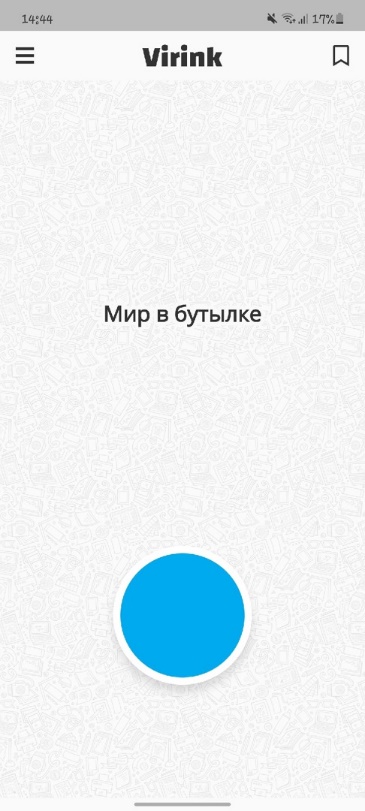


Рисунок 5 – Интерфейс приложения «Virink Что Нарисовать»

Вывод: Изучены мобильные приложения похожего характера. Как можно увидеть, интерфейсы приложений просты и лаконичны, чем завлекают потенциальных пользователей. В дальнейшем мы будем использовать это в разработке интерфейса нашего программного продукта.

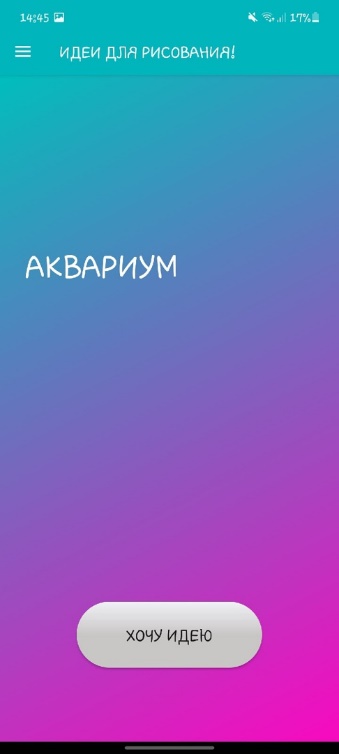


Рисунок 6 - Интерфейс приложения «Идеи для рисования»

* 1. Техническое задание

Мы разрабатываем наше приложение на основании: Приказ 238/1У от 01.09.2021.

Наименование нашей программы – «Генератор идей». Приложение разрабатывается под операционную систему Android. Область применения:

1. Развлекательная деятельность.
2. Художественная (творческая) деятельность.

В нашем приложении необходимо разработать кнопку, при нажатии которой будет показываться одна идея из предложенных 50. Реализовать вывода текста мы будем с помощью пакета java.util.random. Он используется для генерации случайных чисел, однако вместо чисел у нас будет использоваться текст, т. к. пакет позволяет генерацию и других случайно выбранных значений (например, логических, булево). [2]

Количество идей можно соответственно сделать больше, но для начальной версии хватит и этого количества.

Любой пользователь сначала открывает приложение на устройстве путем нажатия на иконку приложения. При открытии можно увидеть активность. Пользователь изучает ее для дальнейшего понимания работы программы, либо он изучил еще на этапе установки приложения. Далее идет нажатие на кнопку. Появляется текст, который человек, пользующийся программой, читает и анализирует. Если пользователю не подходит идея, появившаяся на экране, он повторно нажимает на кнопку и цикл повторяется. После того, как человеку понравилась идея, он выходит из приложения и приступает к ее реализации.

У всех приложений есть технические требования к устройствам. Если технические требования не будут соответствовать рекомендациям разработчиков, то программный продукт может некорректно работать на устройстве или даже вовсе вывести мобильный телефон из рабочего состояния.

Мы возьмем средние технические требования, которые актуальны в данный момент практически на любом современном устройстве. Для нашей программы не нужны очень «мощные» устройства, так как она довольно проста.

Технические требования к мобильным устройствам для нашего разрабатываемого приложения:

1. Операционная система Android 6.1 и выше.
2. Размер ОЗУ – от 2 ГБ.
3. Размер встроенной памяти от 32 ГБ.
4. Емкость аккумуляторной батареи от 3000 мАч.

1.3 Описание структуры приложения

Для понимания структуры и процесса программного продукта, построим схему и UML-диаграмму. Это упростит дальнейшую разработку приложения Созданные схемы обычно служат подсказкой для заказчиков, разработчиков и пользователей продуктом. [3]

Сформируем UML-диаграмму процесса приложения на базе нашего технического задания. Диаграмма представлена на рисунке 7.

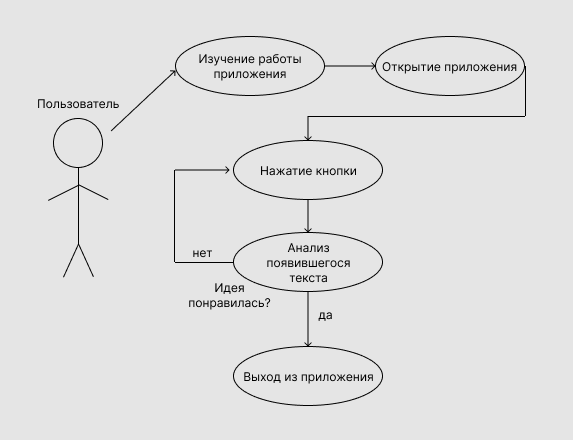


Рисунок 7 – UML диаграмма «Схема процесса»

Для более простого понимания структуры построим схему. Это может быть структурная и/или функциональная схема.

Ниже представлена наша созданная схема, представленная на рисунке 8. Она очень проста в понимании и удобна для изучения структуры проекта.

Структура создаваемого приложения может быть дополнена, если сделать «Главное Меню» для нашей программы. Однако мы меню делать не будем, потому что запуск приложения в общее пользование не будет реализован.

Вход в приложение

Кнопка «НАЖМИ»

Вывод идеи на экран

Выход из приложения

Рисунок 8 – Схема «Структура приложения»

Вывод: Разработана схема структуры мобильного приложения. Построена UML-диаграмма с показанным процессом работы приложения.

1. Описание разработки приложения

Рассмотрим несколько сред разработки приложений, чтобы определить оптимальный вариант для создания нашей программы.

Ознакомимся и проанализируем такие среды, как:

1. Android Studio.
2. IntelliJ Idea.
3. Eclipse.
   1. Обоснование средств разработки

Android Studio [4]

Android Studio – универсальная среда разработки, позволяет оптимизировать работу приложения на смартфонах, на планшетах, портативных ПК.

Среда в основном предназначена для разработки приложений на Android.

Во время создания приложений для операционной системы Android, пользователь программного обеспечения может наблюдать за изменениями в проекте в режиме реального времени.

В программу встроен эмулятор, позволяющий проверить корректную работу приложения на устройствах с разными экранами, с различными соотношениями сторон.

Достоинства:

1. Среда разработки поддерживает работу с несколькими языками программирования.
2. Редактор кода, с которым удобно работать.
3. Большая библиотека с готовыми шаблонами и компонентами.
4. Большой набор средств и инструментов.

Недостатки: для запуска необходим производительный ПК и невозможность написать серверные проекты.

IntelliJ Idea [5]

Данную программу разработала российская компания JetBrains. Эта среда позволяет создавать приложения и программы на нескольких языках программирования, в частности Java, JavaScript, Python. Также IntelliJ Idea не перегружает ПК.

Каждый компонент IntelliJ Idea создан для того, чтобы максимально повысить продуктивность разработки. Умный редактор кода в сочетании с дизайном делают работу эффективной. На основе этой среды основана среда разработки Android Studio.

Достоинства среды разработки:

1. Более оперативная отладка значений.
2. Предусмотрен автозаполнитель методов.
3. Есть переработка кода.
4. Интерфейс более понятный и лаконичный.
5. Подходит для тех, кто программирует на Java.
6. Все необходимые инструменты под рукой.

Недостатком является то, что данная среда разработки платная.

Eclipse [6]

Eclipse – Свободная интегрированная среда разработки приложений. Это бесплатная IDE, разработанная некоммерческой компанией Eclipse Foundation. Программа является базой, которая регулирует процессы создания приложений.

Преимущества Эклипс:

1. Интерфейс переведен на грамотный русский язык (документация прилагается).
2. Отлично работает на компьютерах с низкой производительностью.
3. Имеет дополнительные функции (для серверной работы и анализа базы данных).
4. Может подключаться к модулям.
5. Может работать в групповом режиме (когда проект создают несколько человек одновременно).

Скачать Android Studio можно с официального сайта. Студия подходит для разных операционных систем компьютеров, таких как Windows, Mac, Linux. Мы скачиваем пакет для Windows, 64-разрядной версии. Установка происходит также, как и других приложений компьютера (рисунок 9).

Так как наша программа разрабатываться будет для операционной системы Android, то наилучшим вариантом для нас будет Android Studio. Среда разработки бесплатная, с понятным интерфейсом. Для нашей программы нам не нужна работа с серверами. Оптимальным языком программирования будет Java [7], т. к. он понятен, доступен в изучении и является предпочтительным для разработки мобильных приложений. Возможность следить за результатами разработки на эмуляторе также для нас очень полезна.

Вывод: Мы сделали анализ и в результате определились с языком программирования и средой разработки. Выбранная среда разработки – Android Studio. Выбранный язык программирования – Java.

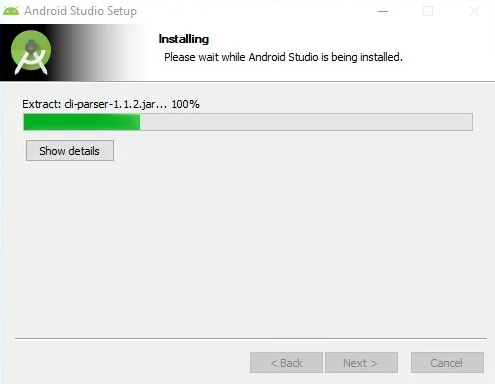


Рисунок 9 – Установка Android Studio

* 1. Разработка интерфейса

Создание интерфейса очень важно в разработке любого программного продукта. Пользователь зачастую сразу обращает внимание на внешнее составляющее того, чем ему предстоит пользоваться.

Так же еще одна задача интерфейса – заинтересовать потребителя, чтобы он использовал приложение на постоянной основе и пользовался программой длительное количество времени.

Интерфейс должен быть понятен пользователю. Так как приложение не является игрой, то лучше всего сделать простой дизайн, который будет приятен глазу и не надоест спустя время.

В результате долгого использования у пользователя не должно появляться недомогания, головокружения и других симптомов плохого самочувствия.

Для создания интерфейса мы будем использовать сайт Figma [8]. Он используется для разработки интерфейсов (дизайнов) сайтов и приложений.

Сначала зарегистрируемся либо войдем в действующую учетную запись. Создадим новый проект. Добавим все нужные нам компоненты.

В целях нашего удобства мы подпишем все элементы. Впоследствии будет удобная возможность сверяться с нашим планом построения приложения.

У нас получился разработанный интерфейс нашего приложения, приведенный на рисунке 10. Далее мы будем придерживаться его во время разработки проекта в Android Studio, на эмуляторе, а в дальнейшем проверять интерфейс на реальном устройстве.

Разработанный интерфейс пользователя доступен и понятен для любого человека, который будет пользоваться данным приложением.

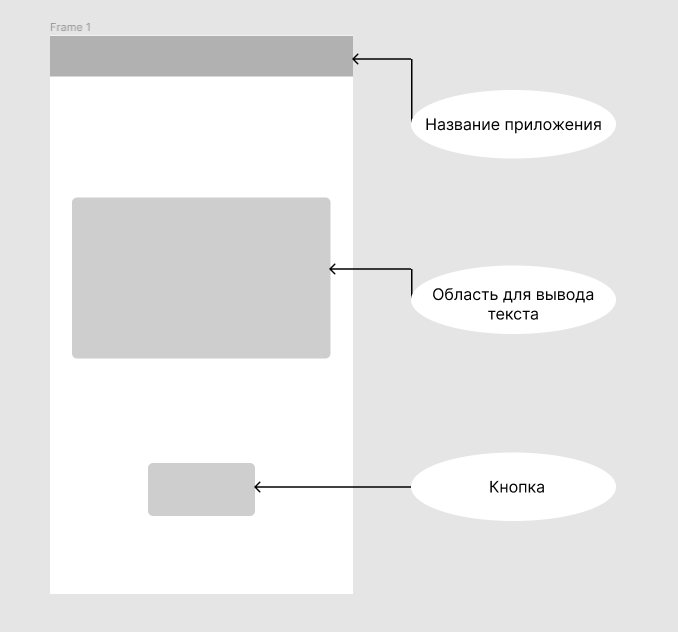


Рисунок 10 – Интерфейс будущего приложения

* 1. Разработка логики работы приложения. Схема взаимодействия компонентов проекта

Наше приложение мы назовем «Drawing Ideas». Если перевести это название на русский язык, то получится «Идеи Рисования».[9]

В Android Studio создаем новый проект. В нем добавляем одну активность, с которой в дальнейшем и будем работать.

Разберем основные компоненты нашего кода в таблице 1 и создадим схему компонентов проекта.

Activity (активность/деятельность) является визуальным интерфейсом пользователя. Данное окно зачастую заполняет весь экран мобильного телефона. Также может быть меньше по размеру, чем экран. Может быть несколько активностей и окон в одном приложении. При открытии следующей активности работа предыдущей деятельности останавливается.

Мы добавляем область TextView и кнопку Button на активность, это представлено на рисунке 11. Они автоматически уже будут в коде нашей активности activity\_main.hml. Мы выставляем им нужный размер. Нужное расположение необходимо зафиксировать, чтобы наши компоненты при запуске на эмуляторе не сместились со своего места.

Таблица 1 – Компоненты приложения

|  |  |
| --- | --- |
| Компонент проекта | Для чего нужен компонент |
| Activity | Визуальный интерфейс пользователя |
| TextView | Отображение текста |
| Button/ImageButton | Кнопка с текстом/изображением |

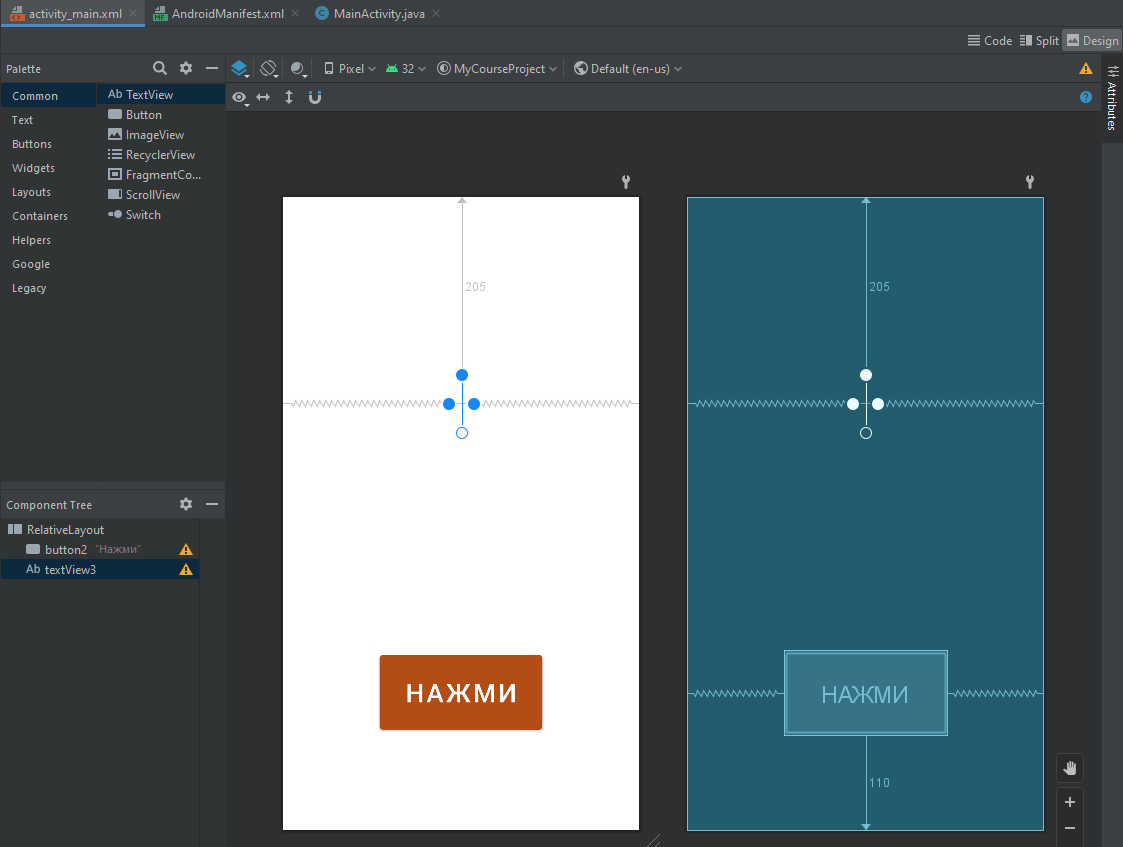


Рисунок 11 – Добавление компонентов на активность

* 1. Описание переменных, компонентов, классов и подпрограмм

Код приложения представлен в Приложении А. Код программы.

Название компонентов, классов, переменных и т. д., их основное описание и дополнительная информация представлены в таблице 2. [10]

Таблица 2 – Компоненты, классы, переменные и подпрограммы

| № | Название компонента, класса, переменной, подпрограммы | Основное описание | Дополнительная информация |
| --- | --- | --- | --- |
| 1 | TextView | Один из самых используемых компонентов в любом приложении. С помощью данного компонента пользователю удобно ориентироваться в программе. Для отображения текста в TextView в файле разметки используется атрибут android:text | Тексту данного компонента можно задать размер (пиксели, дюймы, миллиметры и т.д.), стиль, цвет, выставить какой-либо фон |
| 2 | Button | Кнопка. Наследуется от TextView и является базовым классом для класса СompoundButton. От класса CompoundButton в свою очередь наследуются такие элементы как CheckBox, ToggleButton и RadioButton | Т. к. кнопка является наследником TextView, то использует многие знакомые атрибуты: textColor, textSize и др. Есть еще один компонент ImageButton, у него вместо текста используется изображение (рисунки/картинки). Управлять размером шрифта, цветом текста и прочими свойствами у кнопки тоже можно |
| 3 | Java.util.Random | Пакет, содержащий классы и интерфейсы, для генерации случайных чисел, а также других видов случайно выбранных значений (например, булево) |  |
| 4 | String | Класс в Java, предназначенный для работы со строками. Его | Операции над объектом приведут к созданию нового объекта. Объекты |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название компонента, класса, переменной, подпрограммы | Основное описание | Дополнительная информация |
|  |  | объекты не могут быть изменены после создания | этого класса могут быть использованы в многопоточной среде |
| 5 | String [] items {} | Выпадающий список JComboBox | Бывает редактируемый и не редактируемый |
| 6 | @Override | Относится к маркерным аннотациям и указывает, что метод переопределяет/реализует унаследованный метод | Помогает уменьшить количество ошибок, поскольку при такой аннотации компилятор должен генерировать сообщение об ошибке, если не выполняется одно из двух условий |
| 7 | Protected void | Модификатор доступа. Поля и методы, обозначенные модификатором доступа, будут видны:  в пределах всех классов, находящихся в том же пакете, что и наш;  в пределах всех классов-наследников нашего класса |  |
| 8 | OnCreate(Bundle savedInstanceState) | Защищенный метод. Задаёт начальную установку параметров при инициализации активности | Перед реализацией метода необходимо сначала вызывать версию этого метода из суперкласса super |
| 9 | setContentView | Метод, устанавливающий содержимое Activity из layout-файла | В качестве аргумента мы указываем не путь к layout-файлу, а константу, которая является ID файла |
| 10 | relativelayout | Мощная служебная программа для разработки пользовательского интерфейса | Позволяет дочерним компонентам определять свою позицию относительно родительского компонента или относительно соседних дочерних элементов |
| 11 | public void | Публичный модификатор доступа public. Методы или поля с этим модификатором общедоступны, видимы другим классам из текущего пакета и из внешних пакетов | Это самый широкий уровень доступа |

3 Тестирование и установка приложения

После создания приложения, а также всех компонентов, тестируем получившийся программный продукт на эмуляторе или устройстве, которое можем использовать в качестве эмулятора.

Мы будем использовать физическое устройство. Благодаря этому компьютер не перегружается. Также не нужно будет ждать долгой загрузки виртуального эмулятора.

После открытия приложения на физическом устройстве в качестве эмулятора изначально область для вывода текста пустая. Это можно увидеть на рисунке 12, приведенном ниже.

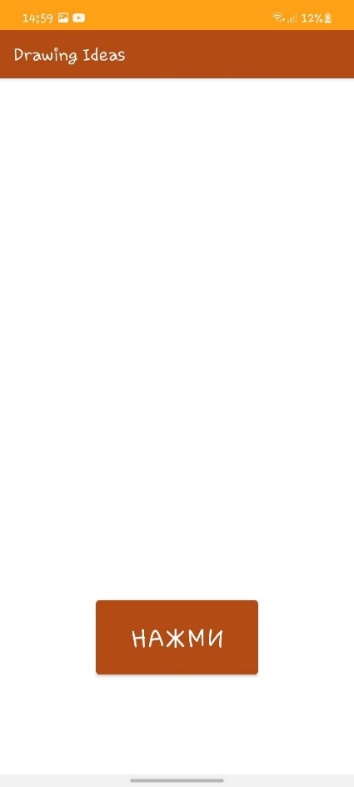


Рисунок 12 – Открытое приложение

При смене темы всего телефона на «темную» наше разработанное приложение тоже меняет тему. Это можно увидеть на рисунке 13. Еще при создании нового проекта в Android Studio, эта функция уже автоматически встроена в нашей программе.



Рисунок 13 – Темная тема приложения

При тестировании приложения была выявлена незначительная ошибка. Она заключается в том, что из-за ручного ввода текста, который предоставляется пользователю после нажатия кнопки, в некоторых фразах имеются грамматические ошибки. Одна из таких фраз представлена на рисунке 14. Ошибка в слове «карамельного». Пропущена одна буква.

Решение проблемы: путем исправления данного слова в коде нашего приложения и повторного запуска на эмуляторе. Таких фраз с ошибкой было несколько, т.к. ошибки похожи, рассматривать мы их не будем. Они решаются тем же способом.

Следующая ошибка, выявленная в результате тестирования, это неправильный перенос слов. Как можно заметить на рисунке 15, в фразе «Нарисуй акулу-журналиста», из-за отсутствия пробелов между словами и тире, слово перенеслось на следующую строчку неправильно.

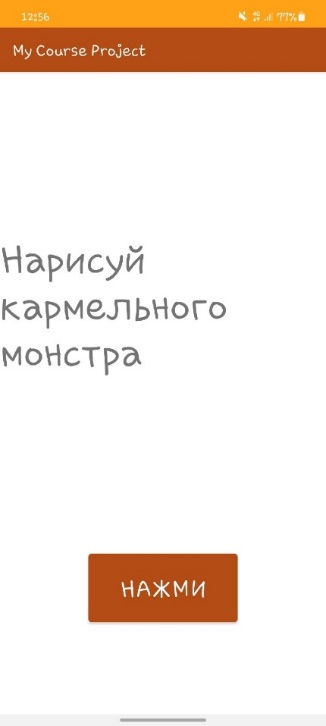


Рисунок 14 – Грамматическая ошибка текста

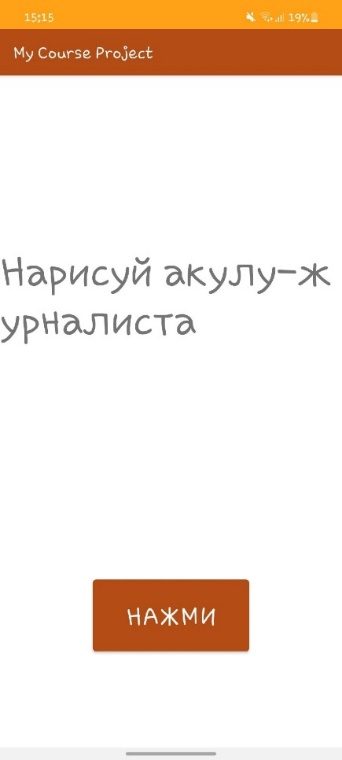


Рисунок 15 – Ошибка переноса текста

Решение: исправить эту проблему можно, добавив пробелы в коде и перезапустить приложение на эмуляторе. Все похожие фразы с тире и отсутствием пробелов можно решить тем же способом, поэтому рассматривать мы их не будем. Больше ошибок выявлено не было.

В результате тестирования текст оказался слишком маленьким (рисунок 16) и был увеличен для более удобного прочтения.

Также размер и текст кнопки был увеличен для того, чтобы было удобнее нажимать на кнопку.

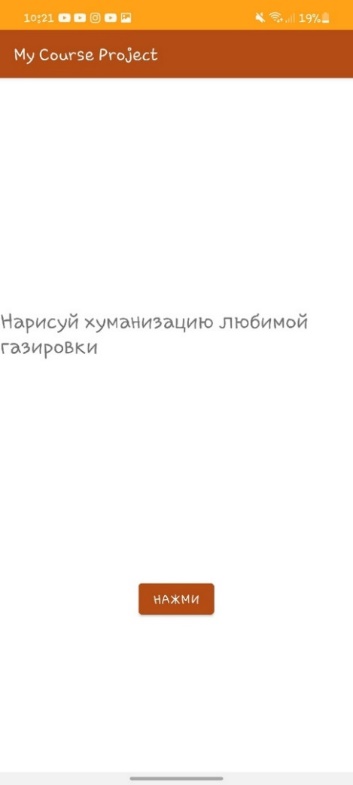


Рисунок 16 – Маленький размер текста и кнопки

Установить готовое приложение можно через APK файл либо запустить приложение на устройстве в качестве эмулятора, тогда Android Studio автоматически установит наше приложение на устройство и запустит его.

Заключение

Подводя итоги курсового проекта, в рамках которого было разработано приложение, можно сделать следующие выводы.

Выполнены цели и задачи, которые были поставлены. Работая в среде разработки Android Studio. Сравнив несколько сред разработки и выбрав наиболее подходящую, установили ее. Разработано приложение под Android «Генератор идей». Приложение создано для художников, которые ищут новые идеи для своих рисунков. Сделан анализ похожих приложений конкурентов и на его основании определена структура приложения.

Программный продукт функционирует на внешних устройствах. В результате тестирования приложения были выявлены и исправлены возникшие ошибки.

В курсовом проекте показан процесс разработки мобильного приложения, а точнее структуры и интерфейса.

Список использованной литературы

1. Блинов, И.Н. Java. Методы программирования: уч.-мет. пособие / И.Н. Блинов, В.С. Романчик. – Минск: издательство «Четыре Четверти» 2013. – 896с.
2. Шилдт, Герберт Java: руководство для начинающих, 7-е изд. Пер. с англ. – СПб.: ООО «Диалектика», 2019. – 816 с.
3. Филлипс Б. Android. Программирование для профессионалов. 3-е изд. / Филлипс Б., Стюарт К., Марсикано К. – СПб.: Питер, 2017. – 688 с.
4. Android Studio и SDK | Разработчики Android: [Электронный ресурс]. URL: https://developer.android.com/studio
5. IntelliJ IDEA: функциональная и эргономическая IDE для разработки на Java от JetBrains: [Электронный ресурс]. URL: https://www.jetbrains.com/idea/?ysclid=l2syc6r4og
6. Eclipse Среда Разработки: [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Eclipse\_(среда\_разработки)
7. Язык программирования Java: [Электроннный ресурс]. URL: https://metanit.com/java/tutorial/?
8. Figma: инструмент для совместного проектирования интерфейса: [Электронный ресурс]. URL: https://www.figma.com/?fuid=
9. Гриффитс Д. Head First. Программирование для Android / Гриффитс Дон, Гриффитс Дэвид – СПб.: Питер, 2016. – 704 с.
10. Дейтел П. Android для разработчиков. 3-е изд. / Дейтел П., Дейтел Х., Уолд А. – СПб.: Питер. – 2016. – 512 с.

Приложение А. Код программы

MainActivity.java

package com.example.mycourseproject;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

Button button;

TextView textView;

Random r;

String[] items = {"Нарисуй человека с аллергией на солнце",

"Нарисуй хуманизацию своей зубной щетки",

"Нарисуй хуманизацию оливки",

"Нарисуй образ самого запомнившегося тебе прохожего",

"Нарисуй хуманизацию браузера, которым пользуешься",

"Нарисуй персонажей фильма в своем стиле",

"Если ты диджитал художник, нарисуй на бумаге",

"Если ты рисуешь на бумаге, попробуй порисовать на телефоне/планшете",

"Нарисуй хуманизацию любимой газировки",

"Хуманизируй разные кофейные напитки",

"Нарисуй любимых персонажей из разных мультиков вместе",

"Нарисуй логотип своей жизни",

"Нарисуй лисенка в стиле минимализм",

"Нарисуй белочку в виде деревца",

"Нарисуй кита на воздушном шаре",

"Рисуй свою любимую одежду каждый день в течении недели",

"Нарисуй милого эльфа",

"Нарисуй пещеру, в которой растут светящиеся грибы",

"Нарисуй книжную фею",

"Нарисуй пейзаж города забытой цивилизации",

"Нарисуй стеклянный кинжал",

"Нарисуй дверь в параллельную вселенную",

"Нарисуй демона в сахарнице",

"Нарисуй монстра под кроватью",

"Нарисуй морского дракона",

"Нарисуй замок из листьев",

"Нарисуй акулу - журналиста",

"Нарисуй ледяного дракона",

"Нарисуй гитариста на крыше",

"Нарисуй бутербродный лес",

"Нарисуй торговый центр для улиток",

"Нарисуй насекомых из драгоценных камней",

"Нарисуй свое теплое воспоминание",

"Нарисуй магические предметы из Гарри Поттера",

"Нарисуй душу леса",

"Нарисуй ребенка в сказочном лесу",

"Нарисуй ребенка в жутком лесу",

"Нарисуй бога, заваривающего чай",

"Нарисуй повелителя мух",

"Нарисуй карамельного монстра",

"Нарисуй одежду из облаков",

"Нарисуй бугимэна",

"Нарисуй человека - мотылька",

"Нарисуй референсы людей",

"Нарисуй красивые для тебя вещи",

"Нарисуй анатомию и референсы кошки",

"Нарисуй офисное здание художников",

"Хуманизируй свою любимую музыку",

"Нарисуй своего любимого исполнителя",

"Нарисуй себя и своего питомца"};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

button = (Button) findViewById(R.id.button2);

textView = (TextView) findViewById(R.id.textView3);

r = new Random();

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {textView.setText(items[r.nextInt(items .length)]);

}

});

}

}

AndroidManifest.hml

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android=http://schemas.android.com/apk/res/android

package="com.example.mycourseproject">

<application

android:allowBackup="true"

android:icon="@drawable/ic\_palette\_24"

android:label="Drawing Ideas"

android:roundIcon="@drawable/ic\_launcher\_background"

android:supportsRtl="true"

android:theme="@style/Theme.MyCourseProject">

<activity

android:name=".MainActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

activity\_main.hml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
xmlns:app="http://schemas.android.com/apk/res-auto"  
xmlns:tools="http://schemas.android.com/tools"  
android:layout\_width="match\_parent"  
android:layout\_height="match\_parent"  
tools:context=".MainActivity">  
  
<Button  
android:id="@+id/button2"  
android:layout\_width="188dp"  
android:layout\_height="98dp"  
android:layout\_alignParentBottom="true"  
android:layout\_centerHorizontal="true"  
android:layout\_marginBottom="110dp"  
android:text="Нажми"  
android:textSize="30dp" />  
  
<TextView  
android:id="@+id/textView3"  
android:layout\_width="wrap\_content"  
android:layout\_height="wrap\_content"  
android:layout\_alignParentTop="true"  
android:layout\_centerHorizontal="true"  
android:layout\_marginTop="205dp"  
android:text=""  
android:textSize="50dp" />  
</RelativeLayout>