МОБИЛЬНОЕ ПРИЛОЖЕНИЕ

«Список покупок»

Руководство программиста

RU.02500646.00012-01 33 01-1

Листов 8

2019

**АННОТАЦИЯ**

Настоящее руководство программиста определяет требования к техническим средствам и программному обеспечению разработанного приложения, даёт его характеристику, описывает способы обращения к программе, входные и выходные данные.

Также руководство программиста рассказывает о возможных ошибках и сообщениях, которые при этом появляются, и способах решения этих ситуаций.

Руководство программиста составляется на основании ГОСТов:

1. ГОСТ 19.104-78 ЕСПД . Основные надписи;
2. ГОСТ 19.504-79 ЕСПД . Руководство программиста. Требования к содержанию и оформлению.

**СОДЕРЖАНИЕ**

[1 Условия применения программы 4](#_Toc26917445)

[1.1 Требования к техническим средствам 4](#_Toc26917446)

[1.2 Требования к общему программному обеспечению (ОПО) 4](#_Toc26917447)

[2 Характеристика программы 4](#_Toc26917448)

[3 Обращение к программе 5](#_Toc26917449)

[4 Входные и выходные данные 6](#_Toc26917450)

[5 Сообщения 8](#_Toc26917451)

* 1. **Условия применения программы**
  2. **Требования к техническим средствам**

Требования к техническим средствам мобильного приложения «Список продуктов»:

* 2-х ядерный процессор;
* объём оперативной памяти не менее 1Gb;
* 1 ГБ внутренней памяти
  1. **Требования к общему программному обеспечению (ОПО)**

Требования к общему программному обеспечению (ОПО), необходимому для мобильного приложения «Список продуктов» - это операционной система Android версии 5.0 (Lollipop) и выше.

1. **Характеристика программы**

Мобильное приложение предназначено для удобного и наглядного составления списка продуктов.

Приложение включает в себя:

* Область заполнения (наименование продукта, цена продукта, кнопка добавления продукта в список)
* Область вывода информации (выводятся списком на экран все добавленные продукты).

Приложение обеспечивает возможность выполнения перечисленных ниже функций:

* Возможность добавления наименование, цены продуктов в список;
* Вывод списка продуктов.

1. **Обращение к программе**

Внешнее управление содержимым системы осуществляется через интерфейс, описанный в документе «руководство пользователя».

Внутреннее управление приложением происходит через платформу разработки мобильных приложений Android Studio.

Структура проекта находится в папке app\src\main\java\com\example\list\_of\_foods и является полноценным набором файлов и программных компонентов для управления логикой, функциональной частью системы и ее работой.

Проект состоит из следующих папок:

* Java – содержит все java – классы и файлы управляющие логикой проекта;
* Drawable – хранит рисунки и другие графические объекты;
* Layout – хранит xml – разметку объектов интерфейса;
* Menu – содержит xml – разметку главной activity (экрана);
* Values – ресурс, который хранит ссылки на имена, цвета, стили, размеры различных объектов.

1. **Входные и выходные данные**

В данном разделе приводится описание организации используемой входной и выходной информации.

Приложение написано на языке java.

Все входные данные для данного мобильного приложения вводит пользователь. Для возможности ввода данных, первым создаётся интерфейс в xml - файле:

* TextBox для наименования продукта;
* TextBox для цены;
* Button для действия «добавить».

Все элементы интерфейса ввода добавляем в ресурс R, тем самым связываем интерфейс с java – классами и получаем возможность обращаться к ним:

final EditText name = findViewById(R.id.name);  
final EditText price = findViewById(R.id.price);  
final Button add= findViewById(R.id.add);

Пользователь вводит наименование продукта, его цену и нажимает на кнопку «+», тем сам передаёт значения на обработку. Эти значения принимает созданный статический класс Item. В нём содержится 2 параметра:

* String name (наименование со строковым типом данных);
* Int price (цена с целочисленным типом данных).

static class Item {  
 String name;  
 int price;

…

}

Создаётся конструктор для этого класса:

Item(String name, int price){

this.name = name;

this.price = price;}

Чтобы иметь возможность каждый раз выводить хранящиеся в классе Item параметры, создаём список ListView в xml – файле и добавляем его в ресурс R. Но ListView – это лишь оболочка, которая примет параметры и выведет их пользователю на экран, а для их передачи используется adapter.

Вызываем adapter:

final ItemsAdapter adapter = new ItemsAdapter();  
items.setAdapter(adapter);

Затем переопределяем (изменяем класс, который уже есть в Android Studio) класс ItemsAdapter:

class ItemsAdapter extends ArrayAdapter<Item> {{

// обращается к списку Item  
 public ItemsAdapter() {  
 super(MainActivity.this, R.layout.itemtwo);  
 }  
 @NonNull  
 @Override  
 public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {  
 final View view = getLayoutInflater().inflate(R.layout.itemtwo, null);  
 final Item itemtwo = getItem(position);  
 ((TextView) view.findViewById(R.id.name)).setText(itemtwo.name);  
 ((TextView) view.findViewById(R.id.price)).setText(String.valueOf(itemtwo.price));  
 return view;  
 }  
}

Здесь метод getLayoutInflater заполнят список Item данными.

Метод add.setOnClickListener делает так, что нажатие на кнопку «+» вызывает adapter, а тот в свою очередь заполняет список Item, передаёт этот список в элемент интерфейса ListView, а ListView делает вывод данных пользователю.

add.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 adapter.add(new Item(name.getText().toString(), Integer.valueOf(price.getText().toString())));  
 }  
});

1. **Сообщения**

В ходе изменения, доработки кода, программисту выдаются сообщения. Здесь приводится описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

Возможные ошибки, которые могут появиться при передачи данных adapter: Uncoditional layout inflation from View adapter.

Решение проблемы: Установить методу add.setOnClickListener в качестве 2 параметра null.

final View view = getLayoutInflater().inflate(R.layout.itemtwo, null);