



Лаборатори №1

Сэдэв: First Android app (Unit 1)

Гүйцэтгэсэн: О.Саруулгэрэл /22B1NUM1637/

Шалгасан: Г.Ууганбаяр

2024 он

Улаанбаатар хот

Гарчиг

| | |
|---|----|
| Introduction to Kotlin | 3 |
| Quiz (Introduction to Kotlin) | 3 |
| Classes and Objects | 4 |
| Quiz | 4 |
| Set up Android Studio | 5 |
| Quiz(Set up Android Studio) | 5 |
| Build a basic layout | 7 |
| Design a birthday card app..... | 7 |
| Prctice: | 9 |
| Compase Basics | 9 |
| Task Manager | 12 |
| Compose Quadrant..... | 14 |
| Project: Create a Business Card App | 17 |

Introduction to Kotlin

Quiz (Introduction to Kotlin)

- Which of the following variable declarations is valid?
 - `var hello: Int? = ""`
 - `String "hello" = hello`
 - `val hello = "hello"`**
 - `hello: String = "hello"`
- It is considered best practice to declare a variable that will not change using `var` instead of `val`.
 - True
 - False**
- Which of the following are valid ways to update a variable?

Choose as many answers as you see fit.

 - `total++`**
 - `total - 1`
 - `total--`**
 - `total = total + 1`**
- In Kotlin, comments can be single or multi-line and are ignored by the compiler.
 - True**
 - False
- Which of the following is not a data type in Kotlin?
 - String
 - Decimal**
 - Int
 - Boolean
- Float also represents a decimal, but is less precise than Double.
 - True**
 - False
- In Kotlin, the entrypoint of a program is the _____.
 - `println()` statement

- b. val variable
 - c. **main() function**
 - d. return statement
8. Which of the following are true about function return values?
Choose as many answers as you see fit.
- a. **If a function does not specify a return type, the return type is Unit.**
 - b. **A return value can be stored in a variable.**
 - c. Functions with a return type of Unit must include a return statement.
 - d. **A return value's type must match the return type of a function.**
9. Which of the following are true about functions?
Choose as many answers as you see fit.
- a. **Functions can take parameters, or variables as inputs.**
 - b. Function parameters are required to have default arguments.
 - c. **When calling a function with parameters, the values passed in are called arguments.**
 - d. **Breaking up your code into separate functions makes your code easier to maintain.**
10. With named arguments, you can change the order in which you pass arguments into a function.
- a. **True**
 - b. False

Classes and Objects

Quiz

1. What is the default behavior of classes in Kotlin?
- a. All classes are private
 - b. All classes are sealed
 - c. **All classes are final**
 - d. All classes are protected
2. Which of the following is the correct way of declaring objects for class Book in Kotlin?
- a. Book b1 = new Book()
 - b. **val b1 = Book()**
 - c. var b1 = (Book) new()
 - d. val b1 = (new) Book()
3. What is true about extension functions in Kotlin?
Choose as many answers as you see fit.
- a. **Allows you to write new functions for a class that you cannot modify**
 - b. **Use dot notation to call them on variables of the type you are extending**
 - c. It must be declared in the same file as the original class you are extending

- d. Use the extend keyword to declare these functions
- 4. Which of the following is true about data classes?
Choose as many answers as you see fit.
 - a. It is required for you to implement the toString() method
 - b. No parameters are needed for the primary constructor
 - c. Use it for classes that store data
 - d. Getter methods are automatically generated for you (and setter methods for mutable variables)
- 5. Which keyword is used to indicate that a class can be subclassed?
 - a. public
 - b. protected
 - c. internal
 - a. Open

Set up Android Studio

Quiz(Set up Android Studio)

- 1. What does IDE stand for?
 - a. Integrated Development Environment
 - b. Independent Design Environment
 - c. Ideal Developer Environment
 - d. Intelligent Design Environment
- 2. Which of the following are advantages of using Android Studio?
Choose as many answers as you see fit.
 - a. It can help prevent typos and other mistakes in your code.
 - b. It comes with a virtual device called an emulator that can run your app.
 - c. It can show you a real-time preview of how your app will look on-screen while you code.
 - d. It can automatically translate your app into other languages.
- 3. What is the purpose of using a virtual device, or emulator, in Android Studio?
 - a. To show a variety of error messages to users
 - b. To experiment with app code safely
 - c. To test your app on a device without having that physical device
 - d. To see what your app looks like in a web browser
- 4. In Android Studio, what is a project template good for?
Choose as many answers as you see fit.

- a. It causes Android Studio to download files faster.
 - b. It makes getting started on building a new app faster.
 - c. It provides a structure that follows best practices.
 - d. It is the only way to build apps that can be previewed in Android Studio.
 - e. It makes building a new app less error-prone by pre-populating the project with some app code.
5. How do you create a new project in Android Studio?
- a. Log out of Android Studio, and navigate to your project folder to find instructions.
 - b. If you have a project already open, select File > New > New Project from the Android Studio menu.
 - c. In the “Welcome to Android Studio” window, click “Start a new Android Studio project.”
 - d. Create a new file on your computer, and title it “New Android Studio Project.”
 - e. Both B and C are ways to create a new project in Android Studio.
 - f. None of the above
6. ____ is a function that is used to define a layout in your app using Composable functions.
- a. ComponentActivity()
 - b. onCreate()
 - c. DefaultPreview()
 - d. setContent()
7. A Compose function requires the @Composable annotation.
- a. True
 - b. False
8. A ____ is a Composable that has a background color and can contain other Composables.
- a. Color
 - b. Container
 - c. Surface
 - d. Box
9. Padding is an example of a ____
- a. Property
 - b. Composable
 - c. Attribute
 - d. Modifier
10. Which of the following is false about Compose?
- a. The Empty Compose Activity template is used to create a simple app.

- b. Layouts can be viewed in the Preview window, without actually running your app.
- c. All elements and themes in a Compose app are contained in a Surface.
- d. Themes, such as GreetingCardTheme allow you to style Composables.

Build a basic layout

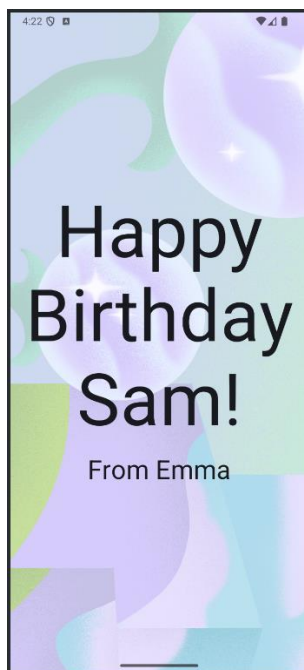
Design a birthday card app

```

1. package com.example.happybirthday
2.
3. import com.example.happybirthday.R as R1
4.
5.
6. class MainActivity : ComponentActivity() {
7.     override fun onCreate(savedInstanceState: Bundle?) {
8.         super.onCreate(savedInstanceState)
9.         enableEdgeToEdge()
10.        setContent {
11.            HappyBirthdayTheme {
12.                Surface(
13.                    modifier = Modifier.fillMaxSize(),
14.                    color = MaterialTheme.colorScheme.background
15.                ) {
16.                    GreetingImage("Happy Birthday Sam!", from = "From Emma", modifier =
Modifier.padding(8.dp))
17.                }
18.            }
19.        }
20.    }
21. }
22. @Composable
23. fun GreetingText(message: String, from: String, modifier: Modifier = Modifier){
24.    Column(
25.        modifier = Modifier
26.            .padding(8.dp)
27.            .fillMaxSize(),
28.        verticalArrangement = Arrangement.Center
29.    ){
30.        Text(
31.            text = message,
32.            fontSize = 100.sp,
33.            lineHeight = 116.sp,
34.            textAlign = TextAlign.Center
35.        )

```

```
36.     Text(
37.         text = from,
38.         fontSize = 36.sp,
39.         modifier = Modifier
40.             .padding(16.dp)
41.             .align(alignment = Alignment.CenterHorizontally)
42.     )
43. }
44. }
45.
46. @Composable
47. fun GreetingImage(message: String, from: String, modifier: Modifier = Modifier){
48.     val image = painterResource(R1.drawable.androidparty)
49.     Box() {
50.         Image(
51.             painter = image,
52.             contentDescription = null,
53.             contentScale = ContentScale.Crop,
54.             alpha = 0.5F
55.         )
56.         GreetingText(
57.             message = message,
58.             from = "From Emma",
59.             modifier = Modifier
60.                 .fillMaxSize()
61.                 .padding(8.dp)
62.         )
63.     }
64. }
65. }
66.
67. @Preview(showBackground = true,
68.     name = "My Preview",
69. )
70. @Composable
71. fun BirthdayCardPreview() {
72.     HappyBirthdayTheme {
73.         GreetingImage(
74.             message = stringResource(R1.string.happy_birthday_text),
75.             from = stringResource(R1.string.signature_text))
76.     }
77. }
```

ЗУРАГ 1: GREETING CARD

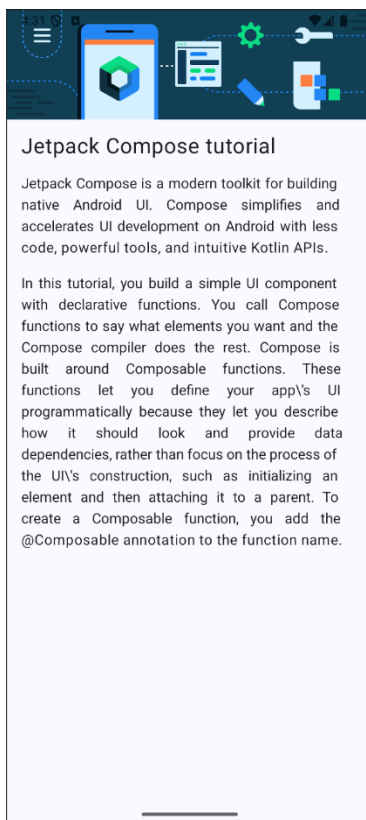
Prctatice:

Compase Basics

```
1. class MainActivity : ComponentActivity() {
2.     override fun onCreate(savedInstanceState: Bundle?) {
3.         super.onCreate(savedInstanceState)
4.         enableEdgeToEdge()
5.         setContent {
6.             TheLearnTogetherTheme {
7.                 Surface(
8.                     modifier = Modifier.fillMaxSize()
9.                 )
10.                {
11.                    BackgroundImage()
12.                }
13.            }
14.        }
15.    }
16. }
17.
18.
19. @Composable
20. fun TutorialText(modifier: Modifier = Modifier) {
```

```
21. Column() {
22.     Text(
23.         text = "Jetpack Compose tutorial",
24.         fontSize = 24.sp,
25.         modifier = Modifier
26.             .padding(16.dp)
27.
28.
29.     )
30.     Text(
31.         text = "Jetpack Compose is a modern toolkit for building native Android UI. Compose
simplifies and accelerates UI development on Android with less code, powerful tools, and
intuitive Kotlin APIs.",
32.         modifier = Modifier
33.             .padding(start = 16.dp, end = 16.dp),
34.         textAlign = TextAlign.Justify
35.
36.
37.     )
38.     Text(
39.         text = "In this tutorial, you build a simple " +
40.             "UI component with declarative functions." +
41.             " You call Compose functions to say what elements you " +
42.             "want and the Compose compiler does the rest. Compose " +
43.             "is built around Composable functions. These functions " +
44.             "let you define your app\\'s UI programmatically because" +
45.             " they let you describe how it should look and provide " +
46.             "data dependencies, rather than focus on the process of " +
47.             "the UI\\'s construction, such as initializing an element" +
48.             " and then attaching it to a parent. To create a Composable " +
49.             "function, you add the @Composable annotation to the function name.",
50.         modifier = Modifier
51.             .padding(16.dp),
52.         textAlign = TextAlign.Justify
53.
54.
55.     )
56. }
57. }
58.
59.
60. @Composable
61. fun BackgroundImage( modifier: Modifier = Modifier){
62.     val image = painterResource(R.drawable.bg_compose_background)
63.     Column(){
```

```
64.     Image(  
65.         painter = image,  
66.         contentDescription = null,  
67.  
68.  
69.     )  
70.     TutorialText(  
71.         modifier = Modifier  
72.             .fillMaxWidth()  
73.  
74.  
75.     )  
76. }  
77. }  
78.  
79.  
80. @Preview(showBackground = true)  
81. @Composable  
82. fun GreetingPreview() {  
83.     TheLearnTogetherTheme {  
84.         BackgroundImage()  
85.     }  
86. }  
87.
```



ЗУРАГ 2: COMPOSE BASICS

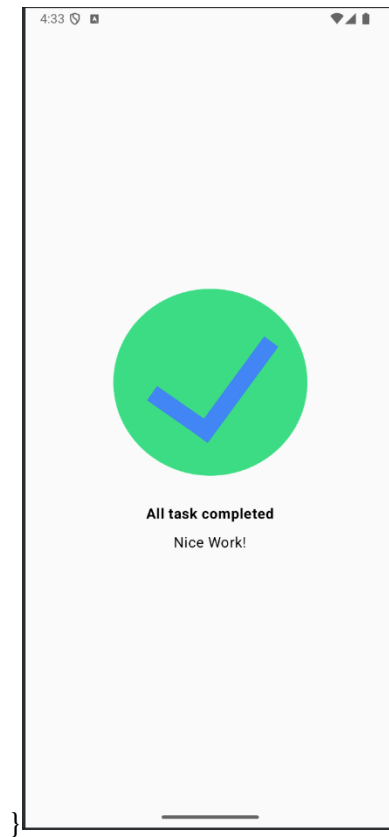
Task Manager

```

1. class MainActivity : ComponentActivity() {
2.     override fun onCreate(savedInstanceState: Bundle?) {
3.         super.onCreate(savedInstanceState)
4.         enableEdgeToEdge()
5.         setContent {
6.             TaskManagerTheme {
7.                 AllTaskCompletedScreen(
8.
9.                 )
10.            }
11.        }
12.    }
13. }
14.
15.
16. @Composable
17. fun AllTaskCompletedScreen(modifier: Modifier = Modifier){
18.     val image = painterResource(R.drawable.ic_task_completed)

```

```
19. Column(  
20.     verticalArrangement = Arrangement.Center,  
21.     horizontalAlignment = Alignment.CenterHorizontally,  
22.     modifier = Modifier  
23.         .fillMaxHeight()  
24. ) {  
25.     Image(  
26.         painter = image,  
27.         contentDescription = null,  
28.         modifier = Modifier  
29.             .fillMaxWidth()  
30.     )  
31.     Text(  
32.         text = "All task completed",  
33.         fontWeight = FontWeight.Bold,  
34.         modifier = Modifier  
35.             .padding(top = 24.dp, bottom = 8.dp)  
36.     )  
37.     Text(  
38.         text = "Nice Work!",  
39.         fontSize = 16.sp  
40.     )  
41. }  
42. }  
43.  
44.  
45. @Preview(showBackground = true)  
46. @Composable  
47. fun GreetingPreview() {  
48.     TaskManagerTheme {  
49.         AllTaskCompletedScreen()  
50.     }  
51. }
```



ЗУРАГ 3: TASK MANAGER

Compose Quadrant

```

1. class MainActivity : ComponentActivity() {
2.     override fun onCreate(savedInstanceState: Bundle?) {
3.         super.onCreate(savedInstanceState)
4.         enableEdgeToEdge()
5.         setContent {
6.             ComposeQuadrantTheme {
7.
8.             }
9.         }
10.    }
11. }
12.
13. @Composable
14. fun ComposeCard(){
15.     Column(modifier = Modifier
16.         .fillMaxHeight()
17.         .fillMaxWidth(),
18.         verticalArrangement = Arrangement.Center,

```

```
19.     horizontalAlignment = Alignment.CenterHorizontally
20. ) {
21.     Row (modifier = Modifier
22.         .weight(1f).
23.         fillMaxHeight(),
24.     ){
25.         Column(
26.             modifier = Modifier.weight(1f)
27.             .fillMaxHeight()
28.             .background(color = Color(0xFFEADDFF))
29.             .padding(16.dp),
30.             verticalArrangement = Arrangement.Center,
31.             horizontalAlignment = Alignment.CenterHorizontally,
32.
33.         ) {
34.             Text(
35.                 modifier = Modifier.padding(bottom = 16.dp),
36.                 text = "Text composable",
37.                 fontWeight = FontWeight.Bold
38.
39.             )
40.             Text(
41.                 text = "Displays text and follows the recommended Material Design guidelines.\n",
42.                 textAlign = TextAlign.Justify,
43.                 fontSize = 14.sp
44.             )
45.         }
46.         Column(
47.             modifier = Modifier.weight(1f)
48.             .fillMaxHeight()
49.             .background(color = Color(0xFFD0BCFF))
50.             .padding(16.dp),
51.             verticalArrangement = Arrangement.Center,
52.             horizontalAlignment = Alignment.CenterHorizontally
53.         ) {
54.             Text(
55.                 modifier = Modifier.padding(bottom = 16.dp),
56.                 text = "Image composable",
57.                 fontWeight = FontWeight.Bold
58.             )
59.             Text(
60.                 text = "Creates a composable that lays out and draws a given Painter class object.\n",
61.                 textAlign = TextAlign.Justify,
62.                 fontSize = 14.sp
63.             )
```

```

64.     }
65.   }
66.   Row(modifier = Modifier.weight(1f).fillMaxHeight()
67.   ) {
68.     Column(
69.       modifier = Modifier.weight(1f)
70.       .fillMaxHeight()
71.       .background(color = Color(0xFFB69DF8))
72.       .padding(16.dp),
73.       verticalArrangement = Arrangement.Center,
74.       horizontalAlignment = Alignment.CenterHorizontally,
75.     ) {
76.       Text(
77.         modifier = Modifier.padding(bottom = 16.dp),
78.         text = "Row composable",
79.         fontWeight = FontWeight.Bold
80.       )
81.       Text(
82.         text = "A layout composable that places its children in a horizontal sequence.\n",
83.         textAlign = TextAlign.Justify,
84.         fontSize = 14.sp
85.       )
86.     }
87.   }
88.   Column(
89.     modifier = Modifier.weight(1f)
90.     .fillMaxHeight()
91.     .background(color = Color(0xFFF6EDFF))
92.     .padding(16.dp),
93.     verticalArrangement = Arrangement.Center,
94.     horizontalAlignment = Alignment.CenterHorizontally,
95.   ) {
96.     Text(
97.       modifier = Modifier.padding(bottom = 16.dp),
98.       text = "Column composable",
99.       fontWeight = FontWeight.Bold
100.    )
101.    Text(
102.      text = "A layout composable that places its children in a vertical sequence.",
103.      textAlign = TextAlign.Justify,
104.      fontSize = 14.sp
105.    )
106.  }
107. }
108. }

```



```
109.     }
110.
111.     @Preview(showBackground = true)
112.     @Composable
113.     fun GreetingPreview() {
114.         ComposeQuadrantTheme {
115.             ComposeCard()
116.         }
117.     }
```



ЗУРАГ 4: COMPOSE QUADRANT

Project: Create a Business Card App

Quiz

1. What is Jetpack Compose?
 - a. A Modern toolkit to develop Android UI
 - b. Toolkit to design libraries
 - c. Database Interface
 - d. Plugin to build APK

2. Composable functions are the basic building block of Compose.
 - a. **True**
 - b. False
3. What annotation is used to annotate a Composable function?
 - a. @Annotation
 - b. @ComposableFunction
 - c. **@Composable**
 - d. @Preview
4. The basic standard layout elements in Compose are:
 - a. *Choose as many answers as you see fit.*
 - b. **Column**
 - c. **Row**
 - d. Text
 - e. **Box**
5. What is the tool window for importing, creating, managing, and using resources in your app?
 - a. Application Manager
 - b. **Resource Manager**
 - c. Resource Tool
 - d. Layout Manager
6. Which class is an automatically generated class by Android that contains the IDs of all resources in the project.
 - a. The Android class
 - b. The Resource class
 - c. **The R Class**
 - d. The ResourceID class
7. Which function is used is to load a drawable image resource.
 - a. The stringResource() function
 - b. **The painterResource() function**
 - c. The ImageResource() function
 - d. The loadResource() function
8. What is the function parameter used to add accessibility text, used by talkback?
 - a. accessibilityText
 - b. contentText

- c. accessibilityDescription
 - d. contentDescription
9. The Box layout stacks the UI elements on top of one another.
- a. True
 - b. False
10. What parameter is used to align the child element to the beginning of the parent?
- a. Alignment.End
 - b. Alignment.Begin
 - c. Alignment.Start
 - d. Alignment.Top