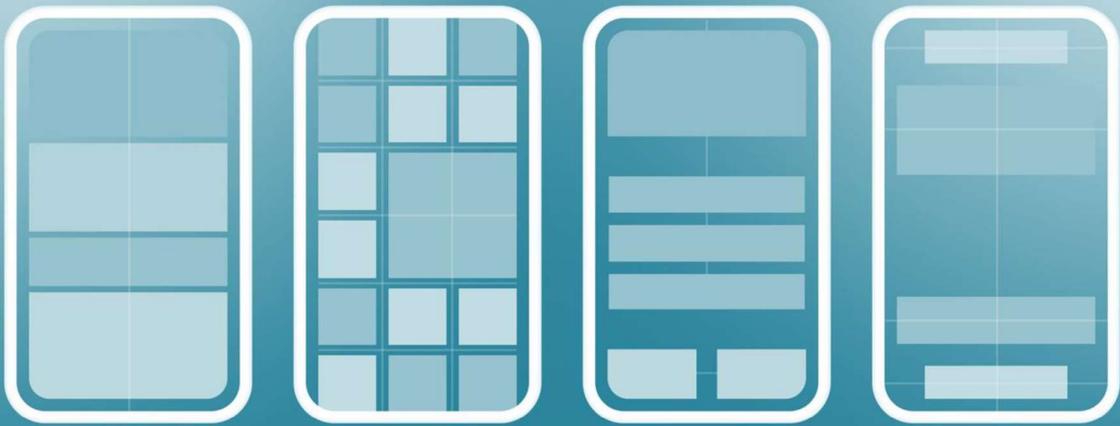


**Т.Г. Ляшук**

# **Програмування мобільних пристроїв.**

**Збірник тестових завдань на тему**

**"Контейнери та макети"**



Міністерство освіти і науки України  
Рівненський державний гуманітарний університет

**ЛЯШУК Т.Г.**

# **Програмування мобільних пристроїв.**

**Збірник тестових завдань на тему**

**"Контейнери та макети"**

Практикум

Рівне – 2026

УДК 004.4

Л 99

Затверджено на засіданні кафедри  
інформаційних систем та моделювання  
Рівненського державного гуманітарного університету  
(протокол № 1 від 27 січня 2026 р.)

Схвалено навчально-методичною комісією  
факультету математики та інформатики  
Рівненського державного гуманітарного університету  
(протокол № 1 від 27 січня 2026 р.)

**Рецензенти:**

**Бабич С.М.** – кандидат технічних наук, доцент кафедри інформаційних технологій та моделювання.

**Шроль Т.С.** – кандидат педагогічних наук, доцент кафедри цифрових технологій та методики навчання інформатики.

**Ляшук Т.Г.**

**Програмування мобільних пристроїв. Збірник тестових завдань на тему "Контейнери та макети" / Т.Г. Ляшук. – Рівне: РДГУ, 2026. – 62 с.**

Збірник тестових завдань розроблено для поточного та підсумкового контролю знань студентів з дисципліни «Програмування мобільних пристроїв». У збірнику охоплено теми компонування візуального інтерфейсу через контейнери та макети, серед яких: відносні, лінійні, табличні, списки тощо.

Матеріали подані у формі тестових завдань різного рівня складності. Такий підхід сприяє ефективному закріпленню теоретичних знань і практичних навичок у сфері мобільної розробки. Практикум може бути використаний як для самоперевірки, так і в аудиторній роботі для студентів, які навчаються за спеціальностями представленими галуззю знань «Інформаційні технології».

© Ляшук Т.Г., 2026

© РДГУ, 2026

## ЗМІСТ

ВСТУП.....	4
Тема 1. Огляд Android-макетів/контейнерів.....	5
Тема 2. ConstraintLayout.....	10
Тема 3. LinearLayout.....	15
Тема 4. RelativeLayout.....	20
Тема 5. FrameLayout.....	25
Тема 6. TableLayout.....	30
Тема 7. GridLayout.....	35
Тема 8. ScrollView.....	39
Тема 9. ListView.....	44
Тема 10. GridView.....	48
Тема 11. RecyclerView.....	52
ВІДПОВІДІ.....	57
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....	61

## ВСТУП

Сучасна розробка мобільних застосунків нерозривно пов'язана зі створенням гнучких, адаптивних та ефективних інтерфейсів користувача. Основою для побудови таких інтерфейсів є використання різноманітних контейнерів та макетів, які визначають структуру, розміщення та взаємодію візуальних елементів на екранах мобільних пристроїв. Розуміння принципів їхньої роботи є фундаментальним для майбутніх фахівців у сфері інформаційних технологій, які прагнуть створювати якісні та зручні мобільні рішення.

Знання особливостей різних типів контейнерів/макетів є необхідною передумовою для розробників, які ставлять за мету створення інтуїтивно зрозумілих, продуктивних та візуально привабливих застосунків. Тому, детальне вивчення таких тем, як макети, контейнери для прокручування та відображення колекцій даних, а також їхні атрибути і методи керування є досить важливим.

Даний збірник тестових завдань створено з метою забезпечення перевірки знань студентів з ключових аспектів використання контейнерів та макетів у розробці мобільних застосунків. До збірника включено різноманітні тести – від базових до більш складних, які охоплюють принципи організації інтерфейсу, характеристики основних макетів (ConstraintLayout, LinearLayout тощо), відображення списків та сіток (ListView, GridView тощо), а також особливості їх застосування при створенні адаптивних макетів. Завдання представлені у форматах одиничного () і множинного вибору (), а також відкритих текстових відповідей ()

Практикум стане в нагоді як під час лабораторних занять, так і для самостійної підготовки, контрольних робіт та тематичного оцінювання. Він покликаний сприяти поглибленню теоретичних знань студентів та формуванню практичних навичок у галузі проектування та реалізації інтерфейсів мобільних застосунків за допомогою різноманітних макетів і контейнерів.

# Тема 1. Огляд Android-макетів/контейнерів

---

- Для чого використовуються контейнери в Android-інтерфейсі?
  - Для зберігання зображень;
  - Для створення анімацій;
  - Для впорядкування та розміщення елементів UI;
  - Для створення бази даних.
  
- Які функції зазвичай виконують контейнери у мобільному інтерфейсі?
  - Рендеринг графіки;
  - Визначення простору між елементами;
  - Упорядкування вкладених компонентів;
  - Розміщення елементів за певною логікою.
  
-  Що таке «макет» у контексті Android-розробки?

Ваша відповідь: \_\_\_\_\_
  
- Що визначає властивість `layout_width` в елементі інтерфейсу?
  - Висоту елемента;
  - Колір тла;
  - Ширину елемента в макеті;
  - Відстань до екрана.
  
- Які типи значень можна використовувати для визначення відступів?

- a) dp (density-independent pixels);
- b) sp (scale-independent pixels);
- c) mm;
- d) px (pixels only).

6.  Який атрибут відповідає за внутрішній відступ елемента?

Ваша відповідь: \_\_\_\_\_

7.  Що таке «margin» в макетах Android?

- a) Колір тла;
- b) Зовнішній відступ елемента;
- c) Висота тексту;
- d) Прозорість елемента.

8.  Які з перелічених одиниць виміру використовують для розмітки інтерфейсу?

- a) dp;
- b) GHz;
- c) pt;
- d) sp.

9.  Який тип відступу впливає на розташування елемента відносно інших?

Ваша відповідь: \_\_\_\_\_

10.  Що таке «padding» у контексті UI?

- a) Внутрішній простір між вмістом і межами елемента;

- b) Зовнішній простір екрана;
- c) Назва стилю;
- d) Одиниця виміру.

11.  **Які переваги дає використання макетів при створенні інтерфейсу?**

- a) Гнучкість при зміні розміру екрана;
- b) Заміна шрифтів;
- c) Чітка структура елементів;
- d) Можливість вкладення інших макетів.

12.  **Який відступ застосовується всередині самого елемента?**

Ваша відповідь: \_\_\_\_\_

13.  **Який принцип використовується при розташуванні елементів у макетах?**

- a) Ієрархія вкладених блоків;
- b) Рандомне вирівнювання;
- c) Плавне прокручування;
- d) Індксація фону.

14.  **Які атрибути належать до групи розміщення у макетах?**

- a) src;
- b) layout\_width;
- c) layout\_margin;
- d) layout\_height.

15.  Яка одиниця виміру вважається рекомендованою для тексту в UI?

Ваша відповідь: \_\_\_\_\_

16.  Що станеться, якщо встановити `layout_width="match_parent"`?

- a) Елемент отримає випадкову ширину;
- b) Елемент не відобразиться;
- c) Елемент займе всю доступну ширину контейнера;
- d) Встановиться ширина екрана в пікселях.

17.  Які з варіантів задають відступи зовнішнього простору?

- a) `fontFamily`;
- b) `layout_marginTop`;
- c) `textSize`;
- d) `layout_marginStart`.

18.  Яка властивість визначає висоту елемента в макеті?

Ваша відповідь: \_\_\_\_\_

19.  Що визначає значення `wrap_content` у макетах?

- a) Елемент буде прихований;
- b) Елемент отримає максимальний розмір;
- c) Елемент буде мати розмір відповідно до свого вмісту;
- d) Елемент отримає фіксовану ширину.

20.  Які властивості впливають на розміри елемента?

- a) layout\_height;
- b) layout\_width;
- c) orientation;
- d) gravity.

## Тема 2. ConstraintLayout

---

- Що є основною метою використання ConstraintLayout у розробці інтерфейсів Android?**
  - Підтримка анімацій;
  - Гнучке позиціонування елементів з використанням обмежень;
  - Побудова списків;
  - Прискорення компіляції.
- Які з переваг притаманні ConstraintLayout порівняно з іншими макетами?**
  - Зменшення глибини ієрархії в інтерфейсі;
  - Підтримка лише горизонтального вирівнювання;
  - Підтримка взаємного позиціонування елементів;
  - Покращення продуктивності у складних макетах.
- Як називається спеціальна лінія, яку можна використовувати для вирівнювання кількох елементів у ConstraintLayout?**

Ваша відповідь: \_\_\_\_\_
- Що станеться, якщо не задати жодного обмеження для елемента в ConstraintLayout?**
  - Він автоматично вирівняється по центру;
  - Він займе весь екран;
  - Він буде розміщений у лівому верхньому куті;
  - Він не відобразиться на екрані.
- Які з перелічених типів обмежень підтримує ConstraintLayout?**

- a) Вирівнювання по краях іншого елемента;
- b) Відстань до батьківського елемента;
- c) Прив'язка до базової лінії тексту;
- d) Прив'язка до бази даних.

6.  **Як називається компонент, який дозволяє групувати декілька елементів для спільного керування їх властивостями в ConstraintLayout?**  
Ваша відповідь: \_\_\_\_\_

7.  **Який атрибут використовується для встановлення відступу між елементами у ConstraintLayout?**

- a) padding;
- b) marginTop;
- c) layout\_margin;
- d) spacing.

8.  **Які способи можна використовувати для створення обмежень у ConstraintLayout у Android Studio?**

- a) Перетягування стрілок у Design View;
- b) Редагування XML вручну;
- c) Плагіни сторонніх розробників;
- d) Constraint Widget у редакторі атрибутів.

9.  **Яка з властивостей дозволяє змусити елемент заповнити простір між двома обмеженнями?**

- a) wrap\_content;
- b) 0dp (match constraints);

- c) match\_parent;
- d) fill\_parent.

10.  Як називається механізм у ConstraintLayout, що дозволяє створити вертикальний або горизонтальний ланцюжок елементів?

Ваша відповідь: \_\_\_\_\_

11.  Які типи ланцюжків (Chains) підтримує ConstraintLayout?

- a) Packed;
- b) Spread;
- c) Spread Inside;
- d) Zigzag.

12.  Що таке «Barrier» у ConstraintLayout?

- a) Група для стилізації елементів;
- b) Динамічний обмежувач, який враховує розмір кількох елементів;
- c) Лінія розмітки;
- d) Компонент для взаємодії з користувачем.

13.  Яке значення ширини або висоти задається, щоб увімкнути режим «match constraints»?

Ваша відповідь: \_\_\_\_\_

14.  Які об'єкти можна використовувати для позиціонування елементів у ConstraintLayout?

- a) Guideline;
- b) Barrier;

- c) Layer;
- d) Anchor points інших елементів.

15.  **Яка з переваг відрізняє ConstraintLayout від FrameLayout?**

- a) Менше функцій;
- b) Можливість складного позиціювання без вкладеності;
- c) Краща підтримка медіа;
- d) Перевага у прокручуванні.

16.  **Які переваги дає використання «Guideline» у макеті?**

- a) Покращення узгодженості вирівнювання елементів;
- b) Можливість адаптації до різних екранів;
- c) Анімація об'єктів;
- d) Гнучке позиціювання без прив'язки до конкретних елементів.

17.  **Яка властивість визначає вертикальне вирівнювання елемента відносно іншого у ConstraintLayout?**

Ваша відповідь: \_\_\_\_\_

18.  **Як можна вирівняти текстові поля по вертикалі в ConstraintLayout?**

- a) Через frame розмітку;
- b) Через фіксовані координати;
- c) Через ланцюжки (chains) або обмеження (constraints);
- d) Через marginLeft.

19.  **Які з наступних властивостей належать до системи обмежень ConstraintLayout?**

- a) `layout_constraintStart_toStartOf`;
- b) `layout_constraintEnd_toEndOf`;
- c) `android:onClick`;
- d) `layout_constraintTop_toBottomOf`.

20.  Як називається режим, при якому елемент вирівнюється автоматично між двома іншими та рівномірно розтягується?

Ваша відповідь: \_\_\_\_\_

## Тема 3. LinearLayout

---

- Яка з наведених ОС є найпопулярнішою серед мобільних пристроїв станом на 2025 рік?**
  - Android;
  - iOS;
  - Windows Phone;
  - Symbian.
  
- Що таке LinearLayout у Android-розробці?**
  - Контейнер для відображення списків;
  - Контейнер, що розміщує елементи в ряд або стовпчик;
  - Контейнер для багаторівневої навігації;
  - Віджет для виводу тексту.
  
- Які властивості впливають на розташування дочірніх елементів у LinearLayout?**
  - orientation;
  - textStyle;
  - gravity;
  - onClick.
  
-  **Яка властивість визначає напрямок розміщення елементів у LinearLayout?**

Ваша відповідь: \_\_\_\_\_

Яке значення `orientation` потрібно встановити, щоб елементи розміщувались зверху вниз?

- a) `horizontal`;
- b) `diagonal`;
- c) `vertical`;
- d) `stack`.

5.  Які значення можна використовувати для властивості `orientation`?

- a) `horizontal`;
- b) `vertical_top`;
- c) `vertical`;
- d) `right_to_left`.

6.  Який атрибут дозволяє встановити вирівнювання дочірніх елементів усередині `LinearLayout`?

Ваша відповідь: \_\_\_\_\_

7.  Як поведуться елементи в `LinearLayout` з `orientation="horizontal"`?

- a) Вони накладаються один на одного;
- b) Вони відображаються вертикально;
- c) Вони розташовуються зліва направо;
- d) Вони зникають.

8.  Які атрибути можуть впливати на розміри елементів у `LinearLayout`?

- a) `layout_width`;
- b) `layout_height`;
- c) `textColor`;

d) layout\_weight.

9.  **Яка властивість дозволяє поділити простір між елементами за певними пропорціями?**

Ваша відповідь: \_\_\_\_\_

10.  **Що станеться, якщо для всіх елементів LinearLayout встановити layout\_width="0dp" і призначити layout\_weight?**

- a) Елементи не з'являться;
- b) Ширина буде розподілена пропорційно згідно з вагами;
- c) Вони займуть повну ширину незалежно від ваги;
- d) Вони зникнуть.

11.  **Які з переваг має використання LinearLayout?**

- a) Простота у використанні;
- b) Прогнозоване розташування елементів;
- c) Динамічне завантаження з API;
- d) Зрозумілий порядок у верстці.

12.  **Який атрибут LinearLayout можна використати для визначення внутрішнього відступу елементів?**

Ваша відповідь: \_\_\_\_\_

13.  **Яке обмеження є у використанні вкладених LinearLayout?**

- a) Вони завжди вирівнюються по центру;
- b) Вони не підтримуються;
- c) Вони займають менше пам'яті;

d) Вони можуть погіршити продуктивність при глибокій ієрархії.

14.  Які атрибути `LinearLayout` використовуються для вирівнювання дочірніх елементів?

- a) `gravity`;
- b) `layout_gravity`;
- c) `orientation`;
- d) `minHeight`.

15.  Яке значення `gravity` вирівнює елементи по центру?

Ваша відповідь: \_\_\_\_\_

16.  Що робить атрибут `layout_gravity` в `LinearLayout`?

- a) Задає фоновий колір;
- b) Визначає вирівнювання елемента в межах `LinearLayout`;
- c) Встановлює `padding`;
- d) Вказує розміри шрифту.

17.  Які значення підтримуються для властивості `gravity`?

- a) `center`;
- b) `left`;
- c) `textStart`;
- d) `right`.

18.  Яке значення `layout_weight` слід вказати, щоб усі елементи займали однакову ширину?

Ваша відповідь: \_\_\_\_\_

19.  Який тип компонування НЕ підтримується безпосередньо **LinearLayout**?

- a) По вертикалі;
- b) По горизонталі;
- c) По діагоналі;
- d) Вкладене розміщення.

20.  Які підходи можна використати для адаптації **LinearLayout** до різних екранів?

- a) Використання `layout_weight`;
- b) Фіксовані розміри в `px`;
- c) `dp` і `sp` одиниці виміру;
- d) Конфігурації з ресурсами для різних розмірів екрана.

## Тема 4. RelativeLayout

---

1.  Що таке RelativeLayout у Android?

- a) Контейнер для списків;
- b) Макет, у якому елементи розташовуються відносно інших елементів або батьківського контейнера;
- c) Контейнер з фіксованим позиціонуванням;
- d) Спеціальний тип адаптера.

2.  Які властивості використовуються для розміщення елементів у RelativeLayout?

- a) layout\_alignParentTop;
- b) layout\_toRightOf;
- c) android:text;
- d) layout\_below.

3.  Яка властивість використовується, щоб розмістити елемент під іншим елементом у RelativeLayout?

Ваша відповідь: \_\_\_\_\_

4.  Що станеться, якщо два елементи отримають однакову позицію без додаткових обмежень?

- a) Вони відобразяться по черзі;
- b) Вони можуть перекривати один одного;
- c) Вони автоматично вирівнюються;
- d) Один із них зникне.

5.  Які з цих властивостей дозволяють вирівнювати елемент по краях контейнера?

- a) `layout_alignParentLeft`;
- b) `layout_alignToRight`;
- c) `layout_alignParentBottom`;
- d) `layout_fixedEdge`.

6.  Який атрибут дозволяє вирівняти елемент по центру горизонталі батьківського контейнера?

Ваша відповідь: \_\_\_\_\_

7.  Як можна вирівняти елемент по центру батьківського контейнера у `RelativeLayout`?

- a) `android:gravity="center"`;
- b) `layout_centerInParent="true"`;
- c) `alignSelf="center"`;
- d) `parentAlign="center"`.

8.  Які з наведених властивостей дозволяють позиціювати елемент щодо іншого елемента?

- a) `layout_above`;
- b) `layout_floatRight`;
- c) `layout_toLeftOf`;
- d) `layout_alignBaseline`.

9.  Яке значення потрібно встановити, щоб увімкнути центр вертикалі в `RelativeLayout`?

Ваша відповідь: \_\_\_\_\_

**Правильна відповідь: `layout_centerVertical="true"`.**

10.  **Яке основне обмеження `RelativeLayout` порівняно з `ConstraintLayout`?**
- a) Краще розміщення;
  - b) Менше гнучкості у складних інтерфейсах;
  - c) Повільніший рендеринг;
  - d) Не підтримує адаптивність.
11.  **Які проблеми можуть виникати при великій кількості вкладених `RelativeLayout`?**
- a) Зниження продуктивності;
  - b) Ускладнення читабельності коду;
  - c) Випадкові помилки при компіляції;
  - d) Ускладнення обслуговування макету.
12.  **Який атрибут відповідає за розміщення елемента зліва від іншого елемента?**
- Ваша відповідь: \_\_\_\_\_
13.  **Що означає атрибут `layout_alignParentRight="true"`?**
- a) Елемент буде вирівняний по правому краю контейнера;
  - b) Елемент отримає `padding` справа;
  - c) Елемент вирівняється по центру правого елемента;
  - d) Нічого не зміниться.
14.  **Які елементи можна вирівняти у `RelativeLayout` за базовою лінією?**

- a) TextView;
- b) FrameLayout;
- c) EditText;
- d) Button.

15.  Який тип розміщення дозволяє уникнути перекриття елементів у **RelativeLayout**?

Ваша відповідь: \_\_\_\_\_

16.  Що станеться, якщо не вказати жодного позиціювання для елемента в **RelativeLayout**?

- a) Він буде знищений;
- b) Він буде розміщений у лівому верхньому куті;
- c) Він буде центруватися;
- d) Він отримає нульовий розмір.

17.  Які значення можуть приймати **boolean**-атрибути позиціювання в **RelativeLayout**?

- a) true;
- b) "yes";
- c) false;
- d) 0.

18.  Який атрибут вирівнює елемент по базовій лінії іншого елемента?

Ваша відповідь: \_\_\_\_\_

19.  У якому випадку використовують **RelativeLayout** замість **LinearLayout**?

- a) Коли потрібне рівномірне вирівнювання;
- b) Коли потрібне позиціонування відносно інших елементів;
- c) Коли елементи мають однакову вагу;
- d) Коли потрібен список.

20.  **Які способи покращити продуктивність при використанні RelativeLayout?**

- a) Уникати глибокої вкладеності;
- b) Переходити на ConstraintLayout для складних екранів;
- c) Використовувати fixed layout;
- d) Оптимізувати обмеження між елементами.

# Тема 5. FrameLayout

---

1.  Що таке FrameLayout у Android?

- a) Макет для прокручуваних екранів;
- b) Макет, що показує лише один елемент поверх інших;
- c) Контейнер для списків;
- d) Спеціальний адаптер.

2.  Які з характеристик притаманні FrameLayout?

- a) Накладання елементів один на одного;
- b) Всі елементи обов'язково вирівнюються по центру;
- c) Показ останнього доданого елемента на передньому плані;
- d) Простота структури.

3.  Яка особливість поведінки дочірніх елементів у FrameLayout без позиціонування?

Ваша відповідь: \_\_\_\_\_

4.  Який елемент відображається останнім у FrameLayout?

- a) Перший доданий;
- b) Той, що має більший ID;
- c) Останній доданий;
- d) Найменший за розміром.

5.  Для чого зазвичай використовують FrameLayout?

- a) Для створення екранів-завантажень (splash screens);
- b) Для відображення одного елемента за раз;

- c) Для складної сітки;
- d) Для накладання елементів (наприклад, тексту поверх зображення).

6.  Який атрибут задає вирівнювання дочірнього елемента у `FrameLayout`?

Ваша відповідь: \_\_\_\_\_

7.  Що трапиться, якщо у `FrameLayout` додати кілька елементів без позиціонування?

- a) Вони не відобразяться;
- b) Вони вирівнюються по центру;
- c) Вони накладатимуться один на одного у верхньому лівому куті;
- d) Вони розташуються вертикально.

8.  Які значення можна використовувати для `layout_gravity` у `FrameLayout`?

- a) top;
- b) background;
- c) center;
- d) bottom.

9.  Яке позиціонування дозволяє елементу залишатись на передньому плані у `FrameLayout`?

Ваша відповідь: \_\_\_\_\_

10.  Чим `FrameLayout` відрізняється від `LinearLayout`?

- a) Підтримує тільки горизонтальне вирівнювання;
- b) Підтримує накладання елементів;

- c) Додає анімації;
- d) Вимагає layout\_weight.

11.  **Які випадки краще підходять для використання FrameLayout?**

- a) Splash screen;
- b) Відображення зображення з підписом;
- c) Динамічна сітка товарів;
- d) Повноекранне відео з кнопкою "закрити" зверху.

12.  **Як зробити елемент у FrameLayout частково прозорим?**

Ваша відповідь: \_\_\_\_\_

13.  **Яка властивість визначає розміщення елемента всередині FrameLayout?**

- a) layout\_gravity;
- b) layout\_weight;
- c) orientation;
- d) frame\_align.

14.  **Які потенційні проблеми виникають при використанні кількох шарів у FrameLayout?**

- a) Перекриття елементів;
- b) Погіршення читабельності;
- c) Труднощі з доступністю;
- d) Автоматичне вирівнювання.

15.  **Як забезпечити, щоб текст не перекривав кнопку у FrameLayout?**

Ваша відповідь: \_\_\_\_\_

16.  Який атрибут дозволяє елементу займати всю ширину або висоту контейнера?

- a) layout\_weight;
- b) match\_parent;
- c) wrap\_content;
- d) gravity.

17.  Які атрибути допомагають розмістити елемент у правому нижньому куті `FrameLayout`?

- a) layout\_gravity="bottom|end";
- b) marginStart;
- c) layout\_gravity="bottom|right";
- d) layout\_marginTop.

18.  Який елемент часто розміщують поверх зображення у `FrameLayout`?

Ваша відповідь: \_\_\_\_\_

19.  Який підхід рекомендується, якщо потрібно керувати кількома шарами у `FrameLayout`?

- a) Додавати елементи у правильному порядку (від заднього до переднього);
- b) Використовувати список;
- c) Використовувати `scrollView`;
- d) Заборонити накладання.

20.  Які альтернативи варто розглянути замість `FrameLayout` у складних макетах?

- a) `ConstraintLayout`;
- b) `RelativeLayout`;
- c) `ListView`;
- d) `CoordinatorLayout`.

## Тема 6. TableLayout

---

1.  Що таке TableLayout у Android?
  - a) Сітка з прокручуванням;
  - b) Віджет для тексту;
  - c) Макет для створення табличної структури з рядків і стовпців;
  - d) Адаптер списку.
  
2.  Які елементи використовуються всередині TableLayout для створення рядків?
  - a) LinearLayout;
  - b) TableRow;
  - c) ListItem;
  - d) TextView або інші View всередині TableRow.
  
3.  Який елемент у TableLayout представляє один горизонтальний ряд?

Ваша відповідь: \_\_\_\_\_
  
4.  Як TableLayout розташовує свої дочірні елементи?
  - a) У стовпчик;
  - b) У рядках, які поділяються на стовпці;
  - c) Випадково;
  - d) В один ряд.
  
5.  Які компоненти можна вставляти у TableRow?
  - a) TextView;
  - b) TableLayout;

- c) Button;
- d) ImageView.

6.  Який атрибут дозволяє View у TableRow розтягуватися на кілька стовпців?

Ваша відповідь: \_\_\_\_\_

7.  Що станеться, якщо кількість елементів у TableRow не збігається з іншими рядками?

- a) TableLayout автоматично вирівняє по кількості найбільшого рядка;
- b) Відбудеться помилка компіляції;
- c) Рядок буде ігнорований;
- d) Елементи зникнуть.

8.  Які атрибути TableLayout впливають на зовнішній вигляд таблиці?

- a) stretchColumns;
- b) onClick;
- c) shrinkColumns;
- d) visibility.

9.  Як задати, що певна колонка має займати доступний простір у TableLayout?

Ваша відповідь: \_\_\_\_\_

10.  Який макет краще підходить для створення чіткої табличної сітки?

- a) LinearLayout;
- b) FrameLayout;

- c) `TableLayout`;
- d) `ConstraintLayout`.

11.  Які атрибути `TableLayout` або `TableRow` можуть впливати на адаптивність таблиці?

- a) `layout_weight`;
- b) `layout_span`;
- c) `textStyle`;
- d) `shrinkColumns`.

12.  Який тег XML використовується для створення рядка в `TableLayout`?

Ваша відповідь: \_\_\_\_\_

13.  Скільки стовпців може містити `TableLayout`?

- a) Не більше 3;
- b) 10 максимум;
- c) Стільки, скільки елементів у `TableRow`;
- d) Визначено в `AndroidManifest`.

14.  Які переваги має `TableLayout`?

- a) Просте відображення табличних даних;
- b) Контроль над розтягуванням стовпців;
- c) Висока прокручуваність;
- d) Легке позиціонування візуальних компонентів по рядках і стовпцях.

15.  Як називається властивість, що дозволяє колонці автоматично зменшуватись при потребі?

Ваша відповідь: \_\_\_\_\_

16.  Що буде, якщо не вказати `stretchColumns` у `TableLayout`?

- a) Колонки будуть однаковими;
- b) Колонки не заповнять весь простір;
- c) Усі елементи вирівнюються по правому краю;
- d) Відбудеться помилка.

17.  Які властивості `TableLayout` можна встановити в XML?

- a) `android:stretchColumns`;
- b) `android:toLeftOf`;
- c) `android:shrinkColumns`;
- d) `android:layout_margin`.

18.  Який елемент найчастіше використовується для тексту всередині таблиці?

Ваша відповідь: \_\_\_\_\_

19.  У чому обмеження `TableLayout`?

- a) Підтримує лише горизонтальну прокрутку;
- b) Не має вбудованої підтримки меж (`border`);
- c) Не працює з `GridLayout`;
- d) Підтримує тільки `TextView`.

20.  Які підходи дозволяють зробити таблицю адаптивною на різних екранах?

- a) dp-одиниці для розмірів;

- b) `stretchColumns/shrinkColumns`;
- c) `layout_gravity=center`;
- d) використання `ScrollView` при потребі прокрутки.

# Тема 7. GridLayout

---

1.  Що таке GridLayout у Android?

- a) Макет для прокрутки списків;
- b) Контейнер для табличного виводу;
- c) Макет, що розміщує елементи в сітці з рядків і стовпців;
- d) Меню з плитками.

2.  Які атрибути впливають на розміщення елементів у GridLayout?

- a) layout\_row;
- b) layout\_column;
- c) layout\_type;
- d) layout\_gravity.

3.  Які два основні параметри описують позицію елемента в GridLayout?

Ваша відповідь: \_\_\_\_\_

4.  Яка перевага GridLayout над TableLayout?

- a) Гнучкіше керування розміщенням і розмірами елементів;
- b) Підтримка scrollView;
- c) Статичне вирівнювання;
- d) Простота стилізації.

5.  Які типи елементів можна вкладати у GridLayout?

- a) TextView;
- b) Button;
- c) ListView;

d) ImageView.

6.  Який атрибут вказує, скільки стовпців займає елемент у GridLayout?

Ваша відповідь: \_\_\_\_\_

7.  Яке значення означає, що елемент займає лише одну клітинку?

a) 0;

b) full;

c) 1;

d) auto.

8.  Які атрибути дозволяють керувати розтягуванням і позиціонуванням елементів у GridLayout?

a) layout\_alignParentLeft;

b) layout\_gravity;

c) layout\_rowSpan;

d) layout\_columnSpan.

9.  Який атрибут дозволяє об'єднати кілька рядків в один елемент?

Ваша відповідь: \_\_\_\_\_

10.  Як задається кількість колонок у GridLayout?

a) layout\_columnsCount;

b) android:columnCount;

c) gridColumns;

d) numColumns.

11.  Які проблеми можуть виникнути при неправильному налаштуванні рядків і стовпців у **GridLayout**?

- a) Перекриття елементів;
- b) Порожні клітинки;
- c) Відсутність стилю;
- d) Вихід за межі екрана.

12.  Яка властивість дозволяє вирівнювати елемент всередині клітинки **GridLayout**?

Ваша відповідь: \_\_\_\_\_

13.  Що буде, якщо вказати занадто велику **rowSpan** для елемента?

- a) Елемент перекриє інші;
- b) Він зникне;
- c) Таблиця поділиться навпіл;
- d) Система оптимізує автоматично.

14.  Які атрибути **GridLayout** відповідають за позицію елемента?

- a) `layout_row`;
- b) `layout_anchor`;
- c) `layout_column`;
- d) `layout_offset`.

15.  Яке значення `layout_gravity` вирівнює елемент по центру?

Ваша відповідь: \_\_\_\_\_

16.  Чим **GridLayout** відрізняється від **TableLayout**?

- a) Працює лише з текстом;
- b) Підтримує контроль над позицією кожного елемента;
- c) Підходить лише для API нижче 14;
- d) Має фіксовану сітку.

17.  Які елементи `layout_gravity` можуть бути корисними у `GridLayout`?

- a) start;
- b) background;
- c) center;
- d) end.

18.  Який тег XML використовується для `GridLayout`?

Ваша відповідь: \_\_\_\_\_

19.  Для яких ситуацій найкраще підходить `GridLayout`?

- a) Створення списків;
- b) Інтерфейс з рівномірно розміщеними плитками/комірками;
- c) Анімація графіки;
- d) Обробка подій.

20.  Які підходи зроблять `GridLayout` адаптивним?

- a) Використання dp-одиниць;
- b) `layout_gravity=center`;
- c) `columnCount` у поєднанні з `wrap_content`;
- d) `layout_alignTop`.

## Тема 8. ScrollView

---

1.  Що таке ScrollView у Android?

- a) Контейнер, який дозволяє прокручування вмісту вертикально;
- b) Віджет списку;
- c) Контейнер, що підтримує анімацію;
- d) Елемент навігації.

2.  Які елементи можна вкласти в ScrollView?

- a) LinearLayout;
- b) FrameLayout;
- c) TextView;
- d) Button.

3.  Який макет найчастіше використовується як дочірній у ScrollView?

Ваша відповідь: \_\_\_\_\_

4.  Яке обмеження має ScrollView?

- a) Може мати лише одного прямого нащадка (дочірній елемент);
- b) Не підтримує прокрутку;
- c) Працює лише з кнопками;
- d) Потребує Adapter.

5.  Які з наступних компонентів можна прокручувати в ScrollView?

- a) ImageView з великою картинкою;
- b) Spinner;
- c) TextView з довгим текстом;

d) Form із багатьма полями.

6.  Який атрибут дозволяє вимкнути прокручування у `ScrollView`?

Ваша відповідь: \_\_\_\_\_

7.  Яка альтернатива `ScrollView` дозволяє горизонтальне прокручування?

a) `FrameScroll`;

b) `HorizontalScrollView`;

c) `ListView`;

d) `SlideLayout`.

8.  Які атрибути `ScrollView` впливають на вигляд прокрутки?

a) `android:scrollbars`;

b) `textSize`;

c) `android:fadingEdge`;

d) `gravity`.

9.  Який тег XML використовується для створення `ScrollView`?

Ваша відповідь: \_\_\_\_\_

10.  Що станеться, якщо в `ScrollView` вставити два дочірніх елементи?

a) Вони відобразяться нормально;

b) Працюватиме лише перший, решта буде проігнорована;

c) Вони автоматично об'єднуються;

d) Прокрутка стане двосторонньою.

11.  У яких випадках доцільно використовувати `ScrollView`?

- a) Довгі форми з полями введення;
- b) Тексти ліцензій або політик конфіденційності;
- c) Меню з 3 пунктами;
- d) Картинка, що не вміщується у вікно.

12.  Який компонент відповідає за горизонтальне прокручування в Android?

Ваша відповідь: \_\_\_\_\_

13.  Який Layout зазвичай використовується всередині ScrollView для організації контенту?

- a) FrameLayout;
- b) RelativeLayout;
- c) LinearLayout;
- d) GridLayout.

14.  Які з наведених атрибутів корисні для оптимізації прокрутки?

- a) android:fillViewport;
- b) maxLines;
- c) android:scrollbars;
- d) android:orientation.

15.  Який атрибут дозволяє вмісту ScrollView розтягуватись до висоти екрану, навіть без прокрутки?

Ваша відповідь: \_\_\_\_\_

16.  Який з варіантів є некоректним використанням ScrollView?

- a) Використання з `LinearLayout`;
- b) Вставка одного `FrameLayout`;
- c) Вставка кількох `View` на одному рівні без обгортки;
- d) Застосування з формою.

17.  Що можна зробити, щоб у `ScrollView` не відображались лінії прокрутки?

- a) Встановити `android:scrollbars="none"`;
- b) `android:scrollbarThumb="transparent"`;
- c) `background="white"`;
- d) Встановити `scrollbars="vertical|horizontal"`, але вимкнути кольори.

18.  Який `Layout` найкраще підходить для `ScrollView` із великою кількістю елементів по вертикалі?

Ваша відповідь: \_\_\_\_\_

19.  Чому не рекомендується використовувати `ScrollView` з `ListView` чи `RecyclerView`?

- a) Немає технічної заборони;
- b) Може викликати конфлікти прокрутки та проблеми з продуктивністю;
- c) `ScrollView` блокує `RecyclerView` повністю;
- d) Вони несумісні.

20.  Які властивості можна встановити, щоб покращити UX прокрутки?

- a) `fillViewport`;
- b) `scrollbars="vertical"`;
- c) `fadingEdgeLength`;

d) `clickable="true"`.

## Тема 9. ListView

---

1.  Що таке **ListView** у **Android**?

- a) Віджет для кнопок;
- b) Контейнер для вертикального списку елементів з прокруткою;
- c) Макет для таблиці;
- d) Контейнер для форм.

2.  Які компоненти потрібні для роботи **ListView**?

- a) **ListView**;
- b) **ViewPager**;
- c) **Adapter**;
- d) Джерело даних (масив, список тощо).

3.  Який тип адаптера найчастіше використовують із простими списками в **ListView**?

Ваша відповідь: \_\_\_\_\_

4.  Яка дія відбувається при натисканні на елемент **ListView** за замовчуванням?

- a) Нічого;
- b) Елемент виділяється синім кольором;
- c) Викликається слухач `onItemClick()`;
- d) Відкривається нова активність.

5.  Які типи адаптерів можна використовувати з **ListView**?

- a) **ArrayAdapter**;

- b) SimpleAdapter;
- c) FrameAdapter;
- d) Custom Adapter (через BaseAdapter).

6.  Який метод дозволяє відстежити натискання на елемент у ListView?

Ваша відповідь: \_\_\_\_\_

7.  Що потрібно зробити, щоб оновити вміст ListView після зміни даних?

- a) Створити нову активність;
- b) Викликати invalidate();
- c) Викликати notifyDataSetChanged() на адаптері;
- d) Очистити кеш.

8.  Які параметри передаються в ArrayAdapter?

- a) Context;
- b) Макет рядка;
- c) Drawable;
- d) Список даних.

9.  Який XML-тег використовується для створення ListView?

Ваша відповідь: \_\_\_\_\_

10.  Що трапиться, якщо адаптер не буде встановлено для ListView?

- a) Виведе пусті рядки;
- b) Нічого не буде відображено;
- c) Виведе назву ListView;
- d) Створить порожній масив.

11.  **Які переваги має використання ListView?**

- a) Підтримка великої кількості елементів;
- b) Прокрутка вбудована за замовчуванням;
- c) Можливість кастомізації рядків;
- d) Потреба в RecyclerView.

12.  **Який метод у адаптері відповідає за повернення кількості елементів?**

Ваша відповідь: \_\_\_\_\_

13.  **Який метод адаптера відповідає за створення/повернення View елемента списку?**

- a) getItemId();
- b) toString();
- c) getView();
- d) onCreate().

14.  **Які елементи можна виводити в ListView?**

- a) TextView;
- b) ImageView;
- c) ButtonGroup;
- d) Комбіновані кастомні макети.

15.  **Як передати список рядків у ListView з використанням ArrayAdapter?**

Ваша відповідь: \_\_\_\_\_

16.  **Яка перевага використання кастомного адаптера з ListView?**
- a) Менше коду;
  - b) Швидше завантаження;
  - c) Можливість виводити складні макети з кількох елементів;
  - d) Більше кольорів.
17.  **Які класи адаптерів реалізують BaseAdapter?**
- a) SimpleAdapter;
  - b) CursorAdapter;
  - c) Custom Adapter;
  - d) ArrayAdapter.
18.  **Яка подія викликається при натисканні на елемент ListView?**
- Ваша відповідь: \_\_\_\_\_
19.  **Який компонент краще використовувати для списків у нових версіях Android?**
- a) ListView;
  - b) ViewPager;
  - c) RecyclerView;
  - d) ScrollView.
20.  **Які обмеження має ListView у порівнянні з RecyclerView?**
- a) Менше гнучкості при анімації;
  - b) Немає вбудованої підтримки ViewHolder;
  - c) Менша продуктивність з великими списками;
  - d) Важко застосувати прокрутку.

# Тема 10. GridView

---

1.  Що таке GridView у Android?

- a) Контейнер для форм;
- b) Контейнер для списку з розгортанням;
- c) Виджет для відображення елементів у вигляді сітки (таблиці);
- d) Меню вкладок.

2.  Які компоненти необхідні для роботи з GridView?

- a) GridView;
- b) Adapter;
- c) Spinner;
- d) Джерело даних (наприклад, масив чи список).

3.  Який тип адаптера найчастіше використовують з GridView?

Ваша відповідь: \_\_\_\_\_

4.  У якому вигляді GridView розміщує елементи?

- a) В один ряд;
- b) У рядках і стовпцях, утворюючи сітку;
- c) У таблиці без прокрутки;
- d) У списку з чекбоксами.

5.  Які події можна обробити для GridView?

- a) onItemClick;
- b) onItemClickLong;
- c) onKeyDown;

d) onScroll.

6.  Який XML-тег використовується для GridView?

Ваша відповідь: \_\_\_\_\_

7.  Що буде, якщо не вказати адаптер для GridView?

- a) Виведеться список з null;
- b) GridView не відобразить жодного елемента;
- c) Буде виняток;
- d) Виведе placeholder.

8.  Які параметри зазвичай задають у GridView?

- a) android:numColumns;
- b) android:textSize;
- c) android:verticalSpacing;
- d) android:horizontalSpacing.

9.  Який атрибут визначає кількість стовпців у GridView?

Ваша відповідь: \_\_\_\_\_

10.  Що таке ViewHolder у контексті GridView?

- a) Об'єкт для зберігання стовпців;
- b) Клас для прокручування;
- c) Патерн оптимізації, який зменшує кількість викликів findViewById();
- d) Додатковий адаптер.

11.  Які елементи інтерфейсу можна відображати в GridView?

- a) Зображення (ImageView);
- b) Spinner;
- c) Текстові елементи (TextView);
- d) Кастомні макети з кількома компонентами.

12.  Який метод адаптера відповідає за побудову кожного елемента у **GridView**?

Ваша відповідь: \_\_\_\_\_

13.  Що відбувається при натисканні на елемент **GridView**?

- a) Нічого;
- b) Викликається `onItemClickListener`, якщо встановлено;
- c) Завжди відкривається нова активність;
- d) Елемент зникає.

14.  Які переваги має **GridView**?

- a) Відображення великої кількості елементів у сітці;
- b) Підтримка `swipe-to-refresh`;
- c) Гнучка кастомізація макетів;
- d) Просте використання з адаптерами.

15.  Який клас відповідає за подію натискання на елемент **GridView**?

Ваша відповідь: \_\_\_\_\_

16.  Яка альтернатива **GridView** використовується в нових проєктах?

- a) `ScrollView`;
- b) `ListView`;

- c) RecyclerView з GridLayoutManager;
- d) WebView.

17.  **Які атрибути впливають на зовнішній вигляд сітки GridView?**

- a) android:verticalSpacing;
- b) android:inputType;
- c) android:horizontalSpacing;
- d) android:stretchMode.

18.  **Як можна задати відступи між елементами у GridView?**

Ваша відповідь: \_\_\_\_\_

19.  **Що таке stretchMode у GridView?**

- a) Вирівнювання тексту;
- b) Спосіб розтягування стовпців для заповнення ширини;
- c) Тип анімації;
- d) Режим масштабування зображень.

20.  **Що потрібно зробити, щоб обробити натискання на елемент GridView?**

- a) Встановити onItemClickListener;
- b) Використати GestureDetector;
- c) Додати setOnClickListener всередині адаптера;
- d) Реалізувати TouchDelegate.

# Тема 11. RecyclerView

---

1.  Що таке RecyclerView у Android?

- a) Меню для вибору вкладок;
- b) Контейнер для форм;
- c) Гнучкий та оптимізований контейнер для списків і сіток із прокруткою;
- d) Віджет для діалогів.

2.  Які компоненти обов'язково потрібні для роботи RecyclerView?

- a) RecyclerView;
- b) ListAdapter;
- c) LayoutManager;
- d) Adapter.

3.  Який LayoutManager використовується для вертикального списку в RecyclerView?

Ваша відповідь: \_\_\_\_\_

4.  Який метод адаптера відповідає за створення нових ViewHolder?

- a) bindView();
- b) onCreateViewHolder();
- c) onBindViewHolder();
- d) createItem().

5.  Які методи повинен реалізовувати RecyclerView.Adapter?

- a) onCreateViewHolder();
- b) onClick();

- c) onBindViewHolder();
- d) getItemCount().

6.  **Який клас у RecyclerView використовується для кешування посилань на View?**

Ваша відповідь: \_\_\_\_\_

7.  **Що таке ViewHolder у контексті RecyclerView?**

- a) Об'єкт для прокрутки;
- b) Клас для зберігання View кожного елемента, щоб уникати повторного виклику findViewById();
- c) Елемент інтерфейсу;
- d) Тип списку.

8.  **Які LayoutManager-и доступні для RecyclerView?**

- a) LinearLayoutManager;
- b) GridLayout;
- c) GridLayoutManager;
- d) StaggeredGridLayoutManager.

9.  **Який метод викликається для прив'язки даних до елемента в RecyclerView?**

Ваша відповідь: \_\_\_\_\_

10.  **Яка ключова перевага RecyclerView над ListView?**

- a) Простота у використанні;
- b) Краще вирівнювання тексту;

- c) Підтримка ViewHolder і layoutManager;
- d) Підтримка тільки Grid.

11.  **Які події можна обробляти в RecyclerView?**

- a) onClick;
- b) onLongClick;
- c) onOpen;
- d) onScrollListener.

12.  **Який метод повертає кількість елементів у RecyclerView.Adapter?**

Ваша відповідь: \_\_\_\_\_

13.  **Який компонент відповідає за визначення типу розташування елементів у RecyclerView?**

- a) Adapter;
- b) ScrollView;
- c) layoutManager;
- d) View.

14.  **Які переваги має використання ViewHolder?**

- a) Покращення продуктивності;
- b) Менше викликів findViewById();
- c) Легша інтеграція зі Spinner;
- d) Повторне використання елементів View.

15.  **Який layoutManager дозволяє створити сітку в RecyclerView?**

Ваша відповідь: \_\_\_\_\_

16.  Що потрібно зробити для додавання розділювачів між елементами RecyclerView?

- a) Додати margin;
- b) Використати ScrollView;
- c) Додати ItemDecoration;
- d) Встановити background.

17.  Які способи прокручування підтримує RecyclerView?

- a) Вертикальна прокрутка;
- b) android:autoScroll;
- c) Горизонтальна прокрутка;
- d) smoothScrollToPosition().

18.  Який клас відповідає за анімації додавання, видалення елементів у RecyclerView?

Ваша відповідь: \_\_\_\_\_

19.  Що таке notifyDataSetChanged()?

- a) Видаляє всі елементи;
- b) Прокручує список вниз;
- c) Оновлює список після зміни даних;
- d) Додає новий адаптер.

20.  Які кроки потрібно виконати для створення RecyclerView?

- a) Створити макет елемента (item layout);
- b) Створити ViewHolder;

- c) Реалізувати Adapter;
- d) Встановити TextWatcher.

## СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Griffiths D. Head First Kotlin: A Brain-Friendly Guide / Griffiths Dawn, David Griffiths / - O'Reilly, 2019. – 450 p.
2. Khazada J. Smartphones: Each & Everything You Want to Know about Your Smartphone (How To Buy Best # 1) / J. Khazada. – CreateSpace Independent Publishing Platform; Large Print edition, 2016. – 200 p.
3. Makan K. Android Security Cookbook. – Packt Publishing Limited, 2013. – 350 p.
4. Phillips B. Android Programming: The Big Nerd Ranch Guide. 4th edition / Phillips B., Stewart C., Marsicano K. / – Big Nerd Ranch Guides, 2019. – 624 p.
5. Roger Y. Android System Programming. – Packt Publishing Ltd, 2017. – 470 p.
6. Smyth N. Android Studio 4.1 Development Essentials: Java Edition. – Payload Media, 2020. – 810 p.
7. Vaananen J. The Smart Device / J. Vaananen, J. Gosling, B. Balsic. – CreateSpace Independent Publishing Platform, 2016. – 206 p.

Навчально-методичне видання

ЛЯШУК Тарас

# **ПРОГРАМУВАННЯ МОБІЛЬНИХ ПРИСТРОЇВ.**

**ЗБІРНИК ТЕСТОВИХ ЗАВДАНЬ НА ТЕМУ**

**"КОНТЕЙНЕРИ ТА МАКЕТИ"**

Практикум