**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе «Агрегатор кулинарных рецептов на Android».

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

1. постановка задачи на разработку программы;
2. описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
3. описание и обоснование выбора метода организации входных и выходных данных;
4. описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [7].

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

СОДЕРЖАНИЕ

[1. ВВЕДЕНИЕ 8](#_Toc40208061)

[1.1. Наименование программы 8](#_Toc40208062)

[1.2. Документы, на основании которых ведется разработка 8](#_Toc40208063)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 9](#_Toc40208064)

[2.1. Назначение программы 9](#_Toc40208066)

[2.1.1. Функциональное назначение 9](#_Toc40208067)

[2.1.2. Эксплуатационное назначение 9](#_Toc40208068)

[2.2. Краткая характеристика области применения 9](#_Toc40208069)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 10](#_Toc40208070)

[3.1. Постановка задачи на разработку программы 10](#_Toc40208072)

[3.1.1. Задачи работы (Android-клиент) 10](#_Toc40208073)

[3.1.2. Задачи работы (серверная часть) 10](#_Toc40208091)

[3.2. Описание алгоритма и функционирования программы 10](#_Toc40208097)

[3.2.1. Описание построения Android приложения 10](#_Toc40208098)

[3.2.2. Описание построения серверной части 11](#_Toc40208099)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 16](#_Toc40208117)

[3.3.1. Описание метода организации входных и выходных данных 16](#_Toc40208118)

[3.3.2. Обоснования выбора метода организации входных и выходных данных 16](#_Toc40208119)

[3.4. Описание и обоснование выбора состава технических и программных средств 16](#_Toc40208122)

[3.4.1. Состав технических и программных средств 16](#_Toc40208123)

[3.4.2. Обоснование выбора технических и программных средств 17](#_Toc40208132)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 18](#_Toc40208137)

[4.1. Предполагаемая потребность 18](#_Toc40208139)

[4.2. Экономические преимущества разработки по сравнению с отечественными зарубежными образцами или аналогами 18](#_Toc40208140)

[5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 19](#_Toc40208141)

[ПРИЛОЖЕНИЕ 1 ТЕРМИНОЛОГИЯ 20](#_Toc40208155)

[*Парсинг! 20*](#_Toc40208156)

[**ПРИЛОЖЕНИЕ 3 ДИАГРАММА КЛАССОВ** 21](#_Toc40208157)

[**ПРИЛОЖЕНИЕ 4 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ** 22](#_Toc40208158)

[**ПРИЛОЖЕНИЕ 5 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ** 25](#_Toc40208159)

1. ВВЕДЕНИЕ
   1. Наименование программы

Наименование программы – «Агрегатор кулинарных рецептов на Android».

Наименование программы на английском языке – «Aggregator of Recipes for Android»

Краткое наименование программы – «На плите».

* 1. Документы, на основании которых ведется разработка

Разработка ведется на основании приказа Национального исследовательского университета "Высшая школа экономики" № 2.3-02/2004-04 от 20.04.2020.

1. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ
2. 1. Назначение программы
      1. Функциональное назначение

Функциональным назначением программы является поиск кулинарных рецептов с различных интернет-сайтов с помощью встроенных фильтров, разделов или с помощью названия рецепта, а также отображение их в как в кратком, так и в подробном формате.

К функциональным возможностям серверной части относится сбор и хранение информации о кулинарных рецептах, а также взаимодействием с мобильным приложением.

* + 1. Эксплуатационное назначение

Решение поиска кулинарных рецептов на Android является востребованным на сегодняшний день, так как с каждым годом увеличивается рост посещения интернет-сайтов с мобильных устройств, а, зачастую, кулинарные сайты не поддерживают мобильную версию, что приводит к некомфортной работе с материалами сайта.

Основная цель программы – предоставить пользователю кулинарные рецепты по его запросу, а также облегчить работу с кулинарными рецептами на операционной системе Android.

Программа предназначена для пользователей на операционной системе Android.

* 1. Краткая характеристика области применения

«Агрегатор кулинарных рецептов на Android» – программа, позволяющая находить и отображать кулинарные рецепты, взятые с различных кулинарных сайтов или порталов, а это в свою очередь позволяет пользователям экономить своё время на поиске и просмотре кулинарных рецептов.

1. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ
2. 1. Постановка задачи на разработку программы
      1. Задачи работы (Android-клиент)
3. Реализовать возможность просмотра списка кулинарных рецептов;
4. Реализовать постепенную загрузку кулинарных рецептов для экономии трафика и меньшей нагрузки на мобильное устройство;
5. Реализовать отображение индикатора процесса загрузки данных с сервера;
6. Реализовать фильтрацию кулинарных рецептов по следующим фильтрам:
   1. Популярные рецепты;
   2. Новые рецепты;
   3. Случайные рецепты;
7. Реализовать отображение кулинарных рецептов по разделам;
8. Реализовать просмотр расширенной информации о рецепте;
9. Реализовать просмотр списка ингредиентов и списка шагов приготовления рецепта;
10. Реализовать просмотр видеоматериала, приложенного к рецепту;
11. Реализовать возможность добавления рецепта в раздел «Избранное»;
12. Реализовать возможность удаление рецепта из раздела «Избранное»;
13. Реализовать возможность добавление ингредиента в раздел «Корзина продуктов»;
14. Реализовать возможность удаление ингредиента из раздела «Корзина продуктов»;
15. Реализовать возможность перехода на веб-страницу источника рецепта;
16. Реализовать возможность поделиться информацией о рецепте в сторонних приложениях;
    * 1. Задачи работы (серверная часть)
17. Реализовать сбор списка рецептов с кулинарных сайтов, а также сбор информации о конкретном рецепте по запросу пользователя:
18. Реализовать получение данных для каждого сайта с помощью HTTP GET или HTTP POST запросов;
19. Реализовать сбор информации с помощью «CSS селекторов»;
20. Реализовать временное кэширование рецептов в базе данных MySQL;
21. Выдавать информацию о рецепте при получении HTTP-GET запроса в формате JSON;
    1. Описание алгоритма и функционирования программы
       1. Описание построения Android приложения

В Android приложении задумано 7 активностей (см. терминологию) и 3 фрагмента (см. терминологию)

1. **MainActivity** (Рис. 1). Активность, формирующая интерфейс для отображения списка рецептов, где каждый рецепт подкреплён названием рецепта, изображением, а также названием источника. Является самой первой активностью, которая загружается при запуске приложения. В этой активности есть одно поле для поиска кулинарного рецепта по названию, а также кнопка для запуска поиска, выпадающий список для выбора фильтрации (Рис. 2), кнопка для открытия бокового меню (Рис. 3).
2. **RecipeActivity**. Активность, формирующая интерфейс для отображения информации о рецепте. Имеет кнопку «Назад», расположенную в левом верхнем углу, которая возвращает пользователя к предыдущей активности, кнопку «Поделиться», кнопку для перехода на сайт источника, а также плавающую кнопку для добавления рецепта в раздел «Избранное» или удаление рецепта из раздела «Избранное».

Фрагменты:

* 1. **RecipeDescriptionFragment** (Рис. 4). Фрагмент, который содержит краткое описание рецепта и какую-либо информацию о нём, а также видеоматериал при его наличии.
  2. **RecipeIngredientsFragment** (Рис. 5). Фрагмент, который содержит список ингредиентов. Каждый ингредиент подкреплён названием и количеством, а также элементом для добавления ингредиента в раздел «корзина продуктов» или для удаления ингредиента из раздела «корзина продуктов».
  3. **RecipeStepsFragment** (Рис. 6). Фрагмент, который содержит в себе список шагов для приготовления рецепта, где каждый элемент списка содержит в себе описание и изображение при его наличии.

1. **PlatesActivity** (Рис. 7). Активность, формирующая интерфейс для отображения списка разделов, где каждый элемент списка – изображение и название поверх этого изображения. Содержит в себе кнопку, позволяющую вернуться к предыдущей активности.
2. **SectionRecipesActivity** (Рис. 8). Активность, формирующая интерфейс для отображения списка рецептов, где каждый рецепт подкреплён названием рецепта, изображением, а также названием источника. Содержит в себе название раздела, расположенное на вернем части активности, а также кнопки, позволяющей вернуться к предыдущей активности.
3. **SavedIngredientsActivity** (Рис. 9). Активность, формирующая интерфейс для отображения списка сохранённых ингредиентов (наз. «Корзина продуктов»), каждый элемент списка содержит в себе название ингредиента, его количества, название рецепта, которому он принадлежит, а также кнопку для удаления ингредиента из данного раздела. Содержит в себе кнопку, позволяющую вернуться к предыдущей активности.
4. **SavedRecipesActivity** (Рис. 10). Активность, формирующая интерфейс для отображения списка сохранённых рецептов (наз. «Избранное»). Элемент списка содержит в себе название рецепта, изображение, а также название источника. Активность также содержит элемент **SearchView** (см. терминологию), который позволяет фильтровать список по введённому в него названию рецепта, а также кнопку «Назад», чтобы пользователь мог вернуться к предыдущей активности.
5. **InfoActivity**. Активность, содержащая в себе информацию о приложении, а также кнопку «Назад», чтобы пользователь мог вернуться к предыдущей активности.

Изображение выглядит как снимок экрана, еда

Автоматически созданное описание

Рис. 1-3: MainActivity.



Рис. 4-6: RecipeActivity.

RecipeDescriptionFragment

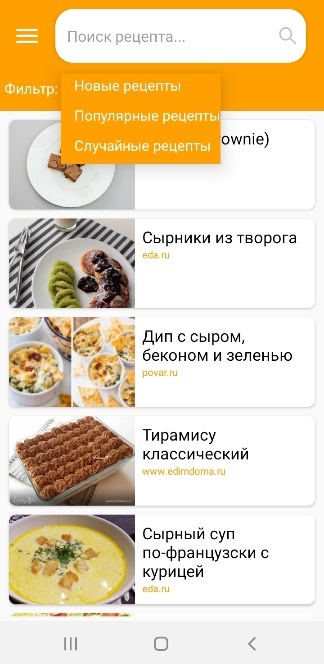


Рис. 2: Выезжающий список.

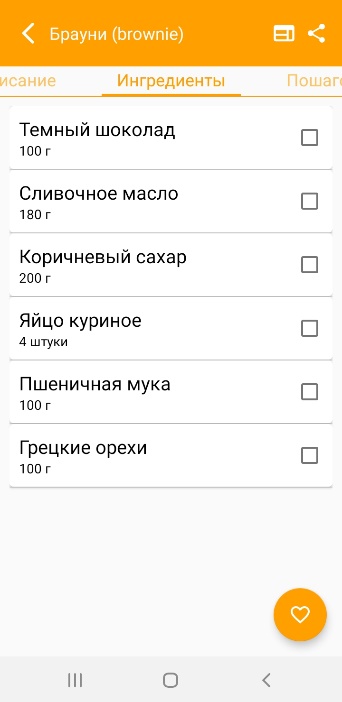


Рис. 5: RecipeIngredientsFragment

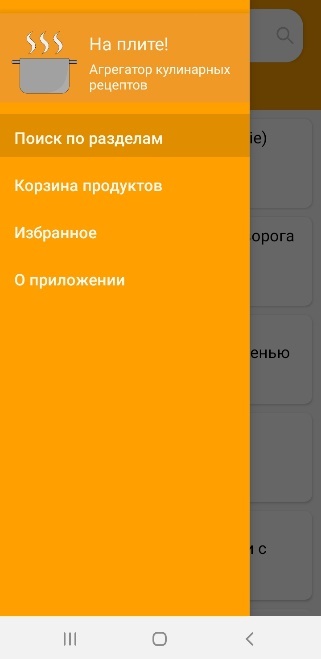


Рис. 3: Боковое меню

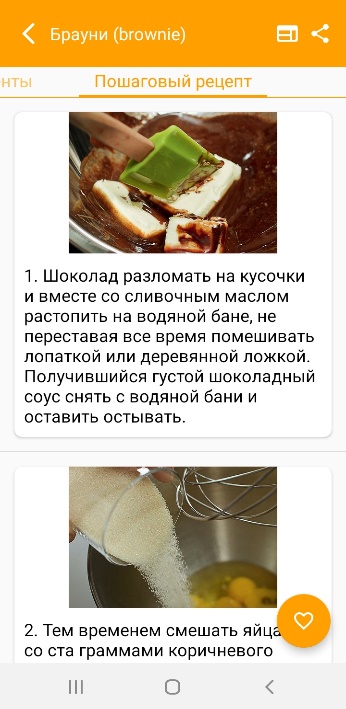


Рис. 6: RecipeStepsFragment

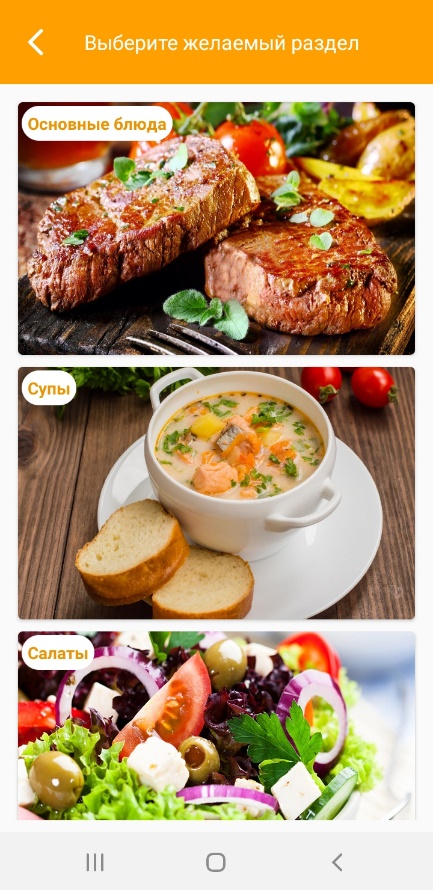


Рис. 7: PlatesActivity.

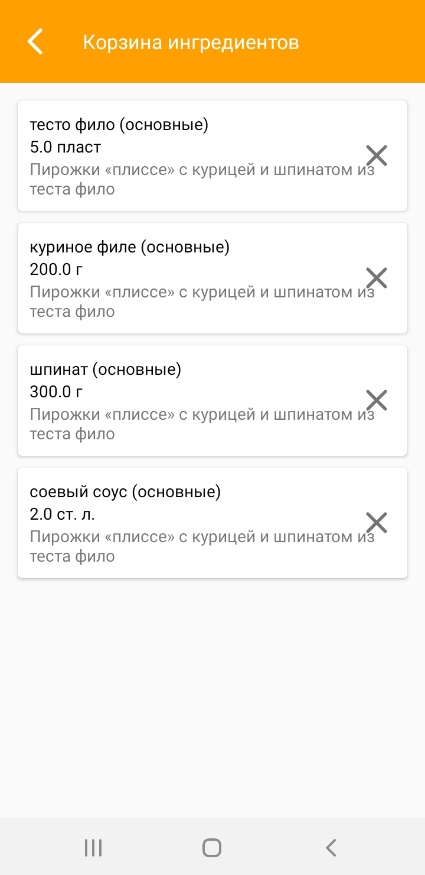


Рис. 9: SavedIngredientsActivity

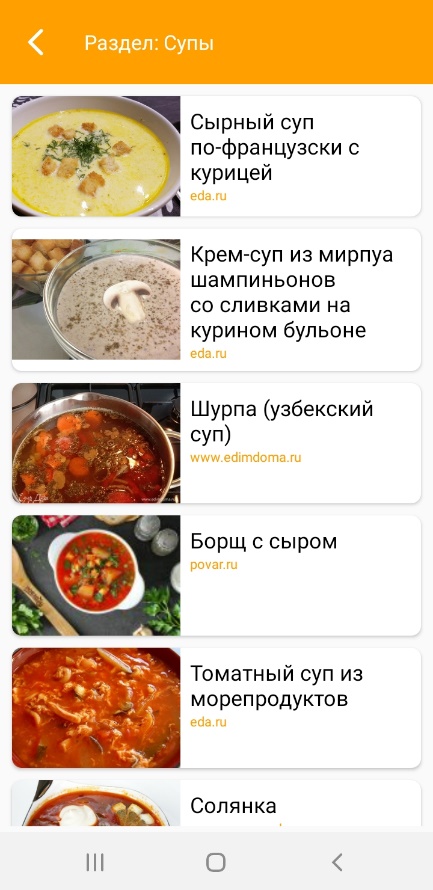


Рис. 8: SectionRecipesActivity.

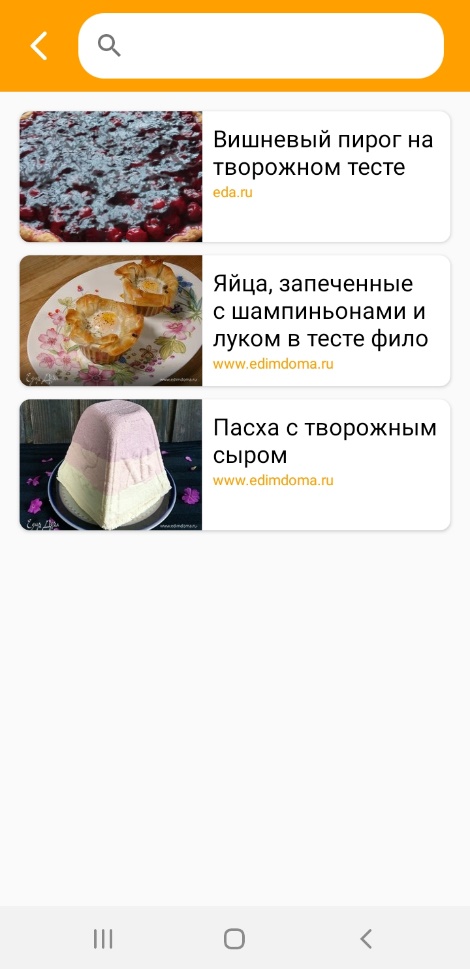


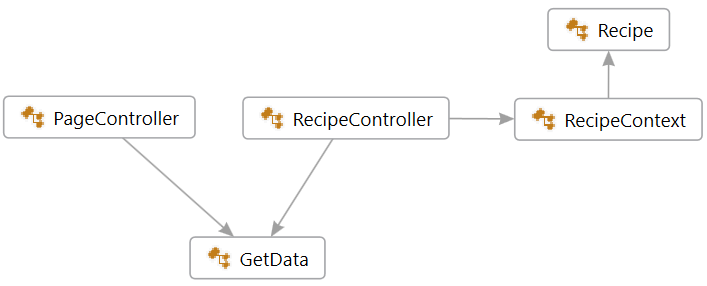
Рис. 10: SavedRecipesActivity

* + 1. Описание построения серверной части
       1. Структура серверной части

Структура серверной части содержит 2 контроллера:

1. **PageController** – контроллер (см. терминологию), отвечающий за обработку входящих HTTP-GET запросов для получения списка рецептов в формате JSON (см. терминологию).
2. **RecipeController** – контроллер, отвечающих за обработку входящих HTTP-GET запросов для получения информации о рецепте в формате JSON.

А также класс RecipeContext, наследующий класс DbContext, предоставляемый библиотекой Entity Framework Core для связки базы данных MySQL и ASP NET Core.

Класс Recipe описывает структуру кэшируемого (см. терминологию) объекта в базе данных.

* + - 2. Контроллеры

**PageController:** Данный контроллер принимает информацию о разделе, номере страницы (необязательный параметр), а также названии рецепта (необязательный параметр):

Параметр section (раздел) поддерживает следующие значения:

1. Popular – популярные рецепты;
2. New – новые рецепты;
3. Random – случайные рецепты;
4. Recipe – поиск по названию;
5. Горячее;
6. Супы;
7. Салаты;
8. Выпечка;
9. Десерты;
10. Закуски;
11. Соусы;

Параметр в виде номера страницы – положительное целочисленное значение, является необязательным параметром для ввода. Значение по умолчанию – 1.

Если был выбран раздел «Recipe», то ещё одним параметром следует указать параметр «recipeName», который принимает название рецепта, по которому следует начать поиск.

**RecipeController:** Данный контроллер принимает информацию в виде одного параметре – url - адреса кулинарного рецепта.

* + - 1. Логирование информации

Каждый из контроллеров производит логирование (см. терминологию) данных как в консольное приложение, так и в текстовые документы, фильтруя их по уровню:

1. Debug
2. Information
3. Warning

Так, в файл «nlog-all-yyyy-mm-dd.log» поступают все данные, в файл «warning.log» поступает информация уровня Warning, а в файл «info.log» информация уровня Information.

В консольное приложению поступает информация уровня Information или Warning.

Пример логирования из класса RecipeController представлен на языке C#:

\_logger.LogDebug($"Рецепт существует в БД. Id = {recipeDb.Id}, Date = {recipeDb.Date}");

// Если рецепт нужно обновить:

if ((startTime - recipeDb.Date).TotalHours > diffHours)

{

\_logger.LogDebug($"Требуется обновление рецепта.");

try

{

recipe = await GetData.GetRecipe(url);

recipeDb.RecipeFull = recipe;

recipeDb.Date = startTime;

db.SaveChanges();

\_logger.LogInformation($"Статус: Ok.");

LogTime(startTime);

return Ok(recipe);

}

catch(Exception e)

{

\_logger.LogWarning(e, $"Произошла ошибка при парсинге рецепта! Выдан рецепт из БД.");

\_logger.LogInformation($"Статус: Ok.");

LogTime(startTime);

return Ok(recipeDb.RecipeFull);

}

}

* + - 1. Кэширование данных.

Кэширование данных, а в частности информации о полном рецепте происходит каждые 24 часа, это сделано для того, чтобы при многочисленных запросах от пользователей не тратить ресурсы сервера, а также чтобы обновлять информацию о новых рецептах, которые не всегда после публикации имеют совершенный вид.

Пример представлен на языке C#:

using(RecipeContext db = new RecipeContext())

{

// Если в БД есть рецепт:

if((recipeDb = db.Recipes.FirstOrDefault(x => x.Url == url)) != null)

{

…

// Если рецепт нужно обновить:

if ((startTime - recipeDb.Date).TotalHours > diffHours)

{

// Происходит обновление существующего в БД рецепта и возвращается новый.

…

}

// Если рецепт есть в БД и обновлять его ненужно:

else

{

// Возвращается рецепт из БД.

…

}

}

// Если в БД нет рецепта:

else

{

// Получаем рецепт с помощью алгоритма парсинга.

\_logger.LogDebug($"Рецепта не существует в БД.");

try

{

…

}

catch (Exception e)

{

// Если происходит ошибка – уведомляет пользователя об этом.

…

}

}

}

* + - 1. Описание построения парсера страниц со списком рецептов

Для парсера веб-страниц со списком рецептов предусмотрены следующие классы: ParserPage, HtmlLoader, PovarPageParser, EdaPageParser, PovarenokPageParser, EdimDomaPageParser, EdaPageSettings, EdimDomaPageSettings, PovarPageSettings, PovarenokPageSettings.

1. **ParserPage –** класс, который направляет полученные от пользователя параметры на получение страниц с рецептами в класс HtmlLoader, а после направляет его результат в соответствующий парсер.
2. **HtmlLoader –** класс, который собирает внутри себя правильную ссылку на страницу с рецептами в соответствии с заданными параметрами и на выходе выдаёт полученный с помощью HTTP-GET или HTTP-POST запросами код страницы кулинарного сайта.
3. **PovarPageParser,** унаследованный от **IParserPage**<RecipeShort[]>. Отвечает за преобразование исходного кода страницы сайта povar.ru в коллекцию типа RecipeShort[]. При помощи библиотеки «AngleSharp» проводит поиск информации по нужным тегам и формирует из них объект.

На языке C# это выглядит следующим образом:

public RecipeShort[] Parse(IHtmlDocument document, IParserPageSettings settings)

{

var recipesBody = document.QuerySelectorAll("div.recipe\_list > div.recipe");

double indexStartPopularity = settings.IndexPopularity;

return (from recipe in recipesBody

let anyBody = recipe.QuerySelector("h3 > a")

let title = anyBody.TextContent

let url = "https://povar.ru" + anyBody.Attributes[0].Value

let pictureBody = recipe.QuerySelector("img")

let imageUrl = pictureBody.Attributes[0].Value

let indexPopularity = indexStartPopularity -= settings.IndexStep

select new RecipeShort(title, new Image(imageUrl), url, indexPopularity)).ToArray();

}

1. **EdaPageParser**, **EdimDomaParser**, **PovarenokPageParser** функционал идентичен с **PovarPageParser**, только действует на сайте «eda.ru», «edimdoma.ru», «povarenok.ru» соответственно.
2. Классы **EdaPageSettings**. **EdimDomaPageSettings**, **PovarPageSettings**, **PovarenokPageSettings** наследуются от **IParserPageSettings** и задают каждому сайту индивидуальные настройки, используемые при формировании ссылки.
   * + 1. Описание построения парсера страниц с рецептом.

Необходимые классы: HtmlLoader, ParserRecipe, PovarenokRecipeParser, EdaRecipeParser, EdimDomaRecipeParser, PovarRecipeParser, PovarenokRecipeSettings, PovarRecipeSettings, EdaRecipeSettings, EdimDomaRecipeSettings.

1. **ParserRecipe** - класс, который направляет полученные от пользователя параметры на получение страниц с рецептами в класс **HtmlLoader**, а после направляет его результат в соответствующий парсер.
2. **HtmlLoader –** класс, который собирает внутри себя правильную ссылку на страницу с рецептами в соответствии с заданными параметрами и на выходе выдаёт полученный с помощью HTTP-GET или HTTP-POST запросами код страницы кулинарного сайта.
3. **PovarRecipeParser,** унаследованный от **IParserRecipe<RecipeFull>**. Отвечает за преобразование исходного кода страницы сайта povar.ru с рецептом в объект типа «RecipeFull». При помощи библиотеки «AngleSharp» проводит поиск информации по нужным тегам и формирует из них объект

На языке C# часть алгоритма выглядит следующим образом:

var recipeBody = document.QuerySelector("div.cont\_area");

if (recipeBody is null)

return new RecipeFull();

Title = recipeBody.QuerySelector("h1.detailed[itemprop='name']").TextContent;

TitleImage = new Image(recipeBody.QuerySelector("div.bigImgBox")?

.QuerySelector("a").FirstElementChild.Attributes[0].Value);

Description = String.Empty;

foreach (var textLine in recipeBody.QuerySelectorAll("span.detailed\_full").Select(x => x.TextContent))

Description += textLine.Trim() + Environment.NewLine;

Description.Replace("\t", "").Replace(" ", "");

var ingredientsBody = recipeBody.QuerySelector("ul.detailed\_ingredients").QuerySelectorAll("li");

int countIngredients = ingredientsBody.Length;

Ingredient[] ingredients = new Ingredient[countIngredients];

for (int i = 0; i < countIngredients; i++)

{

string title = ingredientsBody[i].Attributes[1].Value;

string unit = ingredientsBody[i].TextContent

.Replace(title, string.Empty)

.Remove(0, 35);

ingredients[i] = new Ingredient(title, unit, Title);

}

Ingredients = ingredients;

1. **EdaRecipeParser**, **EdimRecipeParser**, **PovarenokRecipeParser** функционал идентичен с **PovarRecipeParser**, только действует на сайте «eda.ru», «edimdoma.ru», «povarenok.ru» соответственно.
2. Классы **EdaRecipeSettings**. **EdimDomaRecipeSettings**, **PovarRecipeSettings**, **PovarenokRecipeSettings** наследуются от **IParserRecipeSettings** и задают каждому сайту индивидуальные настройки, используемые при формировании ссылки.
   1. Описание и обоснование выбора метода организации входных и выходных данных
      1. Описание метода организации входных и выходных данных

Входными и выходными данными в Android-приложении является интерфейс.

Входными данными серверной части является параметры, содержащиеся в теле HTTP-GET запроса, а выходными данными является документ формата JSON.

* + 1. Обоснования выбора метода организации входных и выходных данных

Формат вывода данных на серверной части был выбран JSON, а не его похожий аналог – XML, по ряду следующих причин:

1. JSON поддерживает: строки, числа, объекты, коллекции.
2. Размер документа формата JSON значительно меньше, а значит повышается скорость передачи данных.
   1. Описание и обоснование выбора состава технических и программных средств
      1. Состав технических и программных средств

Технические требования сервера:

1. Операционная система: Windows 7 SP1 и выше/Mac OS X 10.12+ x64/Ubuntu 16.04+
2. Наличие Microsoft .NET Core SDK 3.1.
3. Наличие MySQL Server 8.0 и выше.
4. Наличие веб-сервера.

Технические требования Android-приложения:

1. Операционная система мобильного устройства должна бывать Android версии 6.0 «Marshmallow» и выше.
2. Постоянное подключение к сети интернет;
3. Для использования кнопок требуется сенсорный экран;
4. Память устройства должна быть не менее 80 Мб (рекомендуется более 120 Мб).
   * 1. Обоснование выбора технических и программных средств

Серверная часть:

Чтобы поддерживать высокую производительность и стабильность серверной части – используется последняя версия Net Core, которая имеет системные требования:

1. Операционная система: Windows 7 SP1 и выше/Mac OS X 10.12+ x64/Ubuntu 16.04+

Наличие MySQL Server необходимо, чтобы создавать и работать с базой данных. При этом выбор MySQL аргументирован так:

1. Поддержка SQL
2. Поддержка многих операционных систем**.**
3. **Высокая скорость и производительность.**

Наличие веб-сервера необходимо, чтобы ретранслировать запросы пользователей из внешней сети на внутреннюю сеть.

Android приложение:

Приложение разрабатывалось под минимальную версию операционной системы Android версии 6.0 «Marshmallow», потому что согласно официальному отчёту компании Google от 10 апреля 2020 года устройств с операционной системой 6.0 и выше – 84.9% от общего количества.

1. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ
2. 1. Предполагаемая потребность

Программу может использовать любой человек, желающий проверить знания других людей.

* 1. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

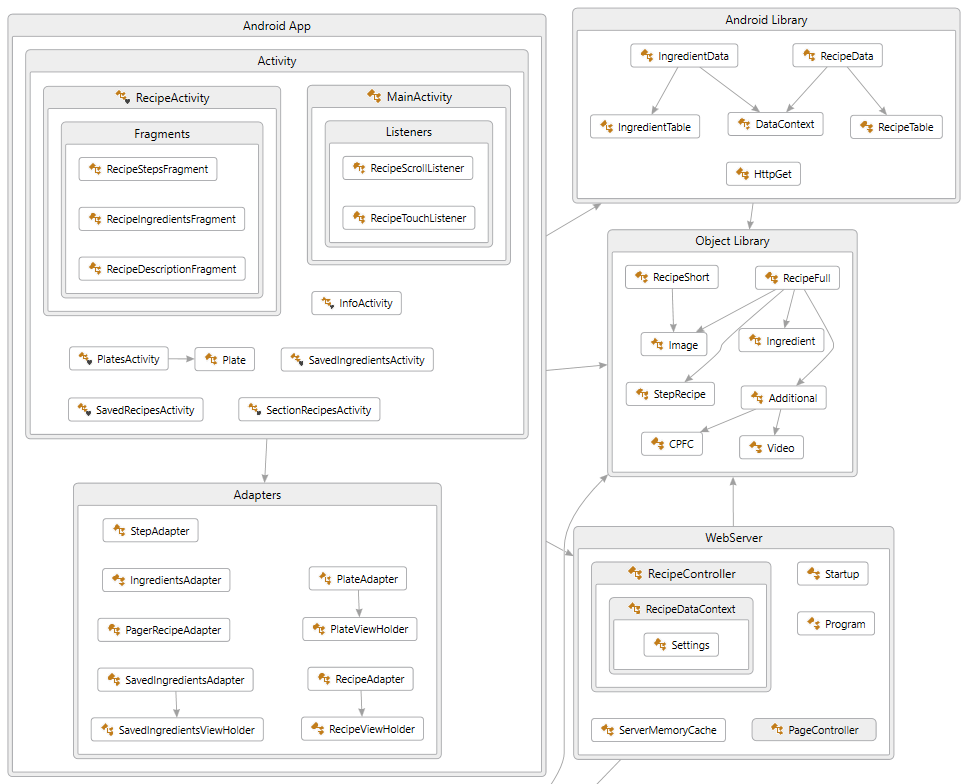
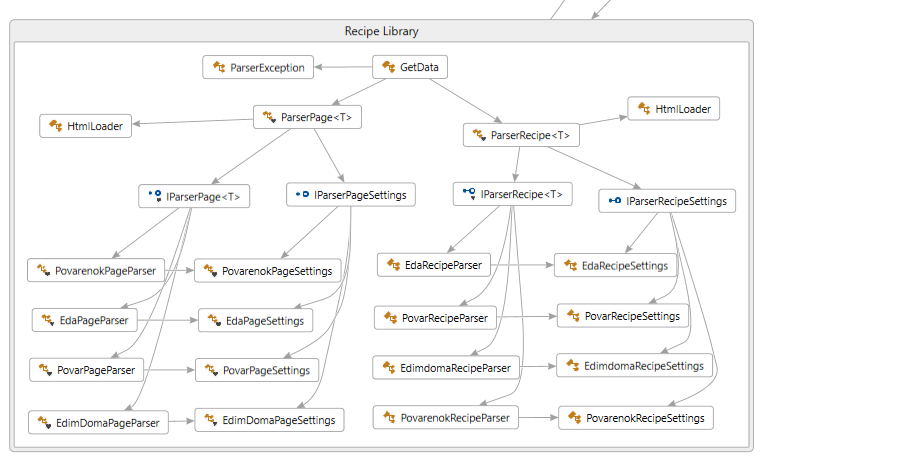
Существует как отечественные, так и зарубежные аналоги, однако данное приложение имеет следующие преимущества:

1. Распространяется бесплатно;
2. Не требует вложения денежных средств во время использования;
3. Не имеет рекламных баннеров;
4. Имеет неограниченный срок службы;
5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ
6. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
15. Серия учебников по работе с ASP.NET Core MVC и EF Core. [Электронный ресурс]// URL: <https://docs.microsoft.com/ru-ru/aspnet/core/data/ef-mvc/?view=aspnetcore-3.1> (Дата обращения: многократно c 01.03.2020 по 11.05.2020)
16. Руководство по ASP.NET Core 3. [Электронный ресурс]// URL: <https://metanit.com/sharp/aspnet5/> (Дата обращения: многократно c 01.03.2020 по 11.05.2020)
17. Руководство по Entity Framework Core [Электронный ресурс]// URL: <https://metanit.com/sharp/entityframeworkcore/> (Дата обращения: многократно c 01.03.2020 по 11.05.2020)
18. Build anything on Android [Электронный ресурс]// URL: <https://developer.android.com/> (Дата обращения: многократно c 01.03.2020 по 11.05.2020)
19. Требования к системе для .NET Core [Электронный ресурс]// URL: <https://docs.microsoft.com/ru-ru/dotnet/core/install/dependencies?pivots=os-windows&tabs=netcore31> (Дата обращения: 13.05.2020, режим доступа: свободный).
20. AngleSharp Documentation [Электронный ресурс]// URL <https://anglesharp.github.io/docs.html> (Дата обращения: 13.05.2020, режим доступа: свободный).
21. MySQL documentation [Электронный ресурс]// URL <https://dev.mysql.com/doc/> (Дата обращения: 13.05.2020, режим доступа: свободный).

# ПРИЛОЖЕНИЕ 1 ТЕРМИНОЛОГИЯ

1. **Контроллер** (англ. *Controller*) – интерпретирует действия пользователя, оповещая модель о необходимости изменения
2. **Кэш** (англ. *cache*) – промежуточный буфер с быстрым доступом к нему, содержащий информацию, которая может быть запрошена с наибольшей вероятностью.
3. **Обратный прокси-сервер** (англ. *reverse proxy*) — тип проекси-сервера, который ретранслирует запросы клиентов из внешней сети на один или несколько серверов, логически расположенных во внутренней сети.
4. **Лог** (протокол, журнал, лог, англ*. log*) — файл с записями о событиях в хронологическом порядке, простейшее средство обеспечения журналирования.
5. **Логирование** – процесс журналирования.
6. **Активность** (англ. *Activity*) – это компонент приложения, который выдает экран, и с которым пользователи могут взаимодействовать для выполнения каких-либо действий.
7. **Фрагмент** (англ. *Fragment*) – Фрагмент представляет поведение или часть пользовательского интерфейса в операции (класс Activity). В одной активности может быть несколько фрагментов.
8. **RecyclerView** – Элемент визуальной группы для отображения списков объектов, переиспользующий место на экране устройства. То есть хранится столько объектов, сколько может уместиться на экране, а при прокрутке элемента, существующие объекты перерисовываются, а не продолжаются хранится в памяти.
9. **SearchView** – элемент визуальной группы представляет собой изображение лупы, которое разворачивается в текстовое поле при нажатии. Используется для поиска, например, в базе данных.
10. **База данных** (сокр. БД) - совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
11. **JSON (**англ. *JavaScript Object Notation*)– Текстовый формат обмена данными.
12. **CSS** **Селектор** (англ. CSS Selectors) – часть CSS-правила, используются для того, чтобы "найти" (или выбрать) HTML элементы.
13. **Парсинг** (англ. *Parsing*) **—** это извлечение, сбор любой открытой информации с веб-сайта.

**ПРИЛОЖЕНИЕ 3  
ДИАГРАММА КЛАССОВ**



**ПРИЛОЖЕНИЕ 4  
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

Таблица 4.1

Описание и функциональное назначение классов

| Класс | Назначение |
| --- | --- |
| RecipeFull | Модель полного рецепта. |
| RecipeShort | Модель краткого рецепта. |
| Additional | Модель дополнительного описания рецепта. |
| CPFC | Модель, содержащая информацию о калориях, белках, жирах и углеводах. |
| Image | Модель изображения. |
| Ingredient | Модель ингредиента. |
| StepRecipe | Модель шага рецепта. |
| ServerMemoryCache | Класс, отвечающий за кэширование ответов сервера. |
| PageController | Контроллер для кратких рецептов. |
| RecipeController | Контроллер для полного рецепта |
| RecipeContext | Класс для работы с базой данных. |
| Recipe | Класс, представляющий структуру объекта, кэшируемого в базе данных. |
| GetData | Класс, отвечающий за получение данных парсером. |
| ParserException | Класс исключения. |
| HtmlLoaderPage | Класс для формирования url-ссылки на страницу со списком рецептов, а также для скачивания исходного кода страницы. |
| ParserPage | Класс для парсинга страницы с рецептами. |
| EdaPageParser | Класс, в котором описан алгоритм парсинга для списка рецептов сайта eda.ru |
| EdimDomaPageParser | Класс, в котором описан алгоритм парсинга для списка рецептов сайта edimdoma.ru |
| PovarenokPageParser | Класс, в котором описан алгоритм парсинга для списка рецептов сайта povarenok.ru |
| PovarPageParser | Класс, в котором описан алгоритм парсинга для списка рецептов сайта povar.ru |
| EdaPageSettings | Класс, в котором описаны настройки для парсинга списка рецептов для сайта eda.ru |
| EdimDomaPageSettings | Класс, в котором описаны настройки для парсинга списка рецептов для сайта edimdoma.ru |
| PovarenokPageSettings | Класс, в котором описаны настройки для парсинга списка рецептов для сайта povarenok.ru |
| PovarPageSettings | Класс, в котором описаны настройки для парсинга списка рецептов для сайта povar.ru |
| HtmlLoaderRecipe | Класс для формирования url-ссылки на страницу с рецептом, а также для скачивания исходного кода страницы. |
| ParserRecipe | Класс для парсинга страницы с рецептом. |

|  |  |
| --- | --- |
| Продолжение таблицы 4.1 | |
| EdaRecipeParser | Класс, в котором описан алгоритм парсинга для одного рецепта с сайта eda.ru. |
| EdimDomaRecipeParser | Класс, в котором описан алгоритм парсинга для одного рецепта с сайта edimdoma.ru |
| PovarRecipeParser | Класс, в котором описан алгоритм парсинга для одного рецепта с сайта povar.ru. |
| PovarenokRecipeParser | Класс, в котором описан алгоритм парсинга для одного рецепта с сайта povarenok.ru. |
| EdaRecipeSettings | Класс, в котором описаны настройки для парсинга рецепта для сайта eda.ru |
| PovarRecipeSettings | Класс, в котором описаны настройки для парсинга рецепта для сайта povar.ru |
| PovarenokRecipeSettings | Класс, в котором описаны настройки для парсинга рецепта для сайта povarenok.ru |
| EdimDomaRecipeSettings | Класс, в котором описаны настройки для парсинга рецепта для сайта edimdoma.ru |
| InfoActivity | Активность, которая содержит информацию о приложении и кнопку «Назад». |
| MainActivity | Главная активность, которая содержит список рецептов, поле для поиска рецепта, фильтры, а также боковое меню. Содержит логику равномерной загрузки рецептов, а также логику фильтрации рецептов. |
| PlatesActivity | Активность, которая содержит список разделов для поиска рецептов, кнопку «Назад». |
| RecipeActivity | Активность, которая содержит информацию о рецепте, кнопку «Поделиться», кнопку «Перейти к источнику рецепта», кнопку «Добавить в избранное», кнопку «Назад». Содержит 3 фрагмента: RecipeDescriptionFragment, RecipeIngredientsFragment, RecipeStepsFragment. |
| SavedIngredientsActivity | Активность, содержащая список сохранённых ингредиентов, кнопку «Назад». Прописана логика просмотра ингредиентов и их удаление. |
| SavedRecipesActivity | Активность, содержащая список сохранённых рецептов, кнопку «Назад». Прописана логика просмотра ингредиентов и их удаление. |
| SectionRecipesActivity | Активность, отображающая список рецептов какого-либо раздела. Содержит кнопку «Назад». |
| IngredientsAdapter | Адаптер, в котором содержится логика взаимодействия с элементом списка, для отображения и формирования списка ингредиентов. |
| PagerRecipeAdapter | Адаптер, где прописана логика отображения фрагментов в RecipeActivity. |
| PlateAdapter | Адаптер, в котором прописана логика инициализации элементов списка с разделами, а также событие при нажатии на элемент списка. |
| RecipeAdapter | Адаптер, в котором прописана логика инициализации элементов списка с рецептами, а также событие при нажатии на элемент списка. |

|  |  |
| --- | --- |
|  | |
| Продолжение таблицы 4.1 | |
| SavedIngredientsAdapter | Адаптер, в котором содержится логика взаимодействия с элементом списка, для отображения и формирования списка сохранённых ингредиентов. |
| StepAdapter | Адаптер, в котором прописана логика инициализации элементов списка шагов к рецепту. |
| RecipeDescriptionFragment | Фрагмент, содержащий описание рецепта. |
| RecipeIngredientsFragment | Фрагмент, содержащий список ингредиентов, взаимодействуя с IngredientsAdapter. |
| RecipeStepsFragment | Фрагмент, содержащий список шагов рецепта, взаимодействуя с StepAdapter. |
| Plate | Модель раздела с рецептами. |
| RecipeScrollListener | Слушатель для отслеживания положения списка рецептов, чтобы запустить обновление списка, если в нём закончились элементы. |
| RecipeData | Модель рецепта для хранения в локальной БД. |
| IngredientData | Модель ингредиента для хранения в локальной БД. |
| HttpContext | Класс, для взаимодействия мобильного-приложения и сервера. |
| DataContext | Класс, содержащий вспомогательные методы для работы с данными. |
| IngredientTable | Класс, предназначенный для работы с ингредиентами в базе данных на Android. |
| RecipeTable | Класс, предназначенный для работы с рецептами в базе данных на Android. |
| PlateViewHolder | Класс, описывающий шаблон в виде плитки из названия и картинки. Используется для создания списка разделов. |
| RecipeViewHolder | Класс, описывающий шаблон в виде краткого рецепта, состоящего из картинки, названия источника и заголовка. |
| SavedIngredientsViewHolder | Класс, описывающий шаблон в виде ингредиентов, состоящих из названия и количества. |

**ПРИЛОЖЕНИЕ 5  
ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ**

Таблица 5.1

Описание полей, методов и свойств класса RecipeFull.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| RecipeFull | public | Конструктор | - | | Конструктор для полного рецепта без параметров |
| RecipeFull | public | Конструктор | string url, string title, Image titleImage, string description, Ingredient[] ingredients,StepRecipe[] stepsRecipe, Additional additional | | Конструктор для полного рецепта с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Key | public | Guid | get, set | Первичный ключ, используется для идентификации рецепта в базе данных. | |
| Url | public | String | get | URL-адрес рецепта. | |
| Title | public | String | get | Заголовок рецепта. | |
| TitleImage | public | Image | get | Главное изображение рецепта. | |
| Description | public | String | get | Описание рецепта. | |
| Ingredients | public | Ingredient[] | get | Список ингредиентов рецепта. | |
| StepsRecipe | public | StepRecipe[] | get | Шаги рецепта. | |
| Additional | public | Additional | get | Дополнительное описание рецепта. | |

Таблица 5.2

Описание полей, методов и свойств класса RecipeShort.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| RecipeShort | public | Конструктор | - | | Конструктор для краткого рецепта без параметров. |
| RecipeShort | public | Конструктор | string title, Image image, string url | | Конструктор для краткого рецепта с параметрами. |
| RecipeShort | public | Конструктор | string title, Image image, string url, double indexPopularity | | Конструктор для краткого рецепта с параметрами. |
| Продолжение таблицы 5.2 | | | | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Title | public | String | get | Название рецепта. | |
| Image | public | Image | get | Изображение рецепта. | |
| Url | public | String | get | Ссылка на рецепт. | |
| IndexPopularity | public | Double | get | Индекс популярности. Больший индекс у первых рецептов в списке. | |

Таблица 5.3

Описание полей, методов и свойств класса Additional.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Additional | public | Конструктор | - | | Конструктор для дополнительной информации о рецепте без параметров. |
| Additional | public | Конструктор | string authorName, int countPortions, double prepMinutes, CPFC cpfc | | Конструктор для дополнительной информации о рецепте с параметрами. |
| Additional | public | Конструктор | string authorName, int countPortions, double prepMinutes,CPFC cpfc, string videoUrl | | Конструктор для дополнительной информации о рецепте с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| AuthorName | public | String | get | Имя автора. | |
| CountPortions | public | Int32 | get | Количество порций рецепта. | |
| PrepMinutes | public | Double | get | Количество минут на для приготовления блюда. | |
| VideoUrl | public | String | get | URL-адрес на видеоматериал. | |

Таблица 5.4

Описание полей, методов и свойств класса CPFC.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| CPFC | public | Конструктор | - | | Конструктор класса без параметров. |
| CPFC | public | Конструктор | double calories, double protein, double fats, double carbohydrates | | Конструктор класса с параметрами |
| **Свойства** | | | | | |
| Продолжение таблицы 5.4 | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Calories | public | Double | get | Количество калорий. | |
| Proteins | public | Double | get | Количество белка. | |
| Fats | public | Double | get | Количество жиров. | |
| Carbohydrates | public | Double | get | Количество углеводов. | |

Таблица 5.5

Описание полей, методов и свойств класса Image.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Image | public | Конструктор | - | Конструктор без параметров. |
| Image | public | Конструктор | string imageUrl | Конструктор с параметрами. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| ImageUrl | public | String | get | URL-адрес рецепта. |

Таблица 5.6

Описание полей, методов и свойств класса Ingredient.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Ingredient | public | Конструктор | - | | Конструктор без параметров. |
| Ingredient | public | Конструктор | string name, string unit, string recipeName | | Конструктор с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Name | public | String | get | Название рецепта. | |
| Unit | public | String | get | Количество рецепта. | |
| RecipeName | public | String | get | Название рецепта-родителя. | |

Таблица 5.7

Описание полей, методов и свойств класса StepRecipe.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| StepRecipe | public | Конструктор | - | | Конструктор без параметров. |
| StepRecipe | public | Конструктор | string description, Image image | | Конструктор с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Description | public | String | get | Описание шага. | |
| Продолжение таблицы 5.7 | | | | | |
| Image | public | Image | get | Изображение шага | |

Таблица 5.8

Описание полей, методов и свойств класса ServerMemoryCache.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| ServerMemoryCache | public | Конструктор | - | Конструктор без параметров. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Catche | public | MemoryCatche | get | Размер в Кб кэшируемых данных. |

Таблица 5.9

Описание полей, методов и свойств класса RecipeContext.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| RecipeContext | public | Конструктор | - | | Конструктор без параметров. |
| OnConfiguring | protected | void | DbContextOptionsBuilder optionsBuilder | | Метод для конфигурации настроек базы данных. |
| ByteToRecipe | public | RecipeFull | Byte[] arrBytes | | Десериализация из массива байтов в рецепт. |
| RecipeToByte | public | Byte[] | RecipeFull recipeFull | | Сериализация полного рецепта в массив байтов. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Recipes | public | DbSet<Recipe> | get | Полные рецепты, хранящиеся в базе данных. | |

Таблица 5.10

Описание полей, методов и свойств класса Recipe.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Recipe | public | Конструктор | - | | Конструктор без параметров. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Id | public | Guid | get, set | Полные рецепты, хранящиеся в базе данных. | |
| Продолжение таблицы 5.10 | | | | | |
| Url | public | String | get, set | URL-адрес рецепта. | |
| Date | Public | DateTime | get, set | Дата последнего обновления рецепта. | |
| RecipeFull | public | Byte[] | get, set | Сериализированный полный рецепт. | |

Таблица 5.11

Описание полей, методов и свойств класса PageController.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| \_logger | private | ILogger<PageController> | | Хранит экземпляр класса ILogger для дальнейшего логирования. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| PageController | public | Конструктор | ILogger<PageController> logger | Конструктор с параметрами. |
| GetPages | public | Task<IActionResult> | string section, int page = 1, string recipeName = null | Обработчик GET запроса для PageController. |
| LogTime | private | void | DateTime startTime | Логирование времени исполнения. |

Таблица 5.12

Описание полей, методов и свойств класса RecipeController.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| \_logger | private | ILogger<PageController> | | Хранит экземпляр класса ILogger для дальнейшего логирования. |
| diffHours | private | Int32 | | Константа, задающая максимальное количество часов для хранения рецепта в базе данны. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| RecipeController | public | Конструктор | ILogger< RecipeController> logger | Конструктор с параметрами. |
| GetPages | public | Task<IActionResult> | string section, int page = 1, string recipeName = null | Обработчик GET запроса для RecipeController. |
| LogTime | private | void | DateTime startTime | Логирование времени исполнения. |

Таблица 5.13

Описание полей, методов и свойств класса GetData.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Продолжение таблицы 5.13 | | | | |
| GetPage | public | Task<RecipeShort[]> | string section, int page, string findName = null | Метод, которые запускает парсеры для каждого сайта. |
| ParseRecipe | private | Task | ParserPage<RecipeShort[]> T, List<RecipeShort> recipeShorts | Асинхронный запуск парсера. |
| GetRecipe | public | Task<RecipeFull> | string url | Метод, который запускает парсер для полного рецепта. |

Таблица 5.14

Описание полей, методов и свойств класса ParserException.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| ParserException | public | Конструктор | - | Конструктор без параметров. |
| ParserException | public | Конструктор | string message | Конструктор с параметрами |
| ParserException | public | Конструктор | string message, Exception inner | Конструктор с параметрами |
| ParserException | protected | Конструктор | System.Runtime.Serialization.SerializationInfo info, System.Runtime.Serialization.StreamingContext context | Конструктор с параметрами |

Таблица 5.15

Описание полей, методов и свойств класса HtmlLoaderPage.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| \_client | private | HttpClient | | Хранит экземпляр класс HttpClient |
| \_url | private | String | | URL-адрес на страницу с рецептами. |
| \_settings | private | IParserPageSettings | | Настройки для парсинга сайта. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Продолжение таблицы 5.15 | | | | |
|  |  |  |  |  |
| HtmlLoaderPage | public | Конструктор | IParserPageSettings settings | Конструктор с параметрами. |
| GetSource | public | Task<string> | int idPage, string recipeName | Метод, получающий исходный код HTML страницы. |

Таблица 5.16

Описание полей, методов и свойств класса ParserPage.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | | Назначение | |
| \_parserSettings | private | IParserPageSettings | | Содержит настройки для парсинга сайта. | |
| \_loader | private | HtmlLoaderPage | | Хранит экземпляр класса HtmlLoaderPage | |
| Random | private | Random | | Хранит экземпляр класса Random | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| ParserPage | private | Конструктор | IParserPageSettings settings | | Конструктор с параметрами. |
| ParserPage | internal | Конструктор | int idPage, string recipeName | | Конструктор с параметрами. |
| GetPageId | private | Int32 | int maxPage | | Метод, генерирующий случайный номер страницы |
| Worker | internal | Task<T> | - | | Метод, который начинает сбор информации и последующий парсинг списка рецептов. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | | Назначение |
| Parser | private | IParserPage<T> | get | | Хранит экземпляр парсера. |
| Settings | private | IParserPageSettings | get, set | | Хранит экземпляр настроек для парсера. |

Таблица 5.17

Описание полей, методов и свойств класса EdaPageParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeShort[] | IHtmlDocument document, IParserPageSettings settings | Парсит полученный исходный код страницы и собирает объекты типа RecipeShort в одну коллекцию. |

Таблица 5.18

Описание полей, методов и свойств класса EdimDomaPageParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeShort[] | IHtmlDocument document, IParserPageSettings settings | Парсит полученный исходный код страницы и собирает объекты типа RecipeShort в одну коллекцию. |

Таблица 5.19

Описание полей, методов и свойств класса PovarenokPageParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeShort[] | IHtmlDocument document, IParserPageSettings settings | Парсит полученный исходный код страницы и собирает объекты типа RecipeShort в одну коллекцию. |

Таблица 5.20

Описание полей, методов и свойств класса PovarPageParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeShort[] | IHtmlDocument document, IParserPageSettings settings | Парсит полученный исходный код страницы и собирает объекты типа RecipeShort в одну коллекцию. |

Таблица 5.21

Описание полей, методов и свойств класса EdaPageSettings.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| EdaPageSettings | private | Конструктор | string section, int pageId | | Конструктор без параметров. |
| EdaPageSettings | public | Конструктор | string section, int pageId, string recipeName | | Конструктор с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Url | public | String | get | URL-адрес сайта. | |
| Section | public | String | get | Раздел, который нужно парсить. | |
| Sections | public | Dictionary<string,string> | get | Словарь, где ключи разделы, а их значения – URL-адреса на разделы. | |
| MaxPageId | public | Int32 | get | Максимальная страница на сайте. Используется для генерации случайной страницы. | |
| PageId | public | Int32 | get | Хранит в себе номер страницы. | |
|  |  |  |  |  | |
| Продолжение таблицы 5.21 | | | | | |
| IndexPopularity | public | Double | get, set | Индекс популярности рецепта, по которому сортируются полученные в конце рецепты. | |
| IndexStep | public | Double | get | Шаг арифметической прогрессии для снижения индекса популярности. | |
| RecipeName | public | String | get | Название рецепта для поиска. | |

Таблица 5.22

Описание полей, методов и свойств класса EdimDomaPageSettings.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| EdimDomaPageSettings | private | Конструктор | string section, int pageId | | Конструктор без параметров. |
| EdimDomaPageSettings | public | Конструктор | string section, int pageId, string recipeName | | Конструктор с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Url | public | String | get | URL-адрес сайта. | |
| Section | public | String | get | Раздел, который нужно парсить. | |
| Sections | public | Dictionary<string,string> | get | Словарь, где ключи разделы, а их значения – URL-адреса на разделы. | |
| MaxPageId | public | Int32 | get | Максимальная страница на сайте. Используется для генерации случайной страницы. | |
| PageId | public | Int32 | get | Хранит в себе номер страницы. | |
| IndexPopularity | public | Double | get, set | Индекс популярности рецепта, по которому сортируются полученные в конце рецепты. | |
| IndexStep | public | Double | get | Шаг арифметической прогрессии для снижения индекса популярности. | |
| RecipeName | public | String | get | Название рецепта для поиска. | |

Таблица 5.23

Описание полей, методов и свойств класса PovarPageSettings.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| PovarPageSettings | private | Конструктор | string section, int pageId | | Конструктор без параметров. |
| PovarPageSettings | public | Конструктор | string section, int pageId, string recipeName | | Конструктор с параметрами. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Продолжение таблицы 5.13 | | | | | |
| Url | public | String | get | URL-адрес сайта. | |
| Section | public | String | get | Раздел, который нужно парсить. | |
| Sections | public | Dictionary<string,string> | get | Словарь, где ключи разделы, а их значения – URL-адреса на разделы. | |
| MaxPageId | public | Int32 | get | Максимальная страница на сайте. Используется для генерации случайной страницы. | |
| PageId | public | Int32 | get | Хранит в себе номер страницы. | |
| IndexPopularity | public | Double | get, set | Индекс популярности рецепта, по которому сортируются полученные в конце рецепты. | |
| IndexStep | public | Double | get | Шаг арифметической прогрессии для снижения индекса популярности. | |
| RecipeName | public | String | get | Название рецепта для поиска. | |

Таблица 5.24

Описание полей, методов и свойств класса PovarenokPageSettings.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Методы** | | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | | Назначение |
| PovarenokPageSettings | private | Конструктор | | string section, int pageId | | Конструктор без параметров. |
| PovarenokPageSettings | public | Конструктор | | string section, int pageId, string recipeName | | Конструктор с параметрами. |
| **Свойства** | | | | | | |
| Имя | Модификатор доступа | Тип | | Доступ | Назначение | |
| Url | public | String | | get | URL-адрес сайта. | |
| Section | public | String | | get | Раздел, который нужно парсить. | |
| Sections | public | Dictionary<string,string> | | get | Словарь, где ключи разделы, а их значения – URL-адреса на разделы. | |
| MaxPageId | public | Int32 | | get | Максимальная страница на сайте. Используется для генерации случайной страницы. | |
| PageId | public | Int32 | | get | Хранит в себе номер страницы. | |
| IndexPopularity | public | Double | | get, set | Индекс популярности рецепта, по которому сортируются полученные в конце рецепты. | |
| IndexStep | public | Double | | get | Шаг арифметической прогрессии для снижения индекса популярности. | |
| RecipeName | public | String | | get | Название рецепта для поиска. | |
| **Методы** | | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение | |
| GetUrl | private | String | String recipeName | | Метод, который меняет кодировку русских символов на нужную. | |

Таблица 5.25

Описание полей, методов и свойств класса HtmlLoaderRecipe.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| \_client | private | HttpClient | | Хранит экземпляр класса HttpClient |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| HtmlLoaderRecipe | public | Конструктор | IParserRecipeSettings settings | Конструктор с параметрами. |
| GetSource | public | Task<string> | string url | Метод, получающий исходный код HTML страницы. |

Таблица 5.26

Описание полей, методов и свойств класса ParserRecipe.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | | Назначение | |
| parserSettings | private | IParserRecipeSettings | | Содержит настройки для парсинга сайта. | |
| loader | private | HtmlLoaderRecipe | | Хранит экземпляр класса HtmlLoaderRecipe | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| ParserRecipe | private | Конструктор | IParserRecipe<T> parser | | Конструктор с параметрами. |
| ParserRecipe | public | Конструктор | IParserRecipe<T> parser, IParserRecipeSettings parserSettings | | Конструктор с параметрами. |
| Worker | internal | Task<T> | - | | Метод, который начинает сбор информации и последующий парсинг списка рецептов. |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | | Назначение |
| Parser | private | IParserPage<T> | get | | Хранит экземпляр парсера. |
| Settings | private | IParserPageSettings | get, set | | Хранит экземпляр настроек для парсера. |

Таблица 5.27

Описание полей, методов и свойств класса EdaRecipeParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeFull | IHtmlDocument document, IParserRecipeSettings parserRecipeSettings | Метод парсит исходный код страницы и собирает из данных объект типа RecipeFull |
| ConvertToMinutes | public | Double | string inputLine | Метод преобразует входящую строку и забирает из неё информацию о количестве минут приготовления блюда. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | private | String | get, set | Хранит URL-адрес на рецепт. |
| Title | private | String | get, set | Хранит название рецепта. |
| TitleImage | private | Image | get, set | Хранит изображение рецепта. |
| Description | private | String | get, set | Хранит описание рецепта. |
| Ingredients | private | Inregredients[] | get, set | Хранит коллекцию ингредиентов. |
| StepsRecipe | private | StepRecipe[] | get, set | Хранит коллекцию шагов. |
| Additional | private | Additional | get, set | Хранит дополнительное описание. |

Таблица 5.28

Описание полей, методов и свойств класса EdimDomaRecipeParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeFull | IHtmlDocument document, IParserRecipeSettings parserRecipeSettings | Метод парсит исходный код страницы и собирает из данных объект типа RecipeFull |
| ConvertToMinutes | public | Double | string inputLine | Метод преобразует входящую строку и забирает из неё информацию о количестве минут приготовления блюда. |
| **Свойства** | | | | |
| Продолжение таблицы 5.28 | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | private | String | get, set | Хранит URL-адрес на рецепт. |
| Title | private | String | get, set | Хранит название рецепта. |
| TitleImage | private | Image | get, set | Хранит изображение рецепта. |
| Description | private | String | get, set | Хранит описание рецепта. |
| Ingredients | private | Inregredients[] | get, set | Хранит коллекцию ингредиентов. |
| StepsRecipe | private | StepRecipe[] | get, set | Хранит коллекцию шагов. |
| Additional | private | Additional | get, set | Хранит дополнительное описание. |

Таблица 5.29

Описание полей, методов и свойств класса PovarenokRecipeParser.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Parse | public | RecipeFull | IHtmlDocument document, IParserRecipeSettings parserRecipeSettings | Метод парсит исходный код страницы и собирает из данных объект типа RecipeFull |
| ConvertToMinutes | public | Double | string inputLine | Метод преобразует входящую строку и забирает из неё информацию о количестве минут приготовления блюда. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | private | String | get, set | Хранит URL-адрес на рецепт. |
| Title | private | String | get, set | Хранит название рецепта. |
| TitleImage | private | Image | get, set | Хранит изображение рецепта. |
| Description | private | String | get, set | Хранит описание рецепта. |
| Ingredients | private | Inregredients[] | get, set | Хранит коллекцию ингредиентов. |
| StepsRecipe | private | StepRecipe[] | get, set | Хранит коллекцию шагов. |
| Additional | private | Additional | get, set | Хранит дополнительное описание. |

Таблица 5.30

Описание полей, методов и свойств класса EdaRecipeSettings.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| EdaRecipeSettings | public | Конструктор | String url | Конструктор с параметрами |
| **Свойства** | | | | |
| Продолжение таблицы 5.30 | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | public | String | get | Хранит URL-адрес на рецепт. |

Таблица 5.31

Описание полей, методов и свойств класса PovarRecipeSettings.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| PovarRecipeSettings | public | Конструктор | String url | Конструктор с параметрами |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | public | String | get | Хранит URL-адрес на рецепт. |

Таблица 5.32

Описание полей, методов и свойств класса PovarenokRecipeSettings.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| PovarenokRecipeSettings | public | Конструктор | String url | Конструктор с параметрами |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | public | String | get | Хранит URL-адрес на рецепт. |

Таблица 5.33

Описание полей, методов и свойств класса EdimDomaRecipeSettings.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| EdimDomaRecipeSettings | public | Конструктор | String url | Конструктор с параметрами |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Url | public | String | get | Хранит URL-адрес на рецепт. |

Таблица 5.34

Описание полей, методов и свойств класса IngredientTable.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| IngredientTable | public | Конструктор | - | Конструктор без параметров. |
| IngredientTable | public | Конструктор | string name, string unit, string recipeName | Конструктор с параметрами. |
| Продолжение таблицы 5.34 | | | | |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Id | public | Int32 | get, set | Индивидуальный номер в базе данных. |
| Name | public | String | get | Название ингредиента. |
| Unit | public | String | get | Мера измерения и количество ингрединта. |
| RecipeName | public | String | get | Название рецепта-родителя. |

Таблица 5.35

Описание полей, методов и свойств класса RecipeTable.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| RecipeTable | public | Конструктор | - | Конструктор без параметров. |
| RecipeTable | public | Конструктор | string name, byte[] recipe | Конструктор с параметрами. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Id | public | Int32 | get, set | Индивидуальный номер в базе данных. |
| Name | public | String | get | Название рецепта. |
| Recipe | public | Byte[] | get | Сериализированный объект полного рецепта в массив байтов. |

Таблица 5.36

Описание полей, методов и свойств класса DataContext.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| ByteArrayToObject<T> | public | T | Byte[] arrBytes | Метод, который десериализирует массив байтов в объект RecipeFull. |
| RecipeToByteArray | public | Byte[] | Object obj | Метод, который сериализирует объект в массив байтов. |

Таблица 5.37

Описание полей, методов и свойств класса HttpContext.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| ipAdress | private | String | | Хранит значение адреса сервера. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| GetSource | private | Task<string> | String query | Метод обращается к серверу с помощью HTTP-GET запроса и возвращает ответ сервера. |
| GetPages | public | List<RecipeShort> | String query | Десериализирует ответ сервера в коллекцию из элементов типа RecipeShort. |
| GetRecipe | public | RecipeFull | String url | Десериализирует овтет сервер в объект типа RecipeFull. |

Таблица 5.38

Описание полей, методов и свойств класса IngredientData.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| dbPath | private | String | | Хранит путь к базе данных. |
| dbConnection | private | SqLiteConnection | | Содержит экземпляр класса SqLiteConnection |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| IngredientData | public | Конструктор | - | Статический конструктор. Проверяет, создана ли база данных, если нет – создаёт её. |
| GetArrayIngredients | public | List<Ingredient> | - | Метод, которые выдаёт все ингредиенты из базы данных. |
| ExistsIngredient | public | bool | Ingredient ingredient | Метод, который указывает, содержится ли ингредиент в базе данных или нет. |
| DeleteIngredient | public | void | Ingredient ingredient | Метод, который удаляет входящий в него ингредиент. |
| SaveIngredient | public | void | Ingredient ingredient | Метод, который сохраняет входящий в него ингредиент. |

Таблица 5.39

Описание полей, методов и свойств класса RecipeData.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| dbPath | private | String | | Хранит путь к базе данных. |
| dbConnection | private | SqLiteConnection | | Содержит экземпляр класса SqLiteConnection |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| RecipeData | public | Конструктор | - | Статический конструктор. Проверяет, создана ли база данных, если нет – создаёт её. |
| GetArrayRecipes | public | RecipeShort[] | - | Метод, которые выдаёт все краткие рецепты из базы данных. |
| ExistsRecipe | public | bool | String url | Метод, который указывает, содержится ли данный краткий рецепт в базе данных или нет. |
| DeleteRecipe | public | void | String url | Метод, который удаляет входящий в него краткий рецепт. |
| SaveRecipe | public | void | RecipeShort recipeShort | Метод, который сохраняет входящий в него краткий рецепт. |

Таблица 5.40

Описание полей, методов и свойств класса InfoActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| Продолжение таблицы 5.40 | | | | |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для кнопки «Назад», возвращает пользователя к предыдущей активности. |

Таблица 5.41

Описание полей, методов и свойств класса MainActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| recyclerView | private | RecyclerView | | Содержит ссылку на элемент типа RecyclerView, который представляет список краткий рецептов. |
| recipeAdapter | private | RecipeAdapter | | Содержит ссылку на элемент типа RecipeAdapter, который отвечает за поведение элементов списка краткий рецептов. |
| drawer | private | DrawerLayout | | Содержит ссылку на элемент типа DrawerLayout, которые отвечает за отрисовку бокового меню. |
| swipeRefreshLayout | private | SwipeRefreshLayout | | Содержит ссылку на элемент типа SwipeRefreshLayout, который отвечает за обновление списка краткий рецептов. |
| linearLayoutManager | private | LinearLayoutManager | | Содержит ссылку на элемент типа LinearLayoutManager. |
| navigationView | private | NavigationView | | Содержит ссылку на элемент типа NavigationView, который отвечает за отрисовку бокового меню. |
| buttonMenu | private | Button | | Содержит ссылку на кнопку, которая отвечает за открытие бокового меню. |
| recipeListener | private | RecipeScrollListener | | Содержит ссылку на экземпляр класса RecipeScrollListener, который требуется для прослушки скроллинга списка рецептов пользователем, чтобы вовремя подгружать новые элементы в список. |
| Продолжение таблицы 5.41 | | | | |
| editText | private | EditText | | Содержит ссылку на экземпляр класса EditText, который является полем для ввода рецепта. |
| spinner | private | Spinner | | Содержит ссылку на элемент типа Spinner, который предоставляет выдвижное меню из фильтров. |
| spinnerAdapter | private | ArrayAdapter | | Содержит ссылку на экземпляр класса ArrayAdapter, который требуется для инициализации элементов выдвижного списка. |
| actionBarDrawerToggle | private | ActionBarDrawerToggle | | Содержит ссылку на экземпляр класса ActionBarDrawerToggle, которые требуется для отрисовки бокового меню. |
| findButton | private | Button | | Содержит ссылку на экземпляр класса Button, который отвечает за кнопку поиска рецептов. |
| recipeShorts | private | List<RecipeShort> | | Коллекция рецептов, отображаемых в списке. |
| page | private | Int32 | | Номер страницы. |
| lastQuery | private | String | | Последний запрос к серверу. |
| messageErrorTitle | private | String | | Заголовок сообщения при ошибке в диалогом окне. |
| messageErrorText | private | String | | Текст сообщения при ошибке в диалоговом окне. |
| messageErrorTextButton | private | String | | Текст кнопки в диалогом окне при ошибке. |
| errorDialog | private | Android.Support.V7.App.AlertDialog.Builder | | Экземпляр класса Android.Support.V7.App.AlertDialog.Builder |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для закрытия бокового меню. |
| Продолжение таблицы 5.41 | | | | |
| OnClickFind | private | void | object sender, EventArgs args | Метод, который отвечает за поиск по нажатию на кнопку поиска. |
| FindByRecipeName | private | void | object sender, View.KeyEventArgs e | Метод, который отвечает за поиск рецепта по нажатию кнопки на клавиатуре пользователя. |
| LoadMoreElements | private | void | object sender, EventArgs args | Метод, который подгружает уже в существующий список рецептов новые рецепты. |
| RefreshLayout | private | void | object sender, EventArgs args | Метод, который отвечает за обновление списка рецептов. Вызывает UpdateListView. |
| SetMenuButtonClick | private | void | object sender, EventArgs args | Метод, в котором прописана логика нажатия на кнопку для вызова бокового меню. |
| UpdateListView | private | void | string query = "section=new", List<RecipeShort> recipeShorts = null | Метод, который отвечает за обновление списка рецептов. |
| OnCreateOptionsMenu | public | void | IMenu menu | Метод, который инициализирует настройки бокового меню. |
| OnRecipeClick | private | void | object sender, int position | Метод, который отвечает за логику нажатия на рецепт. |
| SelectedItemSpinner | private | void | object sender, AdapterView.ItemSelectedEventArgs e | Метод, которые фильтрует элементы списка рецептов. |
| UpdateCollectionRecipes | private | Task<List<RecipeShort>> | string query | Метод, который загружает новый список рецептов по запросу. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| OnNavigationItemSelected | public | void | IMenuItem menuItem | Метод, в котором прописана логика выбора элемента из бокового меню. |
| CloseDrawer | public | DrawerLayout | DrawerLayout drawerLayout | Метод, в котором прописана логика закрытия бокового меню. |

Таблица 5.42

Описание полей, методов и свойств класса PlatesActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| recyclerView | private | RecyclerView | | Хранит ссылку на экземпляр RecyclerView, который отображает список разделов. |
| linearLayoutManager | private | LinearLayoutManager | | Содержит ссылку на элемент типа LinearLayoutManager. |
| plates | private | List<Plate> | | Коллекция разделов. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для кнопки «Назад», возвращает пользователя к предыдущей активности. |
| OnPlatePressed | private | void | object sender, int position | Метод, в котором прописана логика нажатия на раздел. |

Таблица 5.43

Описание полей, методов и свойств класса SavedIngredientsActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| recyclerView | private | RecyclerView | | Хранит ссылку на экземпляр RecyclerView, который отображает список разделов. |
| linearLayoutManager | private | LinearLayoutManager | | Содержит ссылку на элемент типа LinearLayoutManager. |
| ingredients | private | List<Ingredient> | | Коллекция ингредиентов. |
| **Методы** | | | | |
| Продолжение таблицы 5.43 | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для кнопки «Назад», возвращает пользователя к предыдущей активности. |

Таблица 5.44

Описание полей, методов и свойств класса SavedRecipesActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| recyclerView | private | RecyclerView | | Хранит ссылку на экземпляр RecyclerView, который отображает список разделов. |
| linearLayoutManager | private | LinearLayoutManager | | Содержит ссылку на элемент типа LinearLayoutManager. |
| recipeShorts | private | List<RecipeShort> | | Коллекция рецептов. |
| \_adapter | private | RecipeAdapter | | Экземпляр класса RecipeAdapter, который требуется для инициализации элементов списка. |
| \_searchView | private | Android.Support.V7.Widget.SearchView | | Экземпляр класса SearchView, который требуется для поиска элементов в списке. |
| dataRecipes | private | List<RecipeShort> | | Коллекция рецептов, полученная из базы данных. |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| Продолжение таблицы 5.44 | | | | |
| OnItemClick | private | void | object sender, int position | Метод, в котором прописана логика взаимодействия с элементом списка. |

Таблица 5.45

Описание полей, методов и свойств класса RecipeActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| urlRecipe | private | String | | URL-адрес рецепта |
| recipeFull | private | RecipeFull | | Полный рецепт. |
| recipeShort | private | RecipeShort | | Краткий рецепт. |
| buttonBack | private | Button | | Кнопка «Назад» |
| buttonStar | private | FloatingActionButton | | Кнопка, чтобы добавить/убрать рецепт из «Избранное» |
| buttonShare | private | Button | | Кнопка, чтобы поделиться рецептом. |
| buttonWeb | private | Button | | Кнопка, чтобы перейти к источнику. |
| relativeLayoutBack | private | RelativeLayout | | Кнопка «Назад» |
| title | private | TextView | | Заголовок рецепта. |
| viewPager | private | ViewPager | | Элемент для отображения фрагментов внутри себя. |
| messageErrorTitle | private | String | | Заголовок сообщения при ошибке в диалогом окне. |
| messageErrorText | private | String | | Текст сообщения при ошибке в диалоговом окне. |
| messageErrorTextButton | private | String | | Текст кнопки в диалогом окне при ошибке. |
| errorDialog | private | Android.Support.V7.App.AlertDialog.Builder | | Экземпляр класса Android.Support.V7.App.AlertDialog.Builder |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| Продолжение таблицы 5.45 | | | | |
| OnStart | protected override | void | - | Метод непосредственно перед тем, как активность становится видимой пользователю. |
| OnStartWeb | private | void | (object sender, EventArgs args | Метод отвечает за открытие браузера при нажатии кнопки «Открыть источник». |
| OnAddFavRecipe | private | void | object sender, EventArgs args | Метод, в котором прописана логика добавление рецепта в раздел «Изобранное». |
| OnShareButton | private | void |  | Метод, отвечающий за логику кнопки «Поделиться» |
| LoadRecipe | private | Task<RecipeFull> |  | Метод, отвечающий за логику загрузки рецепта. |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для кнопки «Назад», возвращает пользователя к предыдущей активности. |

Таблица 5.46

Описание полей, методов и свойств класса SectionRecipesActivity.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | | Назначение |
| recipeShorts | private | List<RecipeShort> | | Коллекция краткий рецептов. |
| swipeRefreshLayout | private | SwipeRefreshLayout | | Экземпляр класса SwipeRefreshLayout. |
| recyclerView | private | RecyclerView | | Экземпляр класса RecyclerView для отображения списка рецептов. |
| recipeAdapter | private | RecipeAdapter | | Адаптер для списка краткий рецептов. |
| linearLayoutManager | private | LinearLayoutManager | | Экземпляр класса LinearLayoutManager, используется для инициализации адапатера для списка. |
| Продолжение таблицы 5.46 | | | | |
| recipeListener | private | RecipeScrollListener | | Слушатель для обновления списка рецептов, по их окончанию. |
| relativeLayoutBack | private | RelativeLayout | | Элемент, в котором размещает кнопка «Назад». |
| buttonBack | private | Button | | Кнопка «Назад». |
| textViewTitle | private | TextView | | Заголовок списка. |
| title | private | String | | Название раздела. |
| lastQuery | private | String | | Последний запрос. |
| page | private | Int32 | | Номер текущей страницы. |
| section | private | String | | Текущий раздел. |
| messageErrorTitle | private | String | | Заголовок сообщения при ошибке в диалогом окне. |
| messageErrorText | private | String | | Текст сообщения при ошибке в диалоговом окне. |
| messageErrorTextButton | private | String | | Текст кнопки в диалогом окне при ошибке. |
| errorDialog | private | Android.Support.V7.App.AlertDialog.Builder | | Экземпляр класса Android.Support.V7.App.AlertDialog.Builder |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | protected override | void | Bundle savedInstanceState | Метод, который задаёт начальную установку параметров при инициализации активности. |
| OnRequestPemissionsResult | private | void | int requestCode, string[] permissions, [GeneratedEnum] Android.Content.PM.Permission[] grantResults | Метод, который вызывает диалоговое окно при отсутствии какие-либо прав, не предоставленных пользователем приложению, с просьбой предоставить права. |
| OnBackPressed | public override | void | object sender, EventArgs args | Метод для кнопки «Назад», возвращает пользователя к предыдущей активности. |
| LoadMoreElements | private | void | object sender, EventArgs e | Метод для загрузки дополнительных рецептов. |
| RefreshLayout | private | void | object sender, EventArgs e | Метод, в котором содержится логика обновления страницы. |
| UpdateListView | private | void | string query = "section={section}", List<RecipeShort> recipeShorts = null | Метод, в котором описана логика обновления списка. |
| Продолжение таблицы 5.46 | | | | |
| UpdateCollectionRecipes | private | Task<List<RecipeShort>> | string query | Метод для обновления текущей коллекции краткий рецептов. |
| OnRecipeClick | private | void | object sender, EventArgs e | Метод, в котором прописана логика нажатия на рецепт. |

Таблица 5.47

Описание полей, методов и свойств класса IngredientsAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| \_context | private | Context | Объект, который предоставляет доступ к базовым функциям. | |
| \_ingredients | private | Ingredient[] | Коллекция ингредиентов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| IngredientsAdapter | public | Конструктор | Context context, RecipeFull recipeFull | Конструктор с параметрами. |
| GetItemId | public override | Int64 | int position | Метод, который возвращает ID элемента по его индексу. |
| GetView | public override | View | int position, View convertView, ViewGroup parent | Метод, в котором происходит сборка компонента View. |
| Индексатор | public override | RecipeFull | int position | Возвращает экземпляр полного рецепта. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Count | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.48

Описание полей, методов и свойств класса PagerRecipeAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| pagesName | private | string[] | Коллекция из названий заголовков. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| PagerRecipeAdapter | public | Конструктор | FragmentManager fm, RecipeFull recipeFull | Конструктор с параметрами. |
| GetItem | public override | Fragment | int position | Метод выдаёт по индексу элемент класса Framgent. |
| GetPageTitleFormatted | public override | ICharSequence | int position | Метод выдаёт название заголовка для Frament под текущей позицией. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Count | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.49

Описание полей, методов и свойств класса PlateAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_items | private | List<Plate> | Коллекция из плиток. | |
| \_activity | private | Activity | Activity, где размещается список. | |
| ItemClick | public | event EventHandler<int> | Событие нажатия на элемент. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| PlateAdapter | public | Конструктор | List<Plate> plates, Activity activity | Конструктор с параметрами. |
| OnBindViewHolder | public override | void | ecyclerView.ViewHolder holder, int position | Метод, для инициализации View. |
| Продолжение таблицы 5.49 | | | | |
| OnCreateViewHolder | public override | RecyclerView.ViewHolder | ViewGroup parent, int viewType | Метод, для создания View. |
| OnClick | public override | void | int position | Метод, в котором описана логика нажатия на элемент списка. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| ItemCount | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.50

Описание полей, методов и свойств класса RecipeAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_items | private | List< RecipeShort> | Коллекция из кратких рецептов. | |
| \_activity | private | Activity | Activity, где размещается список. | |
| ItemClick | public | event EventHandler<int> | Событие нажатия на элемент. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| RecipeAdapter | public | Конструктор | List<RecipeShort> recipeShorts, Activity activity | Конструктор с параметрами. |
| OnBindViewHolder | public override | void | ecyclerView.ViewHolder holder, int position | Метод, для инициализации View. |
| OnCreateViewHolder | public override | RecyclerView.ViewHolder | ViewGroup parent, int viewType | Метод, для создания View. |
| OnClick | public override | void | int position | Метод, в котором описана логика нажатия на элемент списка. |
| AddItems | public | void | List<RecipeShort> recipes | Добавление в коллекцию новых элементов и обновление текущего списка. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| ItemCount | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.51

Описание полей, методов и свойств класса SavedIngredientsAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_items | private | List< Ingredient> | Коллекция из ингредиентов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| SavedIngredientsAdapter | public | Конструктор | List<RecipeShort> recipeShorts, Activity activity | Конструктор с параметрами. |
| OnBindViewHolder | public override | void | ecyclerView.ViewHolder holder, int position | Метод, для инициализации View. |
| OnCreateViewHolder | public override | RecyclerView.ViewHolder | ViewGroup parent, int viewType | Метод, для создания View. |
| RemoveItem | public | void | int position | Удаление из списка элемента под заданной позицией. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| ItemCount | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.52

Описание полей, методов и свойств класса StepAdapter.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| \_context | private | Context | Объект, который предоставляет доступ к базовым функциям. | |
| \_stepRecipeBoxes | private | List<StepRecipe> | Коллекция шагов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| StepAdapter | public | Конструктор | Context context, RecipeFull recipeFull | Конструктор с параметрами. |
| GetItemId | public override | Int64 | int position | Метод, который возвращает ID элемента по его индексу. |
| GetView | public override | View | int position, View convertView, ViewGroup parent | Метод, в котором происходит сборка компонента View. |
| Индексатор | public override | RecipeFull | int position | Возвращает экземпляр полного рецепта. |
| **Свойства** | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение |
| Count | public override | Int32 | get | Свойство возвращает количество элементов в списке. |

Таблица 5.53

Описание полей, методов и свойств класса RecipeDescriptionFragment.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| \_image | private | ImageView | Изображение рецепта. | |
| \_description | private | TextView | Описание рецепта. | |
| \_cpfcRecipe | private | TextView | Описание о содержании калорий и БЖУ. | |
| \_authorNameRecipe | private | TextView | Имя автора. | |
| \_additionalInfoRecipe | private | TextView | Дополнительная информация о рецепте. | |
| \_urlRecipe | private | TextView | URL-ссылка на рецепт. | |
| Продолжение таблицы 5.53 | | | | |
| \_videoView | private | TextView | URL-ссылка на видеоконтент. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | public override | void | Bundle savedInstanceState | Метод, с которого начинается выполнение Activity |
| OnCreateView | public override | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) | Метод, вызывается для создания компонентов внутри фрагмента. |
| OnActivityCreated | public override | void | Bundle savedInstanceState | Метод, вызывается, когда отработает метод активности OnCreate(). |

Таблица 5.54

Описание полей, методов и свойств класса RecipeIngredientsFragment.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| \_listIngredients | private | ListView | Коллекция ингредиентов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | public override | void | Bundle savedInstanceState | Метод, с которого начинается выполнение Activity |
| OnCreateView | public override | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) | Метод, вызывается для создания компонентов внутри фрагмента. |
| OnActivityCreated | public override | void | Bundle savedInstanceState | Метод, вызывается, когда отработает метод активности OnCreate(). |

Таблица 5.55

Описание полей, методов и свойств класса RecipeStepsFragment.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_recipeFull | private | RecipeFull | Экземпляр текущего рецепта. | |
| \_listSteps | private | ListView | Коллекция шагов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| OnCreate | public override | void | Bundle savedInstanceState | Метод, с которого начинается выполнение Activity |
| OnCreateView | public override | View | LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) | Метод, вызывается для создания компонентов внутри фрагмента. |
| OnActivityCreated | public override | void | Bundle savedInstanceState | Метод, вызывается, когда отработает метод активности OnCreate(). |

Таблица 5.56

Описание полей, методов и свойств класса Plate.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Title | public | string | get | Заголовок плитки. | |
| Background | public | string | get | Адрес на изображение. | |
| Key | public | string | get | Ключ, который содержит индивидуальное название для определения раздела сервером. | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Plate | public | Конструктор | string title, string background, string key | | Конструктор без параметров |

Таблица 5.57

Описание полей, методов и свойств класса RecipeScrollListener.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Поля** | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| layoutManager | private | LinearLayoutManager | Экземпляр типа LinearLayoutManager, требуется для определения местоположения списка. | |
| LoadMoreEvent | public | EventHandler | События для загрузки новых элементов. | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| RecipeScrollListener | public | Конструктор | LinearLayoutManager layoutManager | Конструктор без параметров. |
| OnScrolled | public override | void | RecyclerView recyclerView, int dx, int dy | Метод, в котором определена логика вызова события при скроллинге списка. |

Таблица 5.58

Описание полей, методов и свойств класса PlateViewHolder.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | | Назначение |
| Background | public | RelativeLayout | get | | Фоновое изображение |
| Title | public | TextView | get | | Название раздела |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение | |
| PlateViewHolder | public | Конструктор | View itemView, Action<int> listener | Конструктор без параметров | |

Таблица 5.59

Описание полей, методов и свойств класса RecipeViewHolder.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | | Назначение |
| Image | public | RelativeLayout | get | | Изображение краткого рецепта. |
| Title | public | TextView | get | | Название рецепта. |
| Link | public | TextView | get | | Название источника. |
| **Методы** | | | | | |
| Продолжение таблицы 5.59 | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение | |
| RecipeViewHolder | public | Конструктор | View itemView, Action<int> listener | Конструктор без параметров | |

Таблица 5.60

Описание полей, методов и свойств класса SavedIngredientsViewHolder.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| Title | public | TextView | get | Название ингредиента. | |
| Unit | public | TextView | get | Количество и мера измерения ингредиента. | |
| RecipeName | public | TextView | get | Название рецепта ингредиента. | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| SavedIngredientsViewHolder | public | Конструктор | View itemView, Action<int> listener | | Конструктор без параметров |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулирован |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |