

Програмування мобільних пристроїв

Архітектурний шаблон "Модель-представлення-контролер (MVC)" та його використання в Android

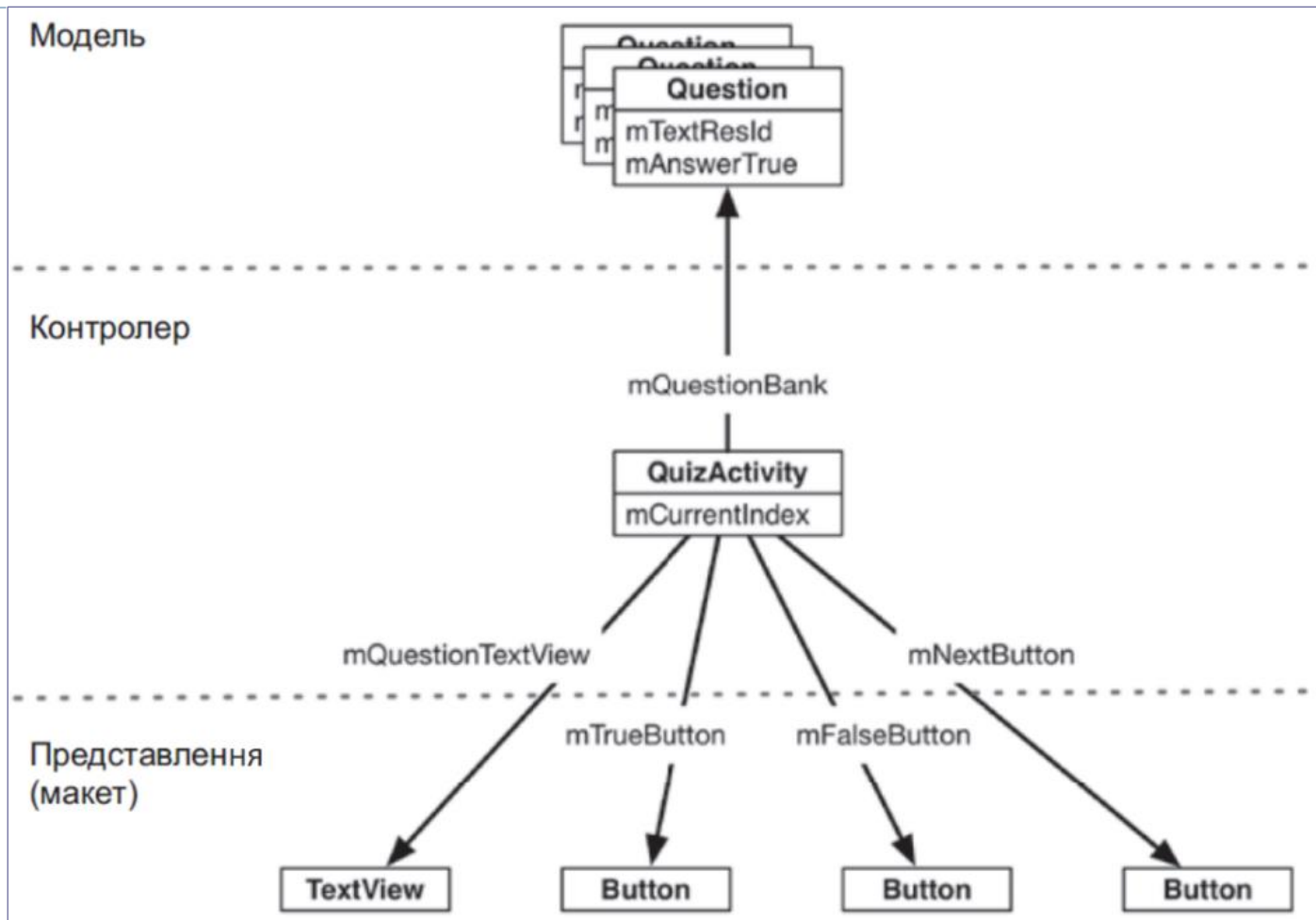
Слайди до лекцій (3 змістовий модуль)

Розробка архітектури застосунку - архітектура MVC

- Об'єкти **рівня моделі** містять дані, які обробляються/використовуються програмою. Класи моделі зазвичай проектуються для моделювання сутностей, з якими працює застосунок: користувач, товар, запитання тесту тощо.
 - Об'єкти **рівня представлення** можуть відображати себе на екрані і реагувати на дії користувача. Рівень представлення GeoQuiz складається з віджетів, заповнених відповідно вмісту файлу `res/layout/activity_main.xml`.
 - Об'єкти **рівня контролерів** пов'язують об'єкти рівня представлення та рівня моделі, вони містять *логіку застосування*. Контролери реагують на різні події, ініційовані об'єктами-представленнями, і керують потоками даних між об'єктами моделі і об'єктами-представленнями. В Android контролер зазвичай є підкласом Activity або Fragment. Рівень контролера GeoQuiz зараз складається тільки з класу MainActivity.
-



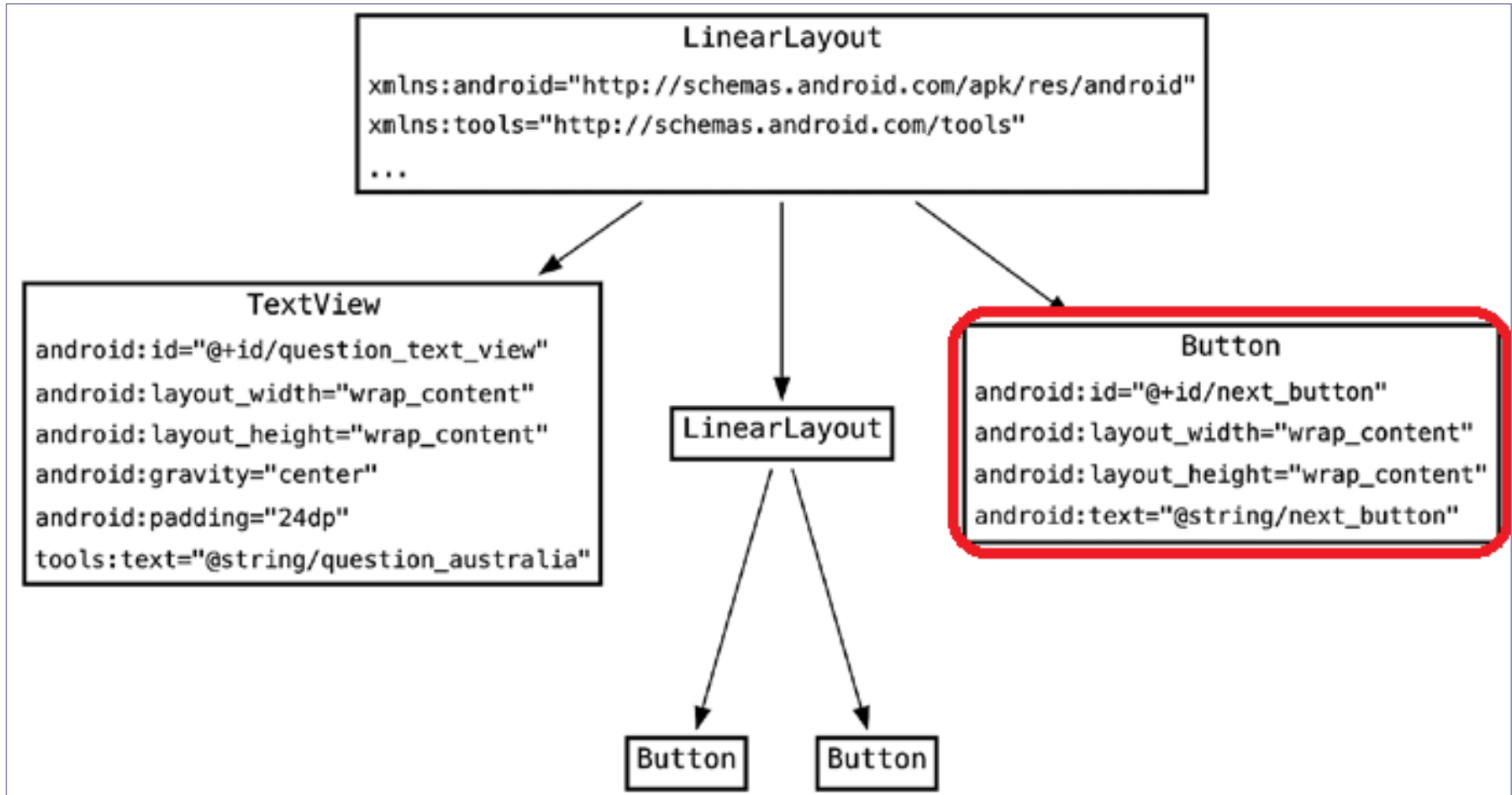
Розробка архітектури застосунку - додання класу рівня моделі



```
data class Question(@StringRes val textResId: Int, val answer: Boolean)
```

Клас рівня моделі

Розробка архітектури застосунку - оновлення рівня представлення



► Доданий віджет-кнопка для переходу до наступного питання

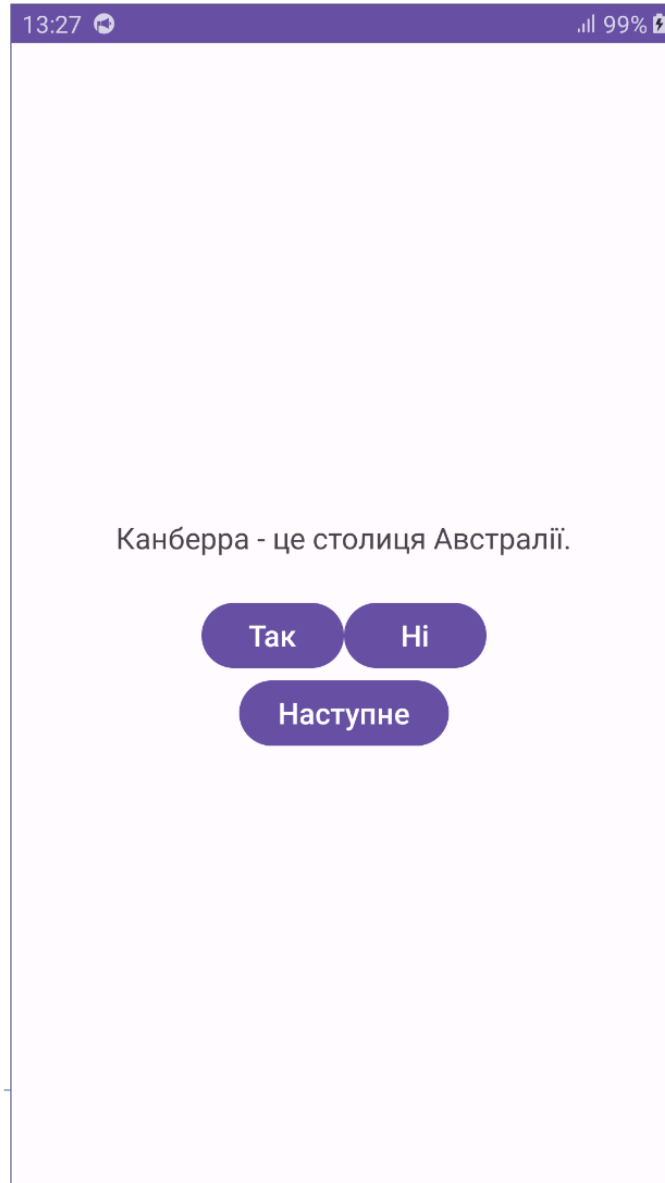
Розробка архітектури застосунку - оновлення рівня представлення

Додані до `values/strings.xml` ресурси-рядки:

```
<resources>  
  <string name="app_name" translatable="false">GeoQuiz</string>  
  <string name="true_button">True</string>  
  <string name="false_button">False</string>  
  <string name="correct_toast">Correct!</string>  
  <string name="incorrect_toast">Incorrect!</string>  
  <string name="next_button" translatable="false">&gt;</string>  
  <string name="question_australia">Canberra is the capital of Australia</string>  
  <string name="question_oceans">The Pacific Ocean is larger than the Atlantic  
    Ocean</string>  
  <string name="question_mideast">The Suez Canal connects the Red Sea and  
    the Indian Ocean</string>  
  <string name="question_africa">The source of the Nile River is in Egypt</string>  
  <string name="question_americas">The Amazon River is the longest river  
    in the Americas</string>  
  <string name="question_asia">Lake Baikal is the world's oldest and deepest  
    freshwater lake</string>  
</resources>
```

▶ Відповідні переклади додані до `values-uk/strings.xml`

Розробка архітектури застосунку - оновлення рівня представлення



Розробка архітектури застосунку - оновлення рівня контролерів

- Створимо список **questionBank** об'єктів Question та змінну **currentIndex** для індексу списку, які використовуватимуться для виведення на екран питань та перевірки правильності відповіді користувача;
- Додаймо змінні-властивості для текстового поля з питаннями – **questionTextView** та кнопки Next – **nextButton**;
- До функції onCreate(savedInstanceState: Bundle?) додаймо код пошуку доданих віджетів за їх ідентифікаторами;
- Створимо функцію **updateQuestion()**, яка встановлюватиме текст питання у віджет за значенням поточного індексу списку питань;



Розробка архітектури застосунку - оновлення рівня контролерів

- Додаймо виклик `updateQuestion()` у кінець функції `onCreate(savedInstanceState: Bundle?)` – для завантаження першого питання на початку роботи програми;
- Додаймо виклик `updateQuestion()` до доданого до віджету кнопки `Next` обробника події натискання на цю кнопку;
- Для забезпечення функціональності перевірки правильності відповіді користувача додаймо приватну функцію `checkAnswer(userAnswer: Boolean): Int`, яка приймає посилання на використану користувачем кнопку (так або ні) та повертає ідентифікатор рядкового ресурсу відповідного повідомлення, що спливатиме. Ця функція буде задіяна у слухачах подій кнопок-відповідей разом з функцією відображення повідомлення.



Розробка архітектури застосунку - оновлення рівня контролерів

```
class MainActivity : AppCompatActivity() {  
    ...  
    private lateinit var nextButton: Button  
    private lateinit var questionTextView: TextView  
    private val questionBank = listOf(  
        Question(R.string.question_australia, true),  
        ...  
        Question(R.string.question_asia, true)  
    )  
    private var currentIndex = 0;  
    override fun onCreate(savedInstanceState: Bundle?) {  
        ...  
        nextButton = findViewById(R.id.next_button)  
        questionTextView = findViewById(R.id.question_text_view)  
        trueButton.setOnClickListener { view: View ->  
            showToast(checkAnswer(true))  
        }  
        falseButton.setOnClickListener { view: View ->  
            showToast(checkAnswer(false))  
        }  
    }  
    ...  
}
```

Розробка архітектури застосунку - оновлення рівня контролерів

```
...
    nextButton.setOnClickListener {
        currentIndex = (currentIndex + 1) % questionBank.size
        updateQuestion()
    }
    updateQuestion()
}

private fun updateQuestion() {
    val questionTextResId = questionBank[currentIndex].textResId
    questionTextView.setText(questionTextResId)
}

private fun checkAnswer(userAnswer: Boolean): Int {
    val correctAnswer = questionBank[currentIndex].answer
    val messageResId = if (userAnswer == correctAnswer) {
        return R.string.correct_toast
    } else {
        return R.string.incorrect_toast
    }
}
}
... }
```

Розробка класу активності - додання обробників подій

