Описание Программы

Lo-Fi Shop

КОМАНДА АРДАМАНА

2021

СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc85797657)

[1 Постановка задачи 3](#_Toc85797658)

[1.1 Обозначение и наименование программы 3](#_Toc85797659)

[1.2 Программное обеспечение, необходимое для функционирования программы 3](#_Toc85797660)

[1.3 Языки программирования, на которых написана программа 3](#_Toc85797661)

[1.4 Функциональная модель 4](#_Toc85797662)

[1.5 Диаграмма Прецедентов 5](#_Toc85797663)

[1.6 Диаграмма последовательности 6](#_Toc85797664)

[1.7 Диаграмма Классов 6](#_Toc85797665)

[2 Функциональное назначение 7](#_Toc85797666)

[2.1 Назначение программы 7](#_Toc85797667)

[3 Описание логической структуры 8](#_Toc85797668)

[3.1 Структура программы 8](#_Toc85797669)

[3.2 Связи между составными частями программы 12](#_Toc85797670)

[4 Чеклист 13](#_Toc85797671)

[5 Текст программы 14](#_Toc85797672)

# АННОТАЦИЯ

В данном программном документе приведено описание программы «Lof-fi Shop» предназначенной для развлекательных целей.

Программа «Lo-fi Shop» написана на языке C# в среде программирования Visual Studio 2019, предназначена для работы в операционной системе Android.

# Постановка задачи

## Обозначение и наименование программы

Программа «Lo-fi Shop» представляет собой мобильную игру. Пользователю предоставляется возможность играть в роли продавца компьютерного магазина и выполнять заказы клиентов появляющихся на экране.

Программа «Lo-fi Shop» имеет следующие атрибуты:

- Наименование исполняемого файла - Lo\_fi\_Shop.Android

- Размер исполняемого файла - 117 Мб

- Версия файла - 1.0

- Название продукта - Lo-fi Shop

- Производитель - АРДАМАНА

- Язык - Русский

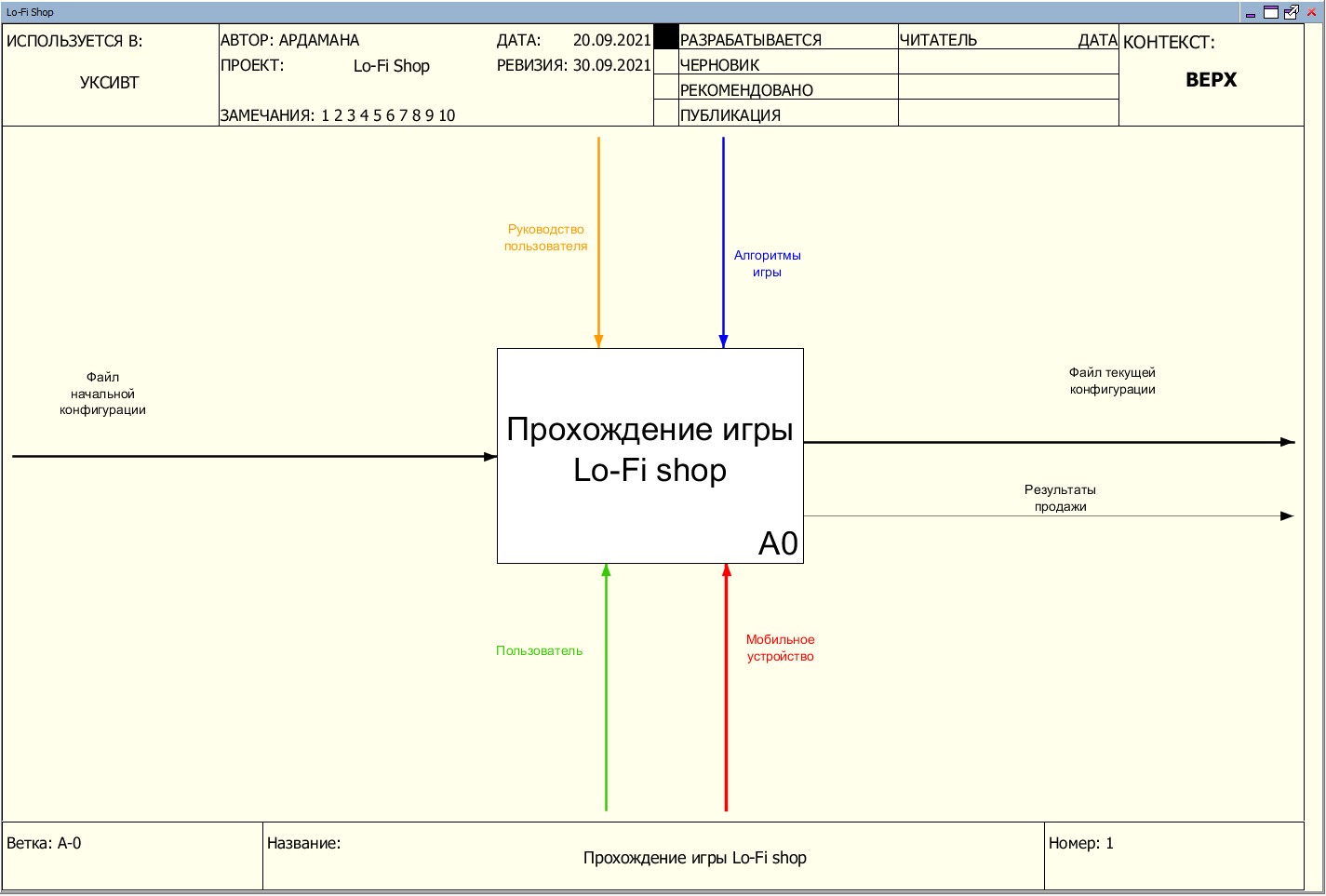
## Программное обеспечение, необходимое для функционирования программы

Программа работает под управлением ОС Android 7.0 и выше.

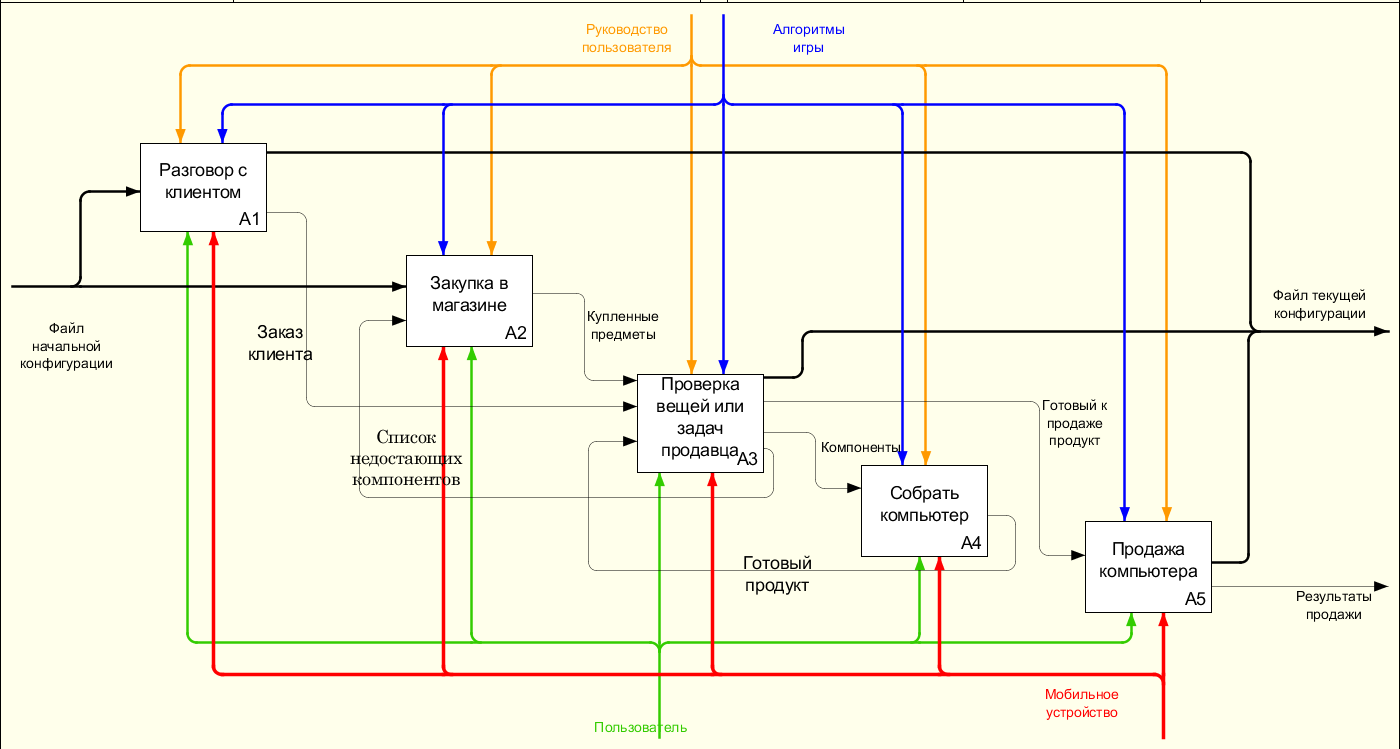
## Языки программирования, на которых написана программа

Исходным языком программирования является C#, среда разработки – Visual Studio 2019

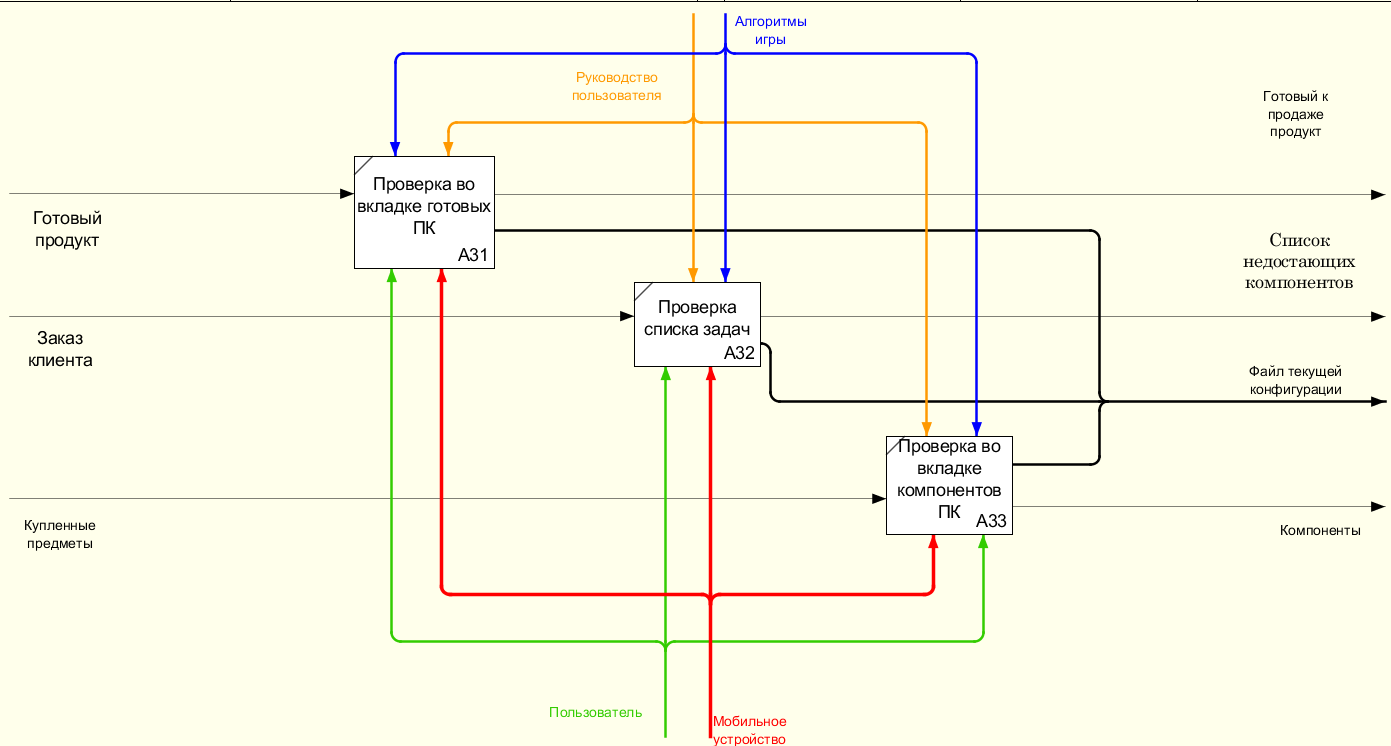
## Функциональная модель



1. Функциональная модель



1. Функциональная модель



1. Функциональная модель

## Диаграмма Прецедентов



1. Диаграмма Прецедентов

## Диаграмма последовательности



1. Диаграмма последовательности

## Диаграмма Классов



1. Диаграмма Классов

# Функциональное назначение

## Назначение программы

Программа «Lo-fi Shop» представляет собой игру, и предназначена для проведения досуга, вследствие чего предусматривается одна категория пользователей - игроки. В процессе работы приложения пользователь является непосредственным участник игрового процесса и оказывает непосредственное влияние на него.

# Описание логической структуры

## Структура программы

Таблица 3.1.1 – Перечень подпрограмм – функций программы Lo-fi Shop

|  |  |  |
| --- | --- | --- |
| Методы | Назначение | |
| 1 | 2 | |
| MainMenuPage.xaml.cs | | |
| public MainMenuPage | | Инициализация компонентов и загрузка главного меню |
| private void BtnHowToPlay\_Clicked | | Переход к экрану "Как Играть" |
| private void BtnPlay\_Clicked | | Переход к главному экрану игры |
| private void BtnSetting\_Clicked | | Переход к экрану "Настройки" |
| private void BtnAbout\_Clicked | | Переход к экрану "Об авторах" |
| private void BtnExit\_Clicked | | Выход из приложения |
| Item.cs | | |
| public Item(string Name, int Sell, string Description = "", bool InInvenrory = false) | | Конструктор класса Item |
| public static Item[] CreateItems | | Создание массива всех существующих предметов |
| PersonClass.cs | | |
| private PersonClass(int Money, int Exp, List<string> InventoryPath, List<string> InventoryWhole, List<int> Settings) | | Конструктор класса PersonClass |

Продолжение таблицы 3.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| public static string Read\_TXT | Чтение данных из файла |
| public static void Write\_TXT | Сохранения данных в файл |
| public static void Write\_TXT2 | Сохранения данных в файл |
| public static void Write\_TXT3 | Сохранения данных в файл |
| public static PersonClass OverwriteData() | Перезапись файла новыми данными |
| public static void First\_Write\_TXT() | Создание начальных данных при первом запуске игры |
| AboutAuthorsPage.xaml.cs | |
| public AboutAuthorsPage() | Инициализация компонентов и загрузка страницы «Об авторах» |
| CraftPage.xaml.cs | |
| public CraftPage | Инициализация компонентов и загрузка страницы «Крафт» |
| private void DisplayInvPath() | Обработчик события нажатия на кнопку |
| private void Sborka\_Clicked | Обработчик события нажатия на кнопку |
| private bool Proverka | Обработчик события нажатия на кнопку |
| private void Sborka\_Proc | Обработчик события нажатия на кнопку |

Продолжение таблицы 3.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| private void Sborka\_Video | Обработчик события нажатия на кнопку |
| private void Sborka\_Mat | Обработчик события нажатия на кнопку |
| private void Sborka\_OP | Обработчик события нажатия на кнопку |
| private void Sborka\_HDD | Обработчик события нажатия на кнопку |
| private void Sborka\_Pit | Обработчик события нажатия на кнопку |
| private void Sborka\_Kyler | Обработчик события нажатия на кнопку |
| private void Sborka\_Korpus | Обработчик события нажатия на кнопку |
| HowToPlayPage.xaml.cs | |
| public HowToPlayPage | Инициализация компонентов и загрузка страницы «Как играть» |
| public void Add\_Al() | Создание Скролл-списка |
| PlayPage.xaml.cs | |
| public PlayPage | Инициализация компонентов и загрузка страницы игры |

Продолжение таблицы 3.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| private void Get\_data() | Заполнение значений в игровом окне (деньги,опыт) |
| private void ImageShkaf\_Clicked | Открытие инвентаря |
| private void ImageTable\_Clicked | Переход к странице «Крафт» |
| private void ImageKassa\_Clicked | Переход к странице «Магазин» |
| private void TableOfQuest\_Open | Открытие доски задач |
| private bool OnTimerTick() | Таймер |
| private void Sky\_Clicked | Управление диалога с покупателем |
| private void ButtonYes\_Clicked | Управление диалога с покупателем |
| private void ButtonNo\_Clicked | Управление диалога с покупателем |
| private void SkyBuy\_Clicked | Управление диалога с покупателем |
| private void ButtonHide\_Clicked | Управление диалога с покупателем |
| QuestPage.xaml.cs | |
| public QuestPage | Инициализация компонентов и загрузка страницы «Доска заданий» |
| SettingsPage.xaml.cs | |
| public SettingsPage | Инициализация компонентов и загрузка страницы «Настройки» |
| ShopPage.xaml.cs | |
| public ShopPage | Инициализация компонентов и загрузка страницы «Магазин» |

Продолжение таблицы 3.1.1

|  |  |
| --- | --- |
| 1 | 2 |
| private void Item\_Clicked | Выбор предмета для покупки |
| private void EmpetyMessage\_Clicked | Закрытие сообщения о покупке |
| private void BuyBtn\_Clicked | Покупка предмета и перенос его в инвентарь |

## Связи между составными частями программы

Связи между основной процедурой и функциями программы выполняются в виде стандартных вызовов подпрограмм

# Чеклист

Таблица 8.1.1 – Чеклист

|  |  |  |  |
| --- | --- | --- | --- |
| № Тест кейса | Название проверки | Результат | Замечания и рекомендации |
| 1 | 2 | 3 | 4 |
| Реализация приложения на разных уровнях API | | | |
| 1 | Приложение корректно отображается на всех уровнях API изложенных в ТЗ(от версии android 7.0) | Положительно |  |
| Работа с активностями | | | |
| 2 | Все активности отображаются при переходе на них | Положительно |  |
| 3 | Добавлены все активности представленные в ТЗ | Положительно |  |
| Игра | | | |
| 4 | При нажатии на интерактивные объекты отображается соответствующая активность | Положительно |  |
| 5 | Клиент приходит | Положительно |  |
| 6 | Диалоговое окно с клиентом | Положительно |  |
| 7 | Список задач | Положительно | Нету вкладки завершенные |
| 8 | Система Магазина | Положительно |  |
|  |  |  |  |

Продолжение Таблицы 8.1.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 9 | Система Инвентаря | Отрицательно | Нету 2 вкладки и окошка информации |
| 10 | Система Крафта | Положительно |  |
| 11 | Разные клиенты | Функционал не реализован |  |
| 12 | Различные заказы | Функционал не реализован |  |
| 13 | Итоговые окна после покупки/продажи | Функционал не реализован | Работает только при покупке |
| 14 | Автосохранение | Положительно | Не сохраняет все данные |
| 15 | Система опыта | Положительно |  |
| 16 | Внутриигровая экономика | Положительно |  |
| Настройки звука | | | |
| 17 | Музыка | Функционал не реализован | Невозможно в Xamarin |
| 18 | Настройка звуков и музыки | Функционал не реализован |  |

# Текст программы