Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Department of computer systems software

PROGRAMMING. PART 2. Fundamentals of Web Programming and Mobile Application Development

Working program of the academic discipline (Syllabus)

Details of the academic discipline

Level of higher education	*First (bachelor)*
Branch of knowledge	*12 Information technologies*
Specialty	*121 Software engineering*
Educational program	*Software Engineering of Multimedia and Information Retrieval Systems*
Discipline status	*Normative*
Form of education	*Daytime*
Year of training, semester	*2nd year of training, 4th semester*
Scope of the discipline	*Lectures: 36 hours, computer workshop: 18 hours, self-study: 51 hours.*
Semester control/ control measures	*Credit, modular control work, calendar control*
Lessons schedule	*According to the schedule for the fall semester of the current academic year (http://roz.kpi.ua/)*
Language of teaching	*English*
Information about head of the course / teachers	Course leader: Ph.D., Peschanskyi V., vladpeschansky@gmail.com
Placement of the course	Google classroom: https://classroom.google.com/c/NTk0MTUyNDkwMDUz?cjc=igwok6x

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

Studying the discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" allows students to develop the competencies necessary for solving practical tasks of professional activities related to the development of software for network and mobile systems.

***The purpose of** study of the discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" is the formation of students' abilities to independently develop software for creating high-performance algorithms and tools for processing large volumes of data in network systems, as well as mobile applications based on the Android operating system.*

The subject of *discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" are technologies for developing software products for network and mobile systems using the Java language.*

Studying the discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" forms ***general (GC) and professional competences (PC)*** *in students of education, necessary for solving practical tasks of professional activity related to the development, improvement and support of intelligent information systems for processing multimedia data:*
PC02 *Ability to participate in software design, including modeling (formal description) of its structure, behavior and functioning processes.*
PC03 *Ability to develop architectures, modules and components of software systems.*
PC08 *Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering tasks.*
PC13 *The ability to reasonably choose and master software development and maintenance tools.*
PC14 *Ability to algorithmic and logical thinking.*
PC17 *Ability to develop software for information and search systems.*
PC19 *Ability to develop software for multimedia and multimedia systems.*

Program learning outcomes *(PLO) according to the educational program:*

PLO01 *Analyze, purposefully search for and select the information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology.*
PLO03 *Know the main processes, phases and iterations of the software life cycle;*
PLO04 *Know and apply professional standards and other legal documents in the field of software engineering;*
PLO06 *The ability to choose and use a software development methodology appropriate to the task.*
PLO07 *Know and apply in practice the fundamental concepts, paradigms and basic principles of the functioning of linguistic, instrumental and computing tools of software engineering.*
PLO08 *Know and be able to develop a human-machine interface.*
PLO09 *To be able to use methods and means of collecting, formulating and analyzing software requirements.*
PLO12 *Apply effective software design approaches in practice.*
PLO13 *Know and apply methods of developing algorithms, designing software and data and knowledge structures;*
PLO15 *Motivated to choose programming languages and development technologies to solve the tasks of creating and maintaining software;*
PLO23 *Be able to document and present the results of software development;*
PLO38 *To be able to apply programming technologies for the development of software for multimedia and information-search systems.*

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

To the successful study of the discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" precedes the study of the disciplines "Programming. Part 1. Object-oriented Programming and Design Patterns", "Fundamentals of Computer Systems and Networks", "Databases", "Fundamentals of Programming", "Algorithms and Data Structures", "Components of software engineering" of the bachelor's curriculum in the specialty 121 Software Engineering.

Received during the assimilation of the discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" theoretical knowledge and practical skillsensure the successful study of the disciplines "Multimedia Systems Software", "Standardization and Technologies for Multimedia and Information Retrieval Software Products Development", "Information Retrieval Systems Software", completion of pre-diploma practice, completion of course projects and diploma projects in the specialty 121 Software engineering .

Content of the academic discipline

Discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" involves the study of the following topics:

Topic 1. Object design using the Java language

Topic 2. Multithreaded and parallel programming using the Java language

Topic 3. Web programming using the Java language

Topic 4. Development of mobile applications for the Android OS

Test

Educational materials and resources

Basic literature:

1. Peschanskyi V. Course materials "Fundamentals of web programming and mobile application development". Provided to registered students.

Additional literature:

2. Herbert Schildt. Java: The Complete Reference, Twelfth Edition: McGraw Hill, 2021. - 1280 p.
3. Cay S. Horstmann. Core Java, Volume I—Fundamentals, Eleventh Edition: Pearson, 2018. – 928 p.
4. Cay S. Horstmann. Core Java, Volume II–Advanced Features, Eleventh Edition: Pearson, 2019. – 1040 p.
5. Richard Warburton. Java 8 Lambdas: O'Reilly Media, Inc, 2014. -168 p.
6. Brian Goetz,Tim Peierls,Joshua Bloch,Joseph Bowbeer,David Holmes, Doug Lea. Java Concurrency in Practice: Addison-Wesley Professional, 2006. -432 p.
7. Bryan Sills,Brian Gardner,Kristin Marsicano,Chris Stewart. Android Programming: The Big Nerd Ranch Guide, 5th Edition: Addison-Wesley Professional, 2022. - 688 p.
8. Reto Meier. Professional Android 4 Application Development, 3rd edition: Wrox, 2012. - 864 p.
9. SGGanesh, Tushar Sharma. Java SE7 Programmer Exams 1Z0-804 and 1Z0-805. A Comprehensive OCPJP 7 Certification Guide. – APRESS, New York, 2012.-644 p.

Educational content

5. Methods of mastering an educational discipline (educational component)

No	*Type of training session*	*Description of the training session*
Topic 1. Object design using the Java language		
1	*Lecture 1. Implementation of algorithms. The main operators of the Java language.*	*Comparison operations. Data type boolean. The if and if-else branching statements. Cycles. The for operator. The scope of the variable. Nested loops. While and do-while statements. Types of arrays, methods of their description.*

		One-dimensional arrays. The length field. Operators of division by modulo %, increment, decrement, operations with assignment, ternary operator. Logical function XOR, short-circuit logic functions. Operators switch-case, break, continue, foreach. *Tasks on self-study: item 6 №1.*
2	*Lecture 2. Encapsulation, MODEL-VIEW-CONTROLLER pattern.*	*Introduction to Java API, java.lang package, concepts of method overloading, import, static keyword and classes java.util. Arrays, java.util.ArrayList. Concept of encapsulation, class structure. Creating your own classes. Declaration and initialization of fields. Declaration of methods. Announcement and creation of objects. Operator new. The literal is null. MODEL-VIEW-CONTROLLER (MVC) architectural pattern for building your own classes. Familiarization with constructors, transfers to the method of references to objects.* *Tasks on self-study: item 6 №2.*
3	*Computer workshop 1 (part 1). Using the MODEL-VIEW-CONTROLLER template to build classes and implement algorithms.*	*Task: to develop Java code for determining the type of intersection of rectangles and for outputting dotted geometric shapes based on the MODEL-VIEW-CONTROLLER template.* *Tasks on self-study: item 6 №3.*
4	*Lecture 3. Basic concepts and syntactic tools of the Java language.*	*The static keyword. Local variables, object and class variables. Peculiarities of passing primitives and references to the method as arguments. Types of methods. Reloaded constructors. Reloading methods. Main() method. Arguments in the main() method. Initialization of variables. Initialization blocks. Initialization procedure. Packages. Import. Static import. Math class.* *Tasks on self-study: item 6 №4.*
5	*Lecture 4. Imitation.*	*The concept of imitation. Has-a, is-a relation. UML diagrams. Classes Calendar, GregorianCalendar, Date. Access modifiers. Keywords this, super. Object class. Overriding methods. toString() method.* *Tasks on self-study: item 6 №5.*
6	*Computer workshop 1 (part 2). Using imitation to build hierarchical class structures.*	*Task: For class hierarchy: Vehicle => Airplane, Car, Ship; Airplane => Passenger plane, Transport plane, Fighter plane, Car => Passenger car, Bus, Truck, Ship => Passenger liner, Tugboat, Tanker - declare 1-3 most appropriate fields in each class, introduce getters/methods in each class setters for all class fields. Give private-level access to the fields of all classes, public-level access to the getters/setters methods for all classes,*

		except for the Fighter class, in which only the classes of one package can change the field values, and the subclasses of the Fighter class from other packages can also read the field values, in to each class, introduce a constructor to initialize all class fields and a toString() method. *Tasks on self-study: item 6№6.*
7	*Lecture 5. Polymorphism.*	*Polymorphic links. Polymorphism. The equals() method. Covariant returns. Casting types. The instanceof operator. The keyword final.* *Tasks on self-study: item 6№7.*
8	*Lecture 6. Interfaces.*	*Abstract classes. Interfaces. Interfaces Comparable, Comparator. Inner classes (regular, local, anonymous, static). Functional interfaces and lambda expressions.* *Tasks on self-study: item 6№8.*
9	*Computer workshop 2 (part 1). Using polymorphism in hierarchical class structures.*	*For the task of forming dotted geometric shapes, develop an efficient hierarchy of Shape classes with overridden methods.* *Tasks on self-study: item 6№9.*
10	*Lecture 7. Handling of exceptional situations. Work with text files.*	*Exceptions. Work with text files.* *Tasks on SRS: item 6№10.*
11	*Lecture 8. Features of some standard Java API classes.*	*Strings, StringBuffers, StringBuilders. Wrappers, autoboxing, autounboxing. Enums. Arrays class, Var-args, Garbage collector, Java SE8 API for working with dates and times. Classes of the java.time package: LocalDate, LocalTime, LocalDateTime, Period. Class java.time.format.DateTimeFormatter.* *Tasks on self-study: item 6№11.*
12	*Computer workshop 2 (part 2). Work with text files.*	*Task: Read and process data from three existing files and enter processed data into files created by Java tools.* *Tasks on self-study: item 6№12.*
Topic 2. Multithreaded and parallel programming using the Java language		
13	*Lecture 9. Algorithms.*	*Processing of text data (parsing) based on regex technology. Serialization of files. Serializable and Externalizable interfaces. Peculiarities of composition and imitation during serialization.* *Tasks on self-study: item 6№13.*

14	*Lecture 10. Containers.*	*Collections framework. Principles of building hash tables. Interfaces Set, Queue, Mar, Iterator. Classes TreeSet, PriorityQueue, Dequeue, TreeMap, HashMap. Construction and use of generic classes and generic methods. Features of polymorphism when using generic collections.* *Tasks on self-study: item 6№14.*
15	*Computer workshop 3 (part 1). Using features of collections in complex algorithmic problems.*	*Task: N points are given on the plane. It is necessary to write two HashMaps in the LINES file:* *- with keys in the form of objects of the Point class defined by integer coordinates of points, and values in the form of the number of straight lines passing through this point and at least one more point,* *- with keys in the form of objects of the Line class, defined by parameters K and B of the line y = K*x + B, and values in the form of the number of points belonging to this line.* *Tasks on self-study: item 6№15.*
16	*Lecture 11. Multithreaded programming based on Threads technology.*	*The basics of creating and using streams. Thread class. Runnable, Callable, Executor, ExecutorService interfaces. Basics of stream synchronization. The keyword synchronized. Synchronization of static methods. Package java.util.concurrent.atomic. Implementation of synchronization based on the resources of the java.util.concurrent package. Thread-safe collections. Implementation of synchronization based on java.util.concurrent.lock package resources. Interaction of flows. Wait, notify methods of the Object class. The concept of deadlock.* *Tasks on self-study: item 6№16.*
17	*Lecture 12. API Stream framework.*	*Intermediate and terminal methods of the Stream interface for forming and processing streams. Optional class. Features of streaming technology. The reduce method. Methods of the Collectors class. Stream parallelization. Comparative analysis of serial and parallel flows, determination of feasibility of parallelization of flows.* *Tasks on self-study: item 6№17.*
18	*Computer workshop 3 (part 2). Construction of multithreaded parallel software systems.*	*Task: Passengers arriving on three (four, five) planes at the same time are transported by minibuses. The aircraft load is exactly 100 passengers. Minibuses have a capacity of 6, 7, or 8 passengers and go to 4 different cities. Minibus occupancy must be 100%, except for minibuses with the last passengers. Passengers travel in families from 1 to 4 people. Families cannot be separated by minibuses. Make a single-thread system first, then a*

		maximally parallelized synchronous multi-thread system, with synchronization on planes or buses. *Tasks on self-study: item 6№18.*
Topic 3. Web programming using the Java language		
19	*Lecture 13. Servlets and JSP.*	*Interfaces ServletContext, ServletConfig, ServletRequest, HttpServletRequest, ServletResponse, HttpServletResponse. Processing requests. Standard action elements. JSP document. JSTL tag library. Expression Language.* *Tasks on self-study: item 6№19.*
20	*Lecture 14. JDBC technology for working with databases.*	*JDBC Drivers and Configuration. Interfaces and classes of the java.sql package. Query operators.* *Tasks on self-study: item 6№20.*
21	*Computer workshop 4. Working with remote databases.*	*Task: Read and enter data from a remote database in the appropriate collections in the format:* *N Bottle Volume Material* *1 Wine 0.75 Glass* *2 Juice 0.25 Metal* *Tasks for forming requests:* *- read bottles with a capacity: no more than 0.5 l, in the range from 0.51 l to 0.99 l, no less than 1.0 l,* *- read bottles for: wine, juices, water,* *- read bottles made of: metal, glass, plastic.* *Enter data from text files into the database.* *Tasks on self-study: item 6№21.*
Topic 4. Development of mobile applications for the Android OS		
22	*Lecture 15. Building a user interface. Basic widgets*	*The structure of the Android project. XML language. TextView widget. Classes Activity, Application. Resource. Work with images. ImageView widget. The concept of markup (to the layout manager). Markup management. ConstraintLayout layout. Entering texts. Text scroll bars. EditText, ScrollView widgets. Working with buttons. Widgets Button, RadioButton, ToggleButton, ImageButton. Indicators. ProgressBar widget. Working with the menu. Interface Menu, classes ContextMenu, SubMenu. Message. Toast widget. Creating custom messages. Dialog boxes with buttons. AlertDialog widget. Dialog window with progress indicators. ProgressDialog widget.* *Tasks on self-study: item 6№22.*

23	*Lecture 16. Navigation. Fragments.*	*Principles of navigation. Intent class. Navigation without data transmission. Navigation with data transmission. Fragments. Fragment, FragmentActivity and FragmentManager classes.* *Tasks on self-study: item 6№23.*
24	*Computer workshop 5 (part 1). Development of multi-window Android applications.*	*Task: using fragment technology, develop a two-window calculator to perform arithmetic (first window) and algebraic (eg exponentiation, second window) functions.* *Tasks on self-study: item 6№24.*
25	*Lecture 17. Adapters*	*Purpose and types of adapters. ListView and ArrayAdapter classes. Interoperability of adapters and collections based on the Map interface. Adapters for stacked collections. BaseAdapter class.* *Tasks on self-study: item 6№25.*
26	*Computer workshop 5 (part 2). Using collections in Android applications.*	*Task: Develop a phone book with the ability to delete and edit addresses.* *Tasks on self-study: item 6№26.*

6. Independent work of a student/graduate student

Discipline "Programming. Part 2. Fundamentals of Web Programming and Mobile Application Development" is based on independent preparations for classroom classes on theoretical and practical topics.

№s/p	*The name of the topic submitted for independent processing*	*Number of hours*	*literature*
1	*Preparation for the lecture 1*	*1*	*1*
2	*Preparation for lecture 2*	*1*	*1*
3	*Preparation for computer workshop 1 (part 1)*	*3*	*1*
4	*Preparation for the lecture 3*	*1*	*1*
5	*Preparation for the lecture 4*	*1*	*1*
6	*Preparation for computer workshop 1 (part 2)*	*3*	*1*
7	*Preparation for the lecture 5*	*1*	*1*
8	*Preparation for the lecture 6*	*1*	*1*
9	*Preparation for computer workshop 2 (part 1)*	*3*	*1*
10	*Preparation for the lecture 7*	*1*	*1*
11	*Preparation for the lecture 8*	*1*	*1*
12	*Preparation for computer workshop 2 (part 2)*	*3*	*1*
13	*Preparation for the lecture 9*	*1*	*1*
14	*Preparation for lecture 10*	*1*	*1*

15	*Preparation for computer workshop 3 (part 1)*	*3*	*1*
16	*Preparation for lecture 11*	*1*	*1*
17	*Preparation for lecture 12*	*1*	*1*
18	*Preparation for computer workshop 3 (part 2)*	*3*	*1*
19	*Preparation for lecture 13*	*1*	*1*
20	*Preparation for lecture 14*	*1*	*1*
21	*Preparation for the computer workshop 4*	*3*	*1*
22	*Preparation for lecture 15*	*1*	*1*
23	*Preparation for lecture 16*	*1*	*1*
24	*Preparation for computer workshop 5 (part 1)*	*3*	*1*
25	*Preparation for lecture 17*	*1*	*1*
26	*Preparation for computer workshop 5 (part 2)*	*3*	*1*
27	*Preparation for MKR*	*4*	*1*
28	*Preparation for the test*	*3*	*1*

Policy and control

7. Policy of academic discipline (educational component)

Attending lectures is mandatory.

Attending computer workshop classes may be occasional and for consultation/protection of computer workshop works.

Rules of behavior in classes: activity, respect for those present, turning off phones.

Adherence to the policy of academic integrity.

Rules for protecting the work of the computer workshop: the work must be done in accordance with the tasks set.

8. Types of control and rating system for evaluating learning outcomes (RSO)

During the semester, students perform 5 computer workshops. The maximum number of points for each computer workshop: 15 points.

Points are awarded for:
- quality of performance of the computer workshop: 0-10 points;
- answer during the defense of the computer workshop: 0-4 points;
- timely presentation of work for defense: 0-1 points.

Performance evaluation criteria:
9-10 points – the work is done qualitatively, in full;
7-8 points – the work is done qualitatively, in full, but has shortcomings;
5-6 points – the work is completed in full, but contains minor errors;
1-4 points – the work is completed in full, but contains significant errors;
0 points - the work is not completed in full.

Answer evaluation criteria:

5-6 points – the answer is complete, well-argued;
3-4 points – the answer is correct, but has flaws or minor errors;
1-2 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

Criteria for evaluating the timeliness of work submission for defense:
4 points – the work is presented for defense no later than the specified deadline;
0 points – the work is submitted for defense later than the specified deadline.

The maximum number of points for performing and defending computer practicals:
RS = 15 points × 5 computers. practice = 75 points.

The modular test consists of 1 theoretical question and 1 practical task. The maximum number of points for a theoretical question is 10 points.
Evaluation criteria for a theoretical question:

Assessment criteria for the practical task:
10 points – the answer is correct, complete, well-argued;
6-9 points - in general, the answer is correct, but has shortcomings;
1-5 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

The maximum number of points for a practical task is 15 points.
Assessment criteria for the practical task:
14–15 points – the answer is correct;
12–13 points – the answer is generally correct, but has flaws;
9–11 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.

R = RS = Rcomp. practical + Rmodular test = 75 points + 25 points = 100 points.

Semester control: assessment
Conditions for admission to the semester control: with a semester rating (RC) of at least 60 points and the enrollment of all computer practical work, the student receives credit "automatically" according to the table (Table of correspondence of rating points to grades on the university scale). Otherwise, he has to perform the final control work. Completion and defense of the computer workshop is a necessary condition for admission to the credit control work. If the student does not agree with the "automatic" grade, he can try to improve his grade by writing a credit test, while his points received for the semester are kept, and the better of the two grades received by the student is assigned ("soft" grading system) .

Table of correspondence of rating points to grades on the university scale:

Scores	*Rating*
100-95	Perfectly
94-85	Very good
84-75	Fine
74-65	Satisfactorily
64-60	Enough
Less than 60	Unsatisfactorily
Admission conditions not met	Not allowed

9. Additional information on the discipline (educational component)

The list of questions submitted for semester control is given in Appendix 1.

Working program of the academic discipline (syllabus):

Folded Peschanskyi V.

Adopted by Computer Systems Software Department (protocol № 12 from 26.04.23)

Approved by the Faculty Board of Methodology (protocol № 10 from 26.05.23)

Appendix 1. List of questions submitted for semester control

1. Implementation of algorithms. The main operators of the Java language.
2. Encapsulation.
3. MODEL-VIEW-CONTROLLER template.
4. Classes Calendar, GregorianCalendar, Date.
5. UML diagrams.
6. The concept of imitation. Has-a, is-a relation. toString() method.
7. Access Modifiers.
8. The keyword this.
9. The keyword super.
10. The toString() method.
11. Keyword static.
12. Peculiarities of passing primitives and references to the method as arguments.
13. Reloading Methods.
14. The main() method. Arguments in the main() method.
15. Initialization of variables. Initialization blocks. Initialization procedure.
16. Packages. Import. Static import.
17. Math class.
18. Polymorphic references. Polymorphism.
19. The equals() method.
20. Covariant returns.
21. Casting types. The instanceof operator.
22. The keyword final.
23.Abstract classes.
24. Interfaces.
25. Comparable, Comparator interfaces.
26. Internal classes (regular, local, anonymous, static).
27. Functional interfaces and lambda expressions.
28. Handling of exceptional situations.
29. Work with text files.
30. Strings, StringBuffers, StringBuilders.
31. Wrappers, autoboxing, autounboxing.
32. Enums.
33. Arrays class.
34. Var-args.
35. Garbage collector.
36. Classes of the java.time package: LocalDate, LocalTime, LocalDateTime, Period.
37. Class java.time.format.DateTimeFormatter.
38. Processing of text data (parsing) based on regex technology.
39. Serialization of files. Serializable and Externalizable interfaces
40. Collections framework. Set, Queue, Iterator interfaces. Classes TreeSet, PriorityQueue, Dequeue.
41. Principles of building hash tables. Interface Mar. TreeMap, HashMap classes.
42. Construction and use of generic classes and generic methods.
43. Features of polymorphism when using generic collections.
44. Basics of creating and using streams. Thread class. Runnable, Callable, Executor, ExecutorService interfaces.

45. Basics of thread synchronization. The keyword synchronized. Synchronization of static methods.
46. Package java.util.concurrent.atomic.
47. Implementation of synchronization based on the resources of the java.util.concurrent package. Thread-safe collections. 48. Implementation of synchronization based on the resources of the java.util.concurrent.lock package.
49. Interaction of streams. Wait, notify methods. The concept of deadlock.
50. Intermediate and terminal methods of the Stream interface
51. Optional class.
52. Features of the streaming technology. The reduce method.
53. Methods of the Collectors class.
54. Parallelization of streams. Comparative analysis of serial and parallel streams.
55. ServletContext, ServletConfig, ServletRequest, HttpServletRequest, ServletResponse, HttpServletResponse interfaces. Processing requests.
56. Standard action elements. JSP document. JSTL tag library. Expression Language.
57. Principles of navigation. Intent class. Navigation.
58. Fragments. Fragment, FragmentActivity and FragmentManager classes.
59. Purpose and types of adapters. ListView and ArrayAdapter classes.
60. Interoperable use of adapters and collections based on the Map interface.
61. Adapters for assembled collections. BaseAdapter class.
62. Text input. TextView widget. Text scroll bars. EditText, ScrollView widgets.
63. Classes Activity, Application. Resource.
64. Work with images. ImageView widget.
65. The concept of layout manager. Markup management. ConstraintLayout layout.
66. Working with buttons. Widgets Button, RadioButton, ToggleButton, ImageButton.
67. Indicators. ProgressBar widget.
68. Working with the menu. Interface Menu, classes ContextMenu, SubMenu.
69. Notice. Toast widget. Creating custom messages.
70. Dialog boxes with buttons. AlertDialog widget.
72. Dialog window with progress indicators. ProgressDialog widget.