

ОСОБЛИВОСТІ СТВОРЕННЯ ВЛАСНИХ УЯВЛЕНЬ В РОЗРОБЦІ МОБІЛЬНИХ ЗАСТОСУНКІВ

Афанасьєва А.М., Гунько М.А.

Науковий керівник – к.т.н., ст. викл. Ткачов В.М.

Харківський національний університет радіоелектроніки
(61166, Харків, просп. Науки, 14, каф. ЕОМ, тел. (057) 702-13-54)

e-mail: [d ec@nure.ua](mailto:dec@nure.ua)

Developers are constantly designing different kinds of user interfaces using XML, but in addition to that, it is easy to learn how to create your own views, which open up new benefits and avoid boilerplate code reuse. Android has a wide range of pre-built widgets and layouts for creating a user interface, but they cannot meet all the requirements of our applications. In addition, here the ability to create your own views comes to the rescue. By creating your own view subclass, you can get the most complete control over the appearance and functionality of a display element.

Для створення власного уявлення необхідно розробити клас, який буде розширювати один з наявних класів, які в свою чергу будуть розширені з класів View або ViewGroup. Далі необхідно додати конструктори до нашого класу. Їх буде декілька. Найпростіший – для створення уявлення з коду програми – приймає тільки один параметр(контекст). Контекст – об'єкт через який можна отримати доступ до поточної теми, ресурсів і т. д.. Є ще 2 конструктори, які приймають ще атрибути та стилі відповідно. Перший, з вище зазначених - конструктор, який викликається при формуванні уявлення з XML-файлу. Він викликається, коли уявлення створюється з XML-файлу, що містить атрибути уявлення. У цьому варіанті конструктора використовується стиль за замовчуванням (0), тому застосовуються тільки ті значення атрибутів, які є в темі контексту і заданому наборі AttributeSet. Також конструктор може бути визначеним за допомогою анотації @JvmOverloads.

У цього класу обов'язково повинен бути метод OnDraw (canvas: Canvas?). Він приймає параметр Canvas (полотно), за допомогою якого компонент уявлення може малювати себе. Для малювання на полотні необхідно створити об'єкт Paint. Як правило, процес малювання визначається двома аспектами:

- що малювати (визначається об'єктом Canvas);
- як малювати (визначається об'єктом Paint).

Наприклад, Canvas надає метод для малювання лінії, а Paint надає методи для визначення кольору цієї лінії. Наприклад, об'єкт Canvas в класі надає метод для малювання кола, а об'єкт Paint заповнює її кольором. Простіше кажучи, Canvas визначає, які фігури можна намалювати на екрані, а Paint визначає властивості намальованих фігур - колір, стиль, шрифт і т. д.

При виконанні відтворення завжди треба пам'ятати про те, що слід повторно використовувати об'єкти замість створення нових. IDE може вказати на потенційні проблеми, але покладатися на неї не варто. Наприклад, вона не зможе відстежити випадок, коли об'єкти створюються всередині методів, що викликаються з методу `onDraw ()`. Тому краще перевіряти всі самостійно.

Отже, ми задали власне уявлення. Якщо ми хочемо оновлювати уявлення при зміні якого-небудь властивості або з якоїсь іншої причини, цього можна досягти двома основними способами:

`invalidate ()` - це метод, який ініціює примусову перемальовування певного уявлення. Простіше кажучи, метод `invalidate ()` слід викликати в разі, коли потрібна зміна зовнішнього вигляду уявлення.

Якщо в якийсь момент відбувається зміна стану уявлення, то метод `requestLayout ()` повідомляє системі уявлень, що необхідно зробити перерахунок фаз «вимір» (`Measure`) і «макет» (`Layout`) для даного подання (вимір → макет → малювання). Простіше кажучи, метод `requestLayout ()` слід викликати в разі, коли потрібна зміна меж уявлення. Також можливо встановити інтервал, через який буду спрацьовувати метод `invalidate ()`. Щоб це зробити необхідно викликати метод `invalidateDelay (val ms: Long)` в методі `OnDraw ()`. Метод `invalidateDelay` приймає один параметр – час, через який буде спрацьовувати метод `invalidate ()` та передається в мілісекундах.

Список використаної літератури:

1. Vitalii Tkachov, Anna Budko, Kateryna Hvozdetzka and Daryna Hrebenuk. Method of Building Dynamic Multi-hop VPN Chains for Ensuring Security of Terminal Access Systems // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 06-09 oct. 2020, Kharkiv.

2. Tkachov, V., Bondarenko, M., Ulyanov, O., & Reznichenko, O. (2019, December). Overlay Network Infrastructure for Remote Control of Radio Astronomy Observatory. In 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT) (pp. 161-165).

3. Tkachov, V., Hunko, M., Volotka, V.: Scenarios for Implementation of Nested Virtualization Technology in Task of Improving Cloud Firewall Fault Tolerance. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), pp. 759-763. IEEE (2019).

4. Hunko M.A., Tkachov V.M. Development of a module for sorting the ipaddresses of user nodes in cloud firewall protection of web resources. Дев'ята міжнародна науково-технічна конференція «Сучасні напрями розвитку інформаційнокомунікаційних технологій та засобів управління». 2019. С. 30.