

[web: suruse.uni-ruse.bg](http://web:suruse.uni-ruse.bg)

GRAPHICAL INTERFACE “FILE COMMANDER” FOR ANDROID

Valentin Velikov, Petya Marinova

Angel Kanchev University of Ruse

Abstract: *This paper presents a File Commander for Android - fully functional application, with intuitive graphical interface. The system is created to demonstrate some of the techniques, advanced technologies and knacks in the subject area and to utilize them in the teaching process. The product requirements, the application architecture and some principles for Android application development are presented.*

Keywords: *Computer Science, Android, File Commander, Android teaching.*

INTRODUCTION

The rapid development of the Internet, digital and mobile technologies leads to their rapid penetration of our everyday life, changing our values, standards and visions of communication and business. In today's life there is an increasing focus on Internet integration with mobile devices, which revolutionizes current technology - anyone with a mobile device, no matter where on the planet he is located, can have an access to personal or corporate information in Internet, to process and to send data to other mobile devices and computers, to shop, to carry out banking operations, to have fun and all this in a small, compact mobile device. Along with business, these technologies also enter the personal lives of people.

For general management of mobile devices and their associated data (both local and in the cloud), a variety of file managers have been created. With the development of this project, two objectives are accomplished:

- To create an application that maximizes the user access to the phone's file system, easy and intuitive manipulation with the desired files;
- To have an own application code that can access the device resources - the implemented techniques and functionality are used to demonstrate and illustrate relevant topics in the teaching process of Mobile Device Programming course.

RELATED WORK

Several other similar applications have been explored before creating this file manager: **ES File Explorer** [2], **ASTRO File Manager** [6], **Total Commander** [3] and others [4]. Each of them has its advantages and disadvantages that have been analysed and on this basis it has been decided what functionalities the current application should have.

1. **ES File Explorer** - This File Manager is one of the highly appreciated and used applications by users. Its functionality increases with every subsequent version. Some of the main features that are supported are:

- copying files and subdirectories between the phone and the computer;
- deleting and creating files and subdirectories;

- backup;
- removing applications;
- creating/restoring backups of applications data;
- creating shortcuts of applications;
- a set of file tools for users with rooted devices;
- accessing the entire file system and all directories and allowing the user to modify the permissions;
- multi-language support;

2. ASTRO File Manager

In its file operations this file manager covers a large part of the user's needs. The user can copy, move or delete files with one long press on them. Once the file or folder is selected, it is possible to add into selection other files and folders. It also has:

- a clear and intuitive interface and navigation between internal and external memory
- cloud storage with just one click files being moved;
- Process Manager and Task Manager, eliminating processes that reduce a battery life;
- backup and restore function in case you lose your phone;
- supports 13 languages;

3. Total Commander

- performs basic operations, including moving entire subdirectories;
- drag and drop (using a long press on a file icon);
- rename and create directories;
- delete without moving to trash bin;
- archiving and extraction;
- dialog for changing options and permissions;
- built-in text editor;
- search function (also for text);
- select / deselect groups of files;
- display a list of installed applications;
- FTP and SFTP client (plug-in);
- plug-in for Cloud Services;
- root support for the main features (optional);
- display thumbnails for photos;
- two-panel mode depicting file lists next to each other;
- multi-language support.

Table 1. Comparative analysis of basic functionalities

functionality \ app	ES File Explorer	Astro File Manager	Total Commander
Multi-language support	Yes	No	Yes
File tools for users with root-devices	Yes	No	Yes
Search function	Yes	Yes	Yes
Basic file operations	Yes	Yes	Yes
Ability to change user permissions	Yes	No	Yes
Process Manager	No	Yes	Yes
Duplex mode displaying a file lists side by side	No	Yes	Yes

Regarding the new Root User mode, it works on rooted devices and it has no effect if they do not have root rights. When acting on a rooted device, the user receives warnings about deleting system files.

The above review and study of these applications (Table 1) show that there are many possibilities for the full rendering of the file system in Android OS.

Among the drawbacks of the ASTRO File Manager is that there is no detailed description of the functionalities and no manual how to work with them in Android Market. The users have to open and examine all panels themselves to know all the features provided. Users also criticize the application's performance at certain (unrelated) times.

DETAILED DESCRIPTION

Android includes complete and overall access to the file system with SD card support and control. But Android itself does not have a built-in file manager, forcing manufacturers to create their file management applications or users to install third-party solutions.

The file manager (browser) can be viewed as a computer program that provides a user interface for accessing and/or managing networks, devices, folders, and files. The most common operations performed over files or groups of files include creating, opening (such as viewing, executing, or editing), renaming, moving or copying,

deleting and searching files, etc. Folders and files can be displayed as hierarchical tree based on their hierarchical structure [7].

The successful implementation of the two objectives, set out in the introduction, is related to the following basic requirements:

- The application must be available to the mass-market mobile devices;
- It should have a clean design;
- It should have a clear and intuitive user interface that does not differ significantly from Windows applications;
- The application has to be accessible to all age groups;
- The functional requirements should represent a set of services provided by the system. The app will provide the user with certain features such as file operations, directory creation, search and settings according to the set principal and basic functions;
- The programming units should be clear, simple, well-structured, well-described and documented.

4. Architecture and main components of Android Application

Every Android app is created as a stack of components. Each component is itself an autonomous subject that can be addressed by other components in the application as well as by external applications. This type of structure encourages component sharing. The Android app consists of four types of components: Activity, Service, Broadcast Receiver, and Content Provider [5].

The Activity component is typically a part of the application's user interface. This is a component that interacts with the user, such as a dialog where data is entered, for example. One Activity can have multiple Views (screens), allowing switching between them. Generally, more complex applications have multiple activities and users can navigate between them (through Intents).

The Service component takes care of the processes that work in the background. This component has no user interface. Services are appropriate for actions that need to be performed, regardless of what happens on the device screen. When executing such a service, it is desirable to run it as a separate thread.

The "general information receiver" (Broadcast Receiver) component is as a mailbox that receives messages from the Android OS or other applications. General information receivers have no visual representation nor are active in memory. When a certain event occurs, they execute some code, such as starting an activity or service. The application components can be registered to track this information in the Manifest file.

The Content Provider component is the application's data repository. The app can share this data with other Android OS components. Content providers are eligible to exchange a relatively large amount of data.

The described components communicate with each other through "Intents". The Intent is an Android mechanism for asynchronous communication. The components create an intent for some action and the receiving component processes it. There are separate mechanisms for delivering intents to each individual component type. Thus,

intents for activities are derived only from activity-type components, and general information intents are received by the recipients of general information.

Android apps are distributed in one package - Application Package Format - APK (ZIP-archive formatted in JAR file format). The APK file stores application components, an application policy file (called manifest), and application resources.

- Manifest file - required for all Android applications. This file should always have the AndroidManifest.xml name and be located in the app's main folder. It sets in advance what resources will be used by which application component and what rights it will have. From this file the operating system can judge whether there are resources to execute this application or not (in this case, it may even refuse to load the .apk file). If no resource is requested for a component, it will not be started;

- Application resource file - it contains data related to its behaviour during execution: images to be displayed on the screen, text message strings, custom colour definitions, fonts, icons, etc.

The basic architecture for building an application for Android is MVC (Model - View - Controller).

Another used architecture (design pattern) is Observer Pattern.

“Observer” is a behavioural design template that is used in object-oriented programming. It is utilised in cases where a group of objects (observers), after their registration to a registered object, are informed about changes in its status.

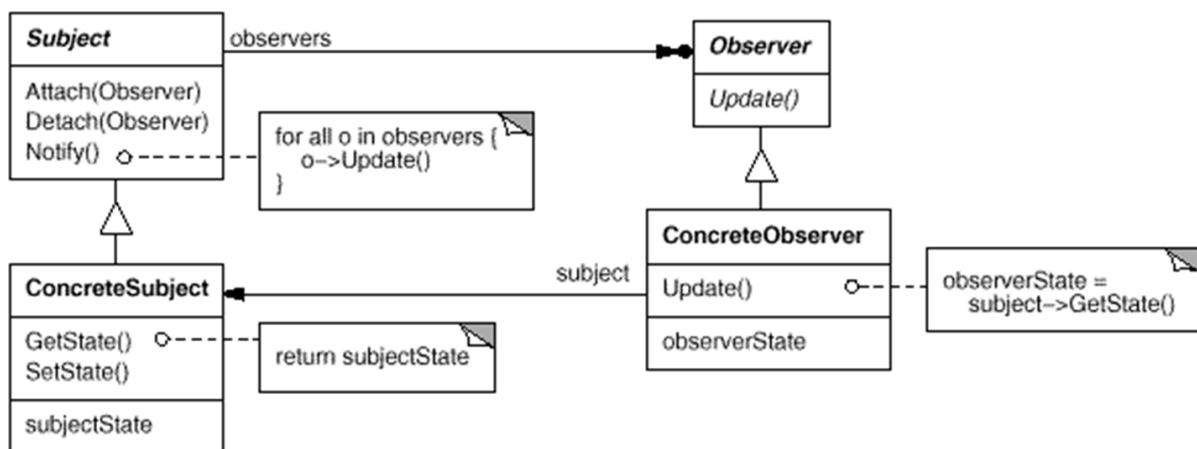


Fig.1. Architecture Observer Pattern

The “Observer” template describes an interface that allows objects to maintain consistency without being bound to specific knowledge about other objects. This does not affect the possibility of their re-use.

The Abstraction in the subject area is made up of two aspects that depend on each other [1]:

We divide the abstraction of two different objects (Subject and Observer) that can be reused and changed. These two objects can be in different layers of the application - as in one of the frequent cases of using the template - multiple view-objects subscribed to changes in one or more data repositories.

- Subject - knows which objects are watching it. The number of observers is not limited. Provides an interface for hanging new observers;
- Observer - Provides an interface for update; in this way the Subject knows how to update an Observer;
- ConcreteSubject - Stores the status of ConcreteObserver objects. Notifies observers when changing their status.

5. Basic application features

The application is implemented on a modular basis. Fig. 2 shows the interaction of the individual modules and the communication between them.

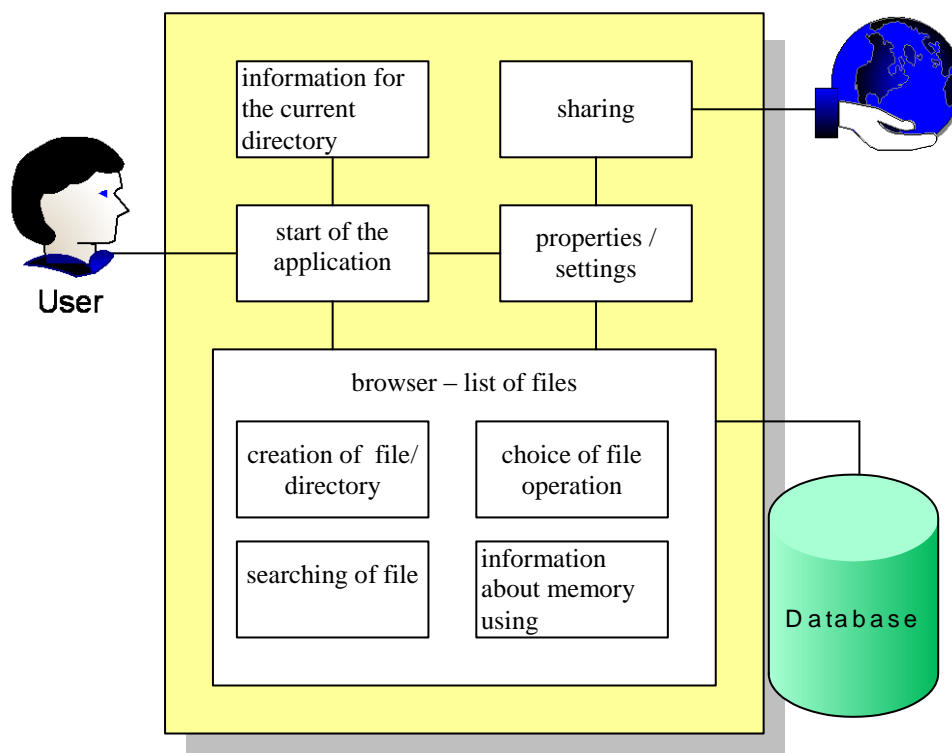


Fig.2. Basic application features

6. User Interface

The main components of the UI are Forms (windows); Buttons (text, control, etc.); Boxes (text, control, etc.); Images (icons, graphics, animations, etc.); Scrollers and other specialized ones.

One of the main problems that arises in the UI design is the insertion into a limited space of all these elements (buttons and menus, etc.) that are necessary for the realization of the functional requirements. At the same time, the main goal is to keep a clean and simple interface (Fig. 3). The user comfort and whether the user will use the application or prefer its analogue depends to a large extent on the type of interface and its structure, whether it is easy, convenient and enjoyable. The information should be presented in a straightforward and easy to understand way.

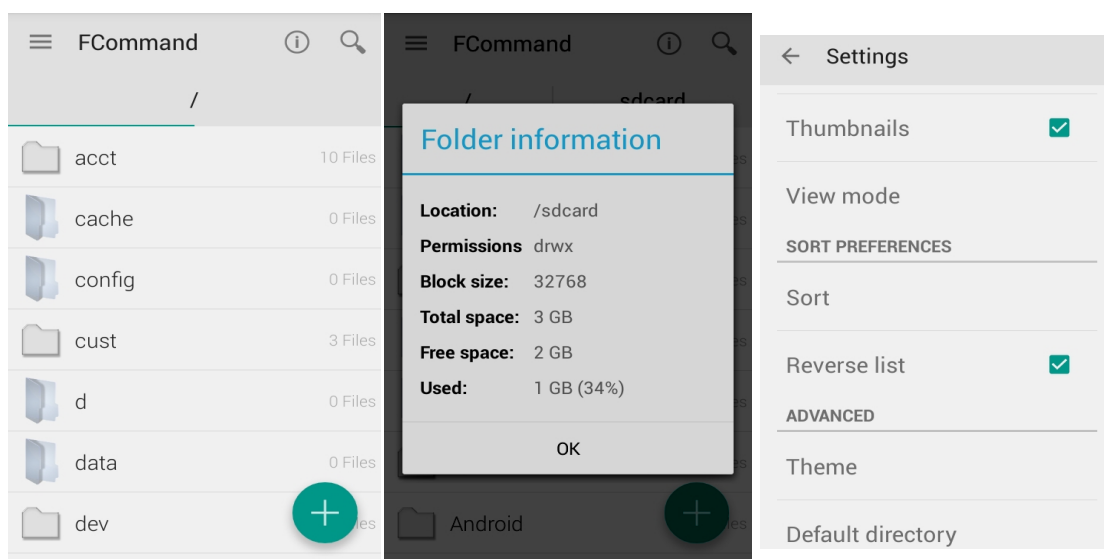


Fig.3. A part of The User Interface

A XML-file format is used to design and deploy the user interface elements.

7. User Interface – A Component Diagram

Fig.4 shows the component diagram of the user interface.

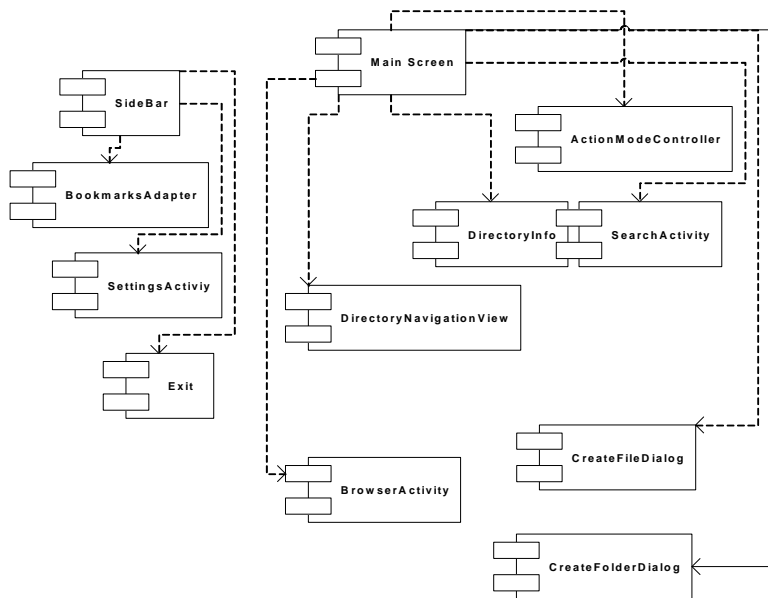


Fig.4 Component diagram of the User Interface

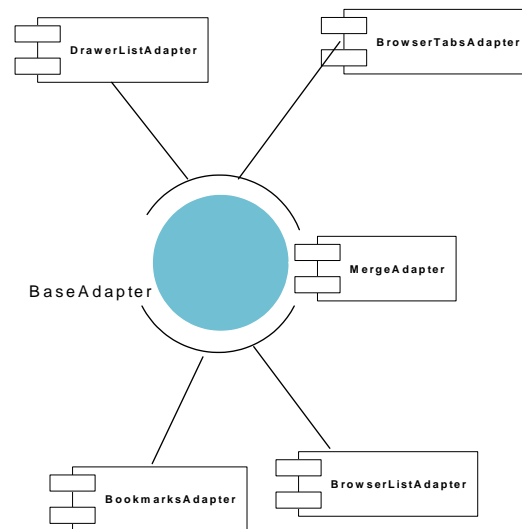


Fig.5 Component diagram of the Adapter

The adapter (Fig.5) is a structural design template that is used in object-oriented programming. It allows the interface of a class to be used by another interface. It is often needed that existing classes work with other classes without changing their source code. The Adapter object acts as a bridge between the AdapterView (View) and related data for that view. The adapter is also responsible for making a View for each item in the dataset.

TEST RESULTS

Many *functional tests* have been performed with this application to detect errors and to check whether the requirements set in the specification are met.

In this type of test, the black-box method is used, where the tester (the user) does not need to know the architecture and application code. Tests are run through the application's graphical interface and on the basis of the provided input data the output results, processes and their compliance with the functional requirements of the system are analysed.

1. File searching test

Input data: "string" type – image name

Output data (display visualization): two pictures with the same name in different folders.

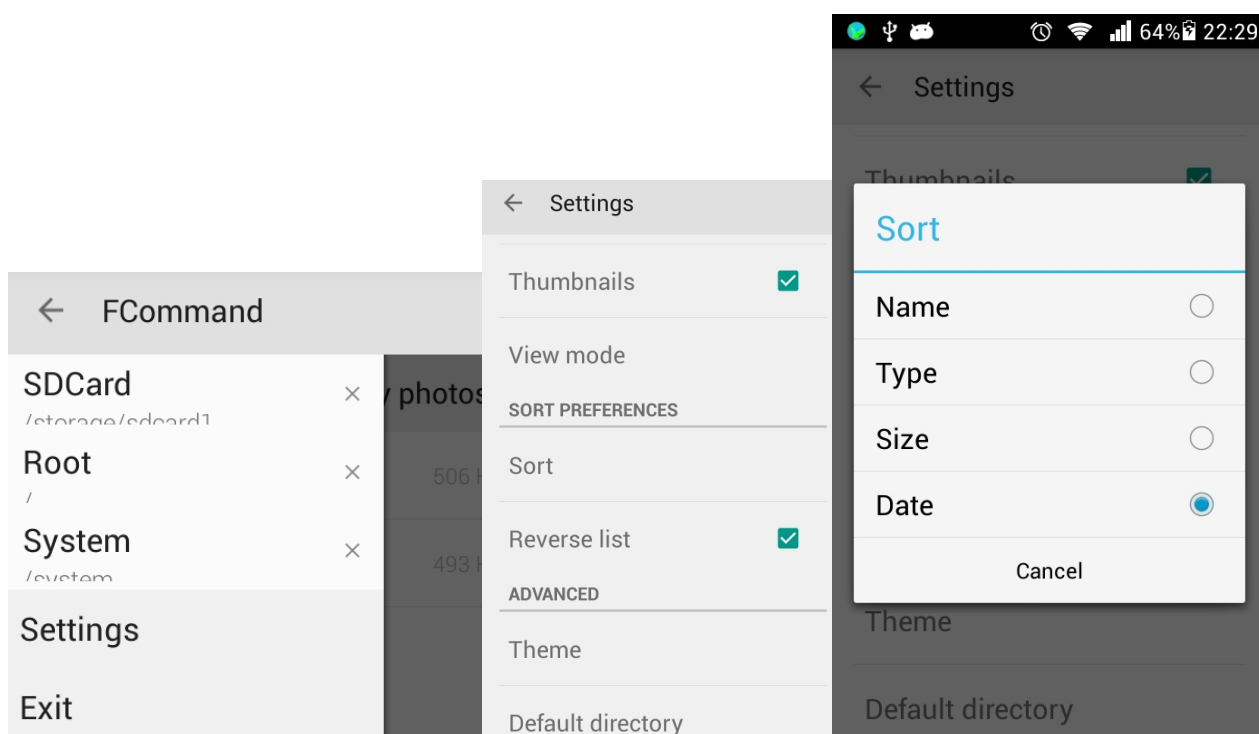


Fig.6. Functional test – settings

2. File operation test

Test performed: Select a file, select a file operation from the context menu, and select a function - Rename. Then a new file name and extension are entered.

3. Backup (zip/unzip)

It is done via the Zip menu option. The result is a file with the specified name and zip extension. When unpacking, the system asks for a path.

4. Copy/Paste

When a **copy** feature is selected, a paste icon appears after the selection has been copied in the clipboard.

It is possible to place the selected object into a directory. If there is a file with the same name, a warning dialog pops up.

5. Delete

When selecting a file and then a delete command, a warning for irreversible **deletion** is displayed, and after a confirmation the file no longer exists in the corresponding directory.

6. Properties

When selecting the **details** option, the user receives basic information about the specified file/directory.

7. Sharing

Selecting **Sharing** can upload or attach the file to the API of the corresponding master repositories installed on the user's device.

8. Testing the Settings panel

The user reaches it by sliding and a pop-up menu.

After that an additional preferred function is selected:

- **Sort** function - the user can select a file sorting arrangement by name, type, size or date;
- select a dark theme;
- select and enter the default path that opens each time when the application is loaded;
- choose how to view basic information (simplified or detailed view).

On the basis of the performed functional tests, the following conclusions can be drawn:

- the aims and objective have been successfully implemented;
- the application is targeted to users who need more options and control over their own files;
- the application offers easy access for all age groups.

CONCLUSION

A File-Commander Manager for Android was implemented with a friendly user interface. It was created as a modular application with a good documentation to be easily used for demonstration in the teaching process.

Application architecture and technology solutions contribute for easy maintenance and provide the opportunity to develop the functionality without requiring design change or serious design rework. The user gains additional access to basic features and settings that are not available in other application.

ACKNOWLEDGEMENTS

This work is supported by the National Scientific Research Fund under the contract NSRF - I02/13.

REFERENCES

- [1] Anders Goransson, Efficient Android Threading: Asynchronous Processing Techniques for Android , 2014.
- [2] Ashish, Guiding Tech, How to Use ES File Explorer to Edit and Modify System Files on Android, <http://www.guidingtech.com/14709/es-file-explorer-edit-modify-system-files-android/>, 14.03.2017.
- [3] Christian Ghisler, Total Commander, <http://www.ghisler.com/index.htm>, 16.03.2017.
- [4] John Corpuz, Best Android File Managers, Tom's Guide, <http://www.tomsguide.com/us/pictures-story/518-best-android-file-managers.html#s1>, 14.03.2017.
- [5] Joseph Annuzzi Jr., Lauren Darcey, Shane Conder. Introduction to Android Application Development: Android Essentials 2014.
- [6] Metago inc., Astro File Manager, <http://www.metago.net>, 16.03.2017.
- [7] <http://www.howtogeek.com/231401/how-to-use-android-6.0%E2%80%99s-built-in-file-manager/> .

CONTACT ADDRESSES

Pr. Assist. Valentin Velikov, PhD
Department of Informatics and
Information Technologies
Faculty of Natural Sciences and
Education
Angel Kanchev University of Ruse
8 Studentska Str., 7017 Ruse, Bulgaria
Phone: (++359 82) 888 326,
Cell Phone: (++359) 886 011 544,
E-mail: val@ami.uni-ruse.bg

Petya Marinova, BSc student
Department of Informatics and
Information Technologies
Faculty of Natural Sciences and
Education
Angel Kanchev University of Ruse
8 Studentska Str., 7017 Ruse, Bulgaria
Cell Phone: (++359) 882660599
E-mail: pmarinova0111@gmail.com

ГРАФИЧЕН ИНТЕРФЕЙС “FILE COMMANDER” ЗА ANDROID

Валентин Великов, Петя Маринова

Русенски университет „Ангел Кънчев”

Резюме: Статията представя *File Commander* за *Android* – функционално пълно приложение, с интуитивен графичен интерфейс. Системата е създадена за демонстрация на някои техники, съвременни технологии и прийоми в предметната област и да ги покаже в процеса на обучение. Представени са изискванията към продукта, архитектурата на приложението и някои принципи при създаването на приложения за *Android*.

Ключови думи: информатика, *Android*, *File Commander*, обучение на *Android*.

Requirements and guidelines for the authors - "Proceedings of the Union of Scientists - Ruse"

The Editorial Board of "Proceedings of the Union of Scientists - Ruse" accepts for publication annually both scientific, applied research and methodology papers, as well as announcements, reviews, information materials, adds. No honoraria are paid.

The paper scripts submitted to the Board should answer the following requirements:

1. Papers submitted in Bulgarian, Russian and English are accepted. Their volume should not exceed 8 pages, formatted following the requirements, including reference, tables, figures and abstract.
2. The text should be computer generated (MS Word 97 for Windows or higher versions up to Word 2003) and printed in one copy, possibly on laser printer and on one side of the page. Together with the printed copy the author should submit a disk (or send an e-mail copy to: desi@ami.uni-ruse.bg).
3. Compulsory requirements on formatting:

font - Ariel 12;
paper Size - A4;
page Setup - Top: 20 mm, Bottom: 15 mm, Left: 20 mm, Right: 20mm;
Format/Paragraph/Line spacing - Single;
Format/Paragraph/Special: First Line, By: 1 cm;
Leave a blank line under Header - Font Size 14;
Title should be short, no abbreviations, no formulas or special symbols - Font Size 14, centered, Bold, All Caps;

One blank line - Font Size 14;
Name and surname of author(s) - Font Size: 12, centered, Bold;
One blank line - Font Size 12;
Name of place of work - Font Size: 12, centered;
One blank line;
abstract – no formulas - Font Size 10, Italic, 5-6 lines ;
keywords - Font Size 10, Italic, 1-2 lines;
one blank line;
text - Font Size 12, Justify;
references;

contact address - three names of the author(s) scientific title and degree, place of work, telephone number, Email - in the language of the paper.

4. At the end of the paper the authors should write:

The title of the paper;
Name and surname of the author(s);
abstract; keywords.

Note: If the main text is in Bulgarian or Russian, parts in item 4 should be in English. If the main text is in English, they should be in Bulgarian and have to be formatted as in the beginning of the paper.

5. All mathematical signs and other special symbols should be written clearly and legibly so as to avoid ambiguity when read. All formulas, cited in the text, should be numbered on the right.

6. Figures (black and white), made with some of the widespread software, should be integrated in the text.

7. Tables should have numbers and titles above them, centered right.

8. Reference sources cited in the text should be marked by a number in square brackets.

9. Only titles cited in the text should be included in the references, their numbers put in square brackets. The reference items should be arranged in alphabetical order, using the surname of the first author, and written following the standard. If the main text is in Bulgarian or Russian, the titles in Cyrillic come before those in Latin. If the main text is in English, the titles in Latin come before those in Cyrillic. The paper cited should have: for the first author – surname and first name initial; for the second and other authors – first name initial and surname; title of the paper; name of the publishing source; number of volume (in Arabic figures); year; first and last page number of the paper. For a book cited the following must be marked: author(s) – surname and initials, title, city, publishing house, year of publication.

10. **The author(s) and the reviewer, chosen by the Editorial Board, are responsible for the contents of the materials submitted.**

Important for readers, companies and organizations

1. Authors, who are not members of the Union of Scientists - Ruse, should pay for publishing of materials.
2. Advertising and information materials of group members of the Union of Scientists – Ruse are published free of charge.
3. Advertising and information materials of companies and organizations are charged on negotiable (current) prices.

Editorial Board

ISSN 1314-3077



9 771314 307000