

# FEATURES OF THE IMPLEMENTATION OF TIME MANAGEMENT TECHNIQUES IN THE PROGRAMMING LANGUAGE DART

Yakov Lvovich, Emma Lvovich

## **Abstract:**

---

*The purpose of this work is to develop a prototype of a mobile application, which is a tool for personal time management, in the programming language Dart using the Flutter framework. In the course of the work, the research of the subject area of time management techniques was carried out, according to the results of which the methodology for implementation was chosen. Also, the research of the subject area and the features of software and language solutions were analyzed for the implementation of the prototype. Application development was carried out in the high-level language Dart (SDK) using the Flutter Framework (SDK) in the integrated development environment Android Studio. The prototype mobile application is designed for a user with minimal skills on Android devices. The application has an intuitive ergonomic interface, it contains a timer tool, a description of the tool, allowing you to use the prototype as a tool for personal time management. The mobile application is developed, the user interface is designed, the information system is developed, algorithms for drawing up personal statistics are developed.*

## **Keywords:**

*Computer application, online store, Android, user.*

## **ACM Computing Classification System**

*Distributed systems organizing principles, software functional properties, specialized application languages*

## **Introduction**

Currently, information technology is an integral part of our lives. Modern man can not imagine his day without using a smartphone, tablet or smart watch. Therefore, the development of various mobile applications to provide users with comfortable operation of their devices is in demand [1].

Such demand creates a demand for programmers and a demand for new, simpler, more convenient and accessible tools for development - languages and environments.

To use the time resource most effectively, both for personal purposes and for work, you need to use time management.

Time management is a set of knowledge, skills and abilities, thanks to which a person is able to prioritize, accurately plan his time, thereby increasing his personal productivity in the organization of his working time. Time management includes many different techniques.

From the analyzed techniques of time management, the method of "Pomodoro" was chosen for implementation. It is simple, but it provides more control than the techniques with sorting tasks by type and importance - Eisenhower Matrix or Franklin Pyramid. It is also possible to use this technique for more effective application of such time management method as "Eat a frog first".

Among other things, statistics on work with such a tool will be easier to implement - you do not need to assess the effectiveness of the work done, because of the short-term method, statistics are generated in a short time, which makes it easy to analyze the effectiveness of the methodology and indicate progress.

Currently, there are many different applications-tools for time management technique "Pomodoro". These applications provide all the classic features of this technique, have a simple intuitive interface, some have the ability to configure the time intervals for work and rest, change the parameters of sound notifications. These solutions are mostly:

- a) designed only for use on one of the platforms - either Android or IOS [2];
- b) they do not have Russian localization;
- c) they are paid or shareware with the presence of advertising banners.

In this paper we consider the development of an application for time management on the basis of the language Dart.

## 1 Technologies Used

When implementing the application, the following technologies were used:

- Dart programming language (SDK version 3.2.0);
- Flutter framework (SDK version 1.5.4);
- Integrated Development Environment Android Studio (version 3.4);
- SQLite, a plug-in library (sqflite version 3) for creating a local database;
- Sourcetree - program for creating backup copies of working code with the ability to roll back to previous working versions;
- draw.io - to create a visual database structure, build the architecture of a software system, edit screen forms.

## 2 Dart Programming Language

Dart is a cross-platform, object-oriented, scripted, multi-paradigm functional programming language.

Google offered Dart at the end of 2011 as a language for web development, devoid of JavaScript flaws, which are associated, in particular, with the inconvenience of structuring programs when their size goes beyond simple web applications. The main goal of this initiative was to create the most convenient programming environment specifically designed for the development of web content [3].

Currently, Dart is used to create both simple scripts and full-featured applications. In addition to web development with it, you can create mobile applications (using the Flutter framework), command line scripts, or server applications. Flexible compilation technology allows you to run Dart code depending on the chosen platform and developer's according goals.

## 3 Mobile Application Implementation

The implementation begins with the creation of a user interface. For this prototype of the mobile application, 2 pages were created for implementing the database and 10 pages for implementing the user interface, then their functionality and structure are described.

### 1. *welcome\_screen.dart*

Description: Welcome page of the application.

Main functions:

- Displays a random motivational quote from a list of quotes;
- carries out transition to the menu page;

Structure: The basis of the page is StatelessWidget. To implement the Random method, to select a random quote from the List, the dart: math package is enabled and the dart\_random\_choice: ^0.0.2 dependency is added in the pubspec.yaml file (Fig.1). Similarly, libraries are connected that are not part of the standard language set - after adding a dependency, the required library is automatically downloaded and installed. The FloatingActionButton button is used to go to home\_screen.dart (Fig.2).

```
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^0.1.2
  flutter_staggered_grid_view: "^0.2.5"
  intl: ^0.15.8
  sqflite: any
  path_provider: any
  dart_random_choice: ^0.0.2
  android_alarm_manager: ^0.4.1+6
```

Fig.1. List of added dependencies in the pubspec.yaml file.

### 2. *home\_screen.dart*

Description: The main menu of the application.

Main functions:

- contains four main buttons for navigating through the sections of the application: "Proceed", "Completed", "Statistics", "About the Tool";
- contains a button to add a new task from the menu;

Structure: The basis of the page is StatefulWidget. The menu is made using the StaggeredGridView, with the connection of the corresponding flutter\_staggered\_grid\_view.dart package and dependencies (Fig.1). Each menu button is an Icon wrapped in InkWell, and redirects to the appropriate section. Use the FloatingActionButton button to go to the new\_task.dart page.

### 3. *new\_task.dart*

Description: Form for adding a new task.

Main functions:

- adding data to the database by entering the text in the appropriate fields - "Name" and "Description". Selecting the priority value from the drop-down list;
- deletion of data from the database, if the record is edited;
- clean form fields, if the task has not yet been saved in the database [4];
- redirection in the menu after saving or deleting data;

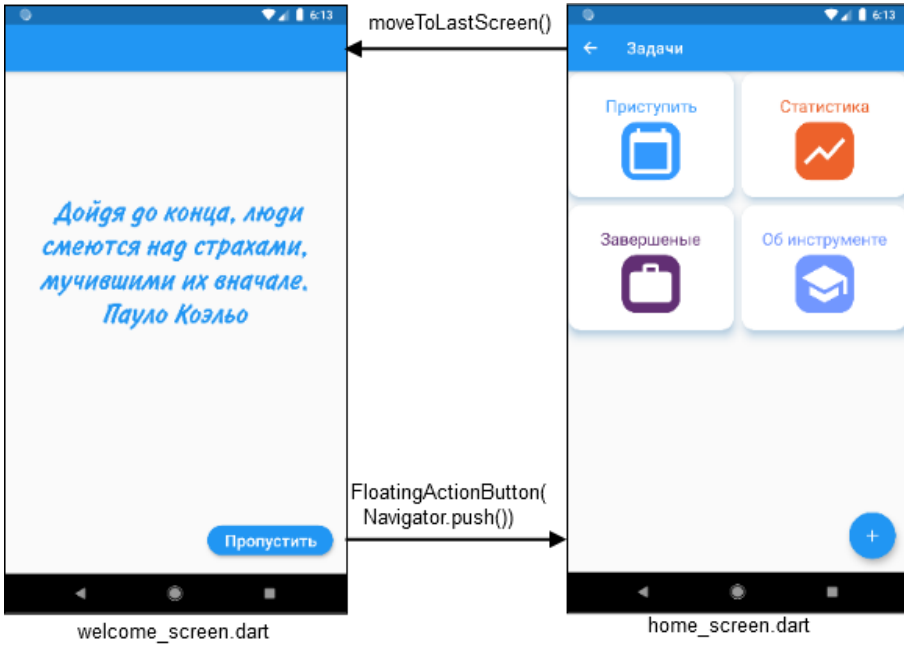


Fig.2. Navigation between application pages, home screen: enter, statistics, completion, description.

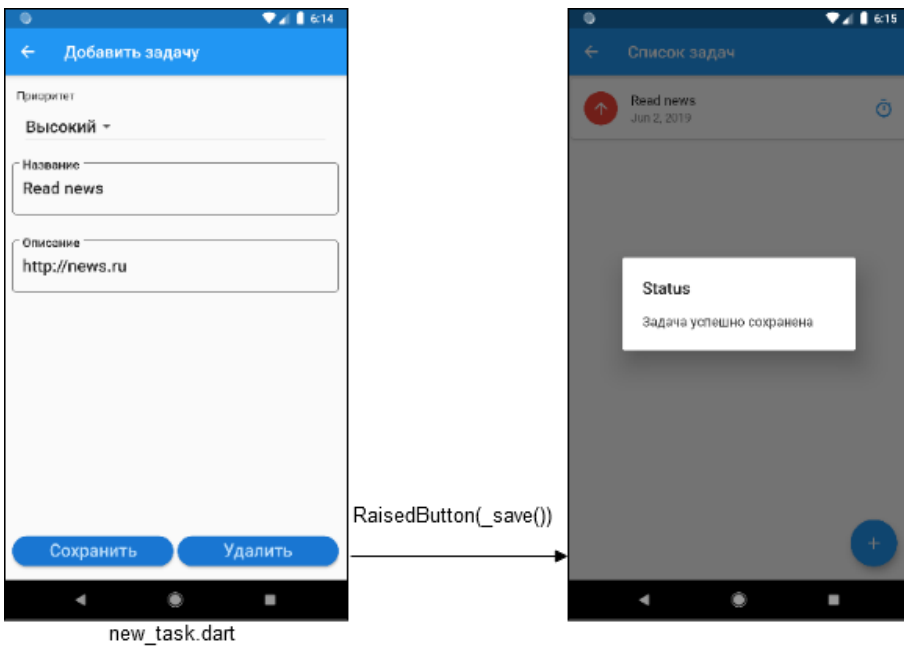


Fig.3. Adding a new task: insert task, set priority, Status: task successfully completed.

Structure: The basis of the page is `StatefulWidget`. To work with dates, the package `intl.dart` is included. The priority value is selected using `DropDownMenuItem`, the String variable is not "High" or "Low", but the Int is "1" or "2", the data type is converted using the `updatePriorityAsInt` function, and from the database is taken by `getPriorityAsString`. To capture text, use a special controller `TextEditingController`. The following `updateTitle` functions are used to work with the database - when entering data in the "Name" field, `updateDescription` - when entering data in the "Description" field, `_save` - when you click the "Save" button, `_delete` - when you click the "Delete" button. The `_showAlertDialog` function calls the status dialog box confirming successful saving (Fig.8) or deleting data in the database or displays an error.

#### 4. `to_do_task.dart`

Description: List of tasks performed.

Main functions:

- displays a list of tasks performed in accordance with the priority.

For each task is displayed:

- name;
- date of creation or editing;
- priority icon before task name. `Icons.arrow_upward` is red for high priority and `Icons.arrow_downward` is yellow for low. Changes when editing priority.
- timer icon that redirects to the timer tool page;
- go to the edit page, with the ability to save changes or delete the task;
- the possibility of adding a new task.

Structure: The basis of the page is a `StatefulWidget` (Fig.4). Using the `updateListView` function, the task list from the database is populated in accordance with the priority. Only tasks with the status "Executed" are displayed in the list. Each task is represented as a Card representing `ListTile` with the following elements: Leading Icon, changing the color and appearance of the priority icon, using the `getPriorityColor` and `getPriorityIcon` methods. Title displays the name of the task [5]. Subtitle the date the task was created or modified. Trailing Icon navigates to the timer tool (Fig.5).

Clicking the `ListTile` navigates to the edit page of the selected task. `FloatingActionButton` redirects to the `new_task.dart` page.

#### 5. `task_timer.dart`

Description: Timer tool.

Main functions:

- displays the task on which the work is being done;
- has a form in which you can put a mark if the task is completed;

Structure: The basis of the page is `StatefulWidget`. Text displays the name of the selected task. The checkbox, when clicked, changes the status of the task in the database from "Running" to "Completed" by passing the variable through `StreamController`.

#### 6. `task_work_timer.dart`, `task_rest_timer.dart`

Description: Timer for work and play.

- working timer has a duration of 25 minutes, the rest timer 5 minutes, Timer change;
- reproduces the animation of filling the circle in accordance with the elapsed time interval of the timer;
- displays the change of timer numbers;
- allows you to stop and resume the timer;
- fixing past tense;

Structure: (continued on page 9)

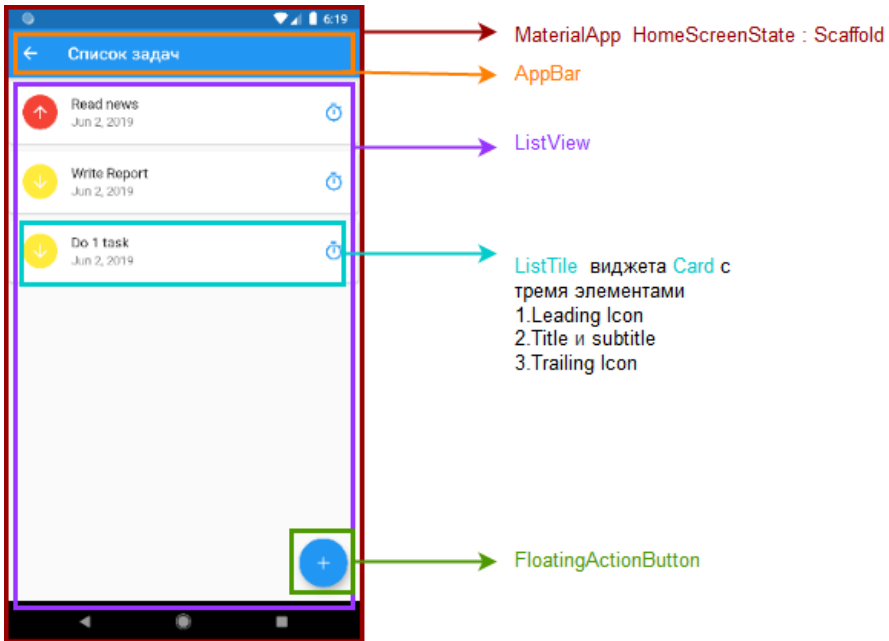


Fig.4. Division of to\_do\_task.dart into widgets: set of tasks, ListTile with three elements.

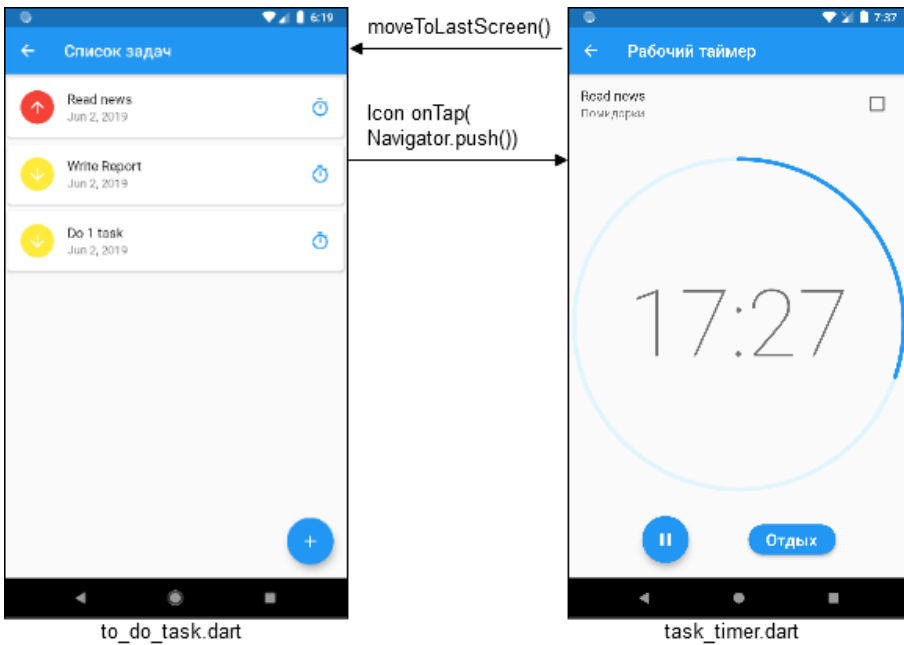


Fig.5. Choosing a timer tool, right: Work timer, button Break to rest.

Main functions:

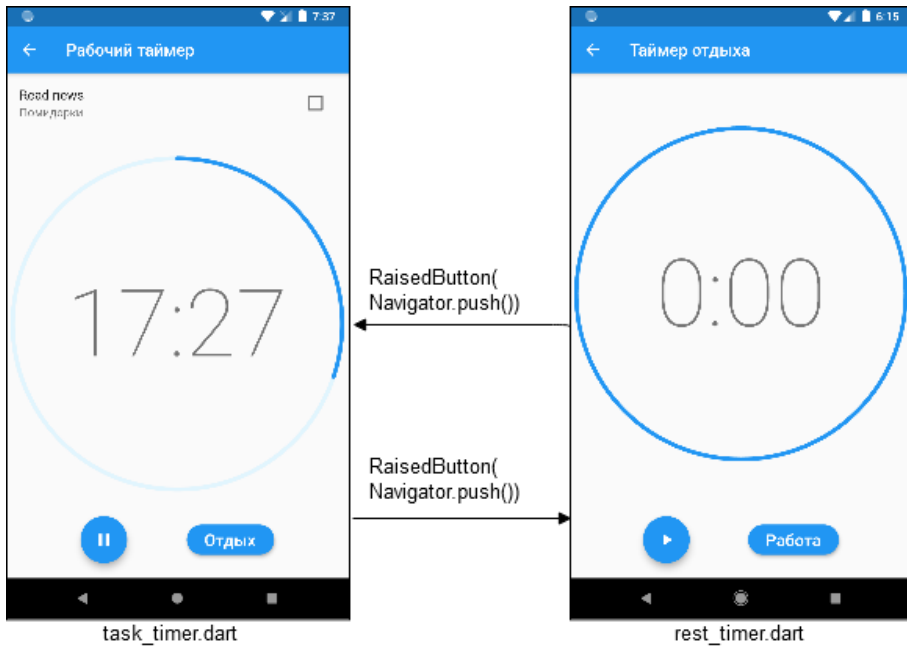


Fig.6. Timer for work and leisure, left: Work timer, button Break to rest, right: Rest timer, button Work start.

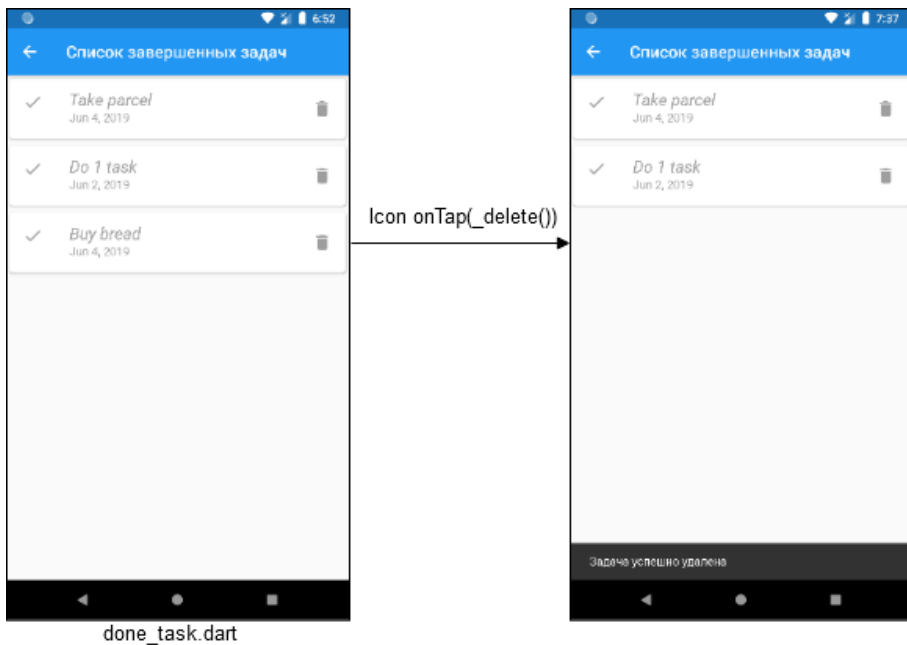


Fig.7. Removing a task from the list of completed tasks.



Fig.8. Short application instructions what to do and what button to press.

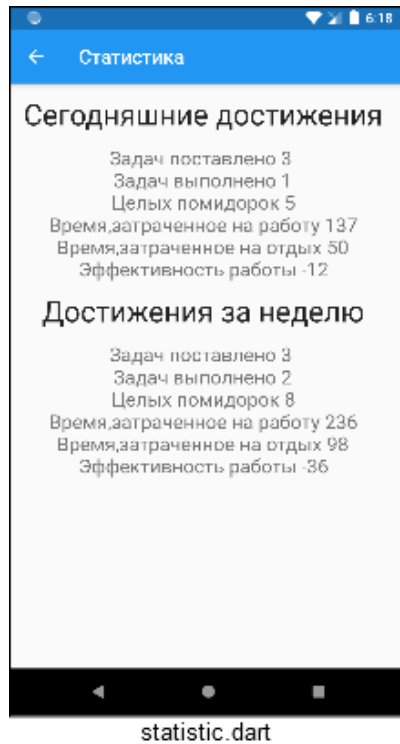


Fig.9. Statistics: Today results and Week results - number of tasks, completed tasks, time spent, time effectivity.

Structure: The basis of the pages is `StatefulWidget`. Besides the differences in the length of the timer, they are no different. First of all, the `math` package is connected and two circles are built for animation. The first is a contour, the second fills it with a brighter gradient. The following widgets were used to create circles and animation: `AspectRatio`, `AnimatedBuilder`, `CustomPaint`.

To transfer the elapsed time from the start of the timer, the `streamController` is used, which allows you to send the value of a variable using `stream` and recognize it with the help of `listen`. The time interval is stored in the `String` data type, which must be converted to an `int` using the `parse` method.

### 7. *done\_task.dart*

Description: List of completed tasks.

Main functions:

- displays a list of completed tasks;
- allows you to delete a task from the list;

Structure: The basis of the page is `StatefulWidget`. Using the `updateListView` function, the list of completed tasks from the database is populated. Each task is presented in the form of a `Card` representing `ListTile` with the following elements: `LeadingIcon`, `Title` displays the name of the task, `Subtitle` displays the date of creation or modification of the task. `TrailingIcon` allows you to delete a task from the database. `SnackBar` pops up on the bottom of the screen with a notification that the task [6, 7] was successfully deleted from the database (Fig.7).

### 8. *statistic.dart*

Description: Display statistics.

Main functions:

- displays the number of all tasks.

Structure: The basis of the page is a `StatefulWidget` with a text image of the statistics of working with the tool (Fig.9).

### 9. *how\_to\_use.task*

Description: Short instructions for using the application.

Main functions:

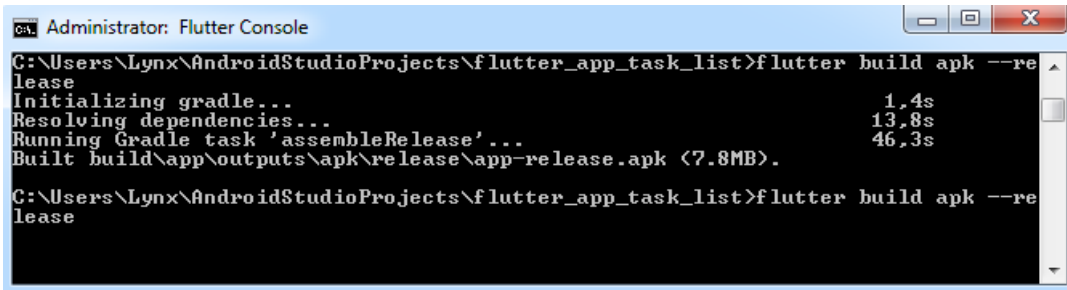
- briefly explain the principle of working with the tool;

Structure: The basis of the page is `StatefulWidget`. `PageView` is used - a scrolling list of pages, by default, i.e. scrolling goes horizontally. It has text widgets briefly describing the principle of working with this tool (Fig.8).

User interface is then created and DB is connected. Tests are conducted during development, due to the hot-reload function. To complete the development project, it remains to receive a release version of the application. To do this, use the console `flutter_console.bat`. The command line contains the location of the project and the corresponding `flutter build apk-release` command (Fig.10). The installation process of the application is standard. The application is waiting for refinement and does not have a digital signature, therefore, at the moment can not be downloaded to `PlayMarket`.

## ▀ Conclusion

Flutter is a handy tool for creating user interfaces. Nevertheless, it requires a thorough knowledge and understanding of the work of the Dart libraries. All planned functions of the application have been implemented. Prototype testing was conducted on `iOS`. The prototype can be improved with extended calculation of statistics and optimization of the widget tree.



```
Administrator: Flutter Console
C:\Users\Lynx\AndroidStudioProjects\flutter_app_task_list>flutter build apk --re
lease
Initializing gradle... 1,4s
Resolving dependencies... 13,8s
Running Gradle task 'assembleRelease'... 46,3s
Built build/app/outputs/apk/release/app-release.apk <7.8MB>.
C:\Users\Lynx\AndroidStudioProjects\flutter_app_task_list>flutter build apk --re
lease
```

Fig.10. Application release.

## References

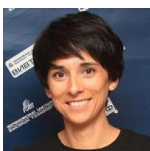
- [1] Preobrazhensky Yu. P. (2018) Information technologies used in the field of management. *Bulletin of Voronezh Institute of high technologies*. № 2(25). P. 43-46.
- [2] Vinokurov A.V. (2019) Features of the use of reactivex technology in mobile applications for the Android operating system. *Bulletin of Voronezh Institute of high technologies*. № 1(28). P. 53-57.
- [3] Kostrova V. N., Gusev A.V. (2018) Methods and tools for developing and evaluating the quality of web applications. *Bulletin of Voronezh Institute of high technologies*. № 2(25). P. 75-78.
- [4] Kamakin V. D., Lvovich I. Ya. (2018) Problems of protection of electronic documents. *Bulletin of Voronezh Institute of high technologies*. № 4(27). P. 64-66.
- [5] Melnikova T. V. (2016) Some features of business analyst's work in it-sphere. *Modeling, optimization and information technologies*. № 1 (12). P.5.
- [6] Chernikov S. Yu., Korolkov R. V. (2014) The Use of system analysis in the management of organizations. *Modeling, optimization and information technologies*. № 2 (5). P. 16.
- [7] Samoilova U. A. (2014) On some characteristics of enterprise management. *Bulletin of Voronezh Institute of high technologies*. No. 12. P. 176-179.

## Authors



**Prof. Yakov Lvovich**

Doctor of Sciences (Engineering), professor  
Voronezh Institute of High Technologies  
office@vivot.ru



**Ass. prof. Emma Lvovich, PhD.**

Candidate of Sciences, associate professor  
Voronezh Institute of High Technologies  
office@vivot.ru