



Galeria na androida

Celem
projektu jest:

Zaprezentowanie różnicy wydajności ładowania całych plików do widoków w aplikacji, a po wcześniejszym ustaleniu jak duża ich część jest nam rzeczywiście potrzebna vs "Glide"[1].

Pokazanie przykładowego kodu mającego na celu zapisanie ostatnich danych użytkownika ("SharedPreferences"[2]).

Wykonanie projektu

Projekt był wykonany w programie Android Studio

Program ten korzysta z Gradle [3], który jest potrzebny do wygenerowania pliku .apk, który możemy przenieść na urządzenie, aby korzystać z napisanej aplikacji

Glide

- Glide jest powiedziałbym bardzo wydajną biblioteką stworzoną do ładowania obrazów i filmów, która dodatkowo daje możliwość wyświetlania gifów, zamiast statycznego obrazu
- Samą bibliotekę implementujemy do Android bardzo szybko poprzez dopisanie do pliku 'build.gradle'
- Przedstawiony kod dodajemy do pliku 'build.gradle' w folderze 'apk' aplikacji

```
dependencies {  
  
    implementation 'com.github.bumptech.glide:glide:4.12.0'  
    annotationProcessor 'com.github.bumptech.glide:compiler:4.12.0'  
}
```

Zadanie 1

cz.1

- Aby umożliwić w aplikacji sortowanie widoku folderów po ich nazwie, w pliku 'Directory.java' proszę o dodanie dwóch komparatorów (po nazwie i rozmiarze – tutaj jako ilość plików w folderze), w tym celu:
 1. W konstruktorze klasy Directory przypisz do zmiennej "name" [typ: String] nazwę folderu na podstawie jego ścieżki (np. Korzystając z metody 'substring')
 2. Do klasy Directory dodaj dwa komparatory porównujące zmienne "name" i "size"

Zadanie 1 cz.2

- 3. Zwróć uwagę na to, że zarówno nazwy jak i rozmiary mogą się powtarzać, w tym celu wewnątrz 'ogólnych' komparatorów dodaj dodatkowe warunki, pozwalające jak najdalej rozstrzygnąć 'pierwszeństwo' przy sortowaniu.
- Przykład implementacji:
- <https://www.baeldung.com/java-comparator-comparable#1-creating-comparators>

Zadanie 2

cz.1

W celu zapisania najprostszych typów danych zastosowałem API "SharedPreferences", którego celem jest przechowanie najprostszych informacji tak, aby przy kolejnym otwarciu aplikacji nie trzeba było ustawiać wszystkiego od początku.

1. Uzupełnij plik 'starting_data_values.xml' przykładowymi podstawowymi danymi startowymi aplikacji.

Zad2 cz.2

- 2. W pliku SettingsScreen.java w metodzie saveData() zapisz do pliku (zmienna "dataFilename") każdą daną zainicjowaną wcześniej w pliku starting_data_values.xml
- Krótki przykład implementacji:
- <https://developer.android.com/training/data-storage/shared-preferences>

Bibliografia (teoria):

- [1] <https://bumptech.github.io/glide/>
- [2] <https://developer.android.com/reference/android/content/SharedPreferences>
- [3] <https://www.geeksforgeeks.org/android-build-gradle/>

Linki do stron, z których korzystałem podczas nauki

- <https://www.baeldung.com/java-comparator-comparable>
- <https://www.geeksforgeeks.org/gridview-in-android-with-example/>
- <https://developer.android.com/guide/topics/ui/layout/recyclerview>
- <https://www.youtube.com/watch?v=AkiltTv0CjA>
- <https://developer.android.com/training/system-ui/navigation>
- <https://www.youtube.com/watch?v=xHCsL5QOSv4&t>