

Zadanie – walidacja danych

10:45

Formularz Rejestracyjny

Imię i nazwisko

E-mail

E-mail musi zawierać znak '@'

Nr telefonu

Podano niepoprawny numer telefonu

Hasło

Powtórz hasło

Hasła nie pasują do siebie

Wyślij

Witaj "imię i nazwisko". Twoje dane: "e-mail", "nr telefonu", "hasło".

Twoim zadaniem jest utworzenie formularza rejestracji, którego układ przedstawiono powyżej. Formularz będzie zawierać 5 pól tekstowych (każde pole tekstowe odpowiedniego typu do danych, które będzie przechowywać), 5 obiektów `TextView` oraz 1 przycisk „Wyślij”.

1. **Struktura głównego kontenera** - Zastosowano układ liniowy wertykalny, elementy w tym układzie powinny być wyrównane do środka w pionie i poziomie. Układ liniowy posiada marginesy 12 jednostek.
2. **Obiekt `TextView` „Formularz rejestracyjny”** - rozmiar czcionki 24 jednostki odpowiadające tekstom, styl tekstu pogrubiony.
3. **Pola edycji tekstu „Imię i nazwisko” oraz „E-mail”** – szerokość elementów powinna być dopasowana do rodzica, pole przechowujące imię i nazwisko powinno być typu tekstowego, a pole z adresem e-mail powinno być typu `E-mail`.

4. **Obiekt TextView z walidacją adresu e-mail** – będzie wyświetlał tekst w kolorze czerwonym, który informuje o niepoprawnym adresie e-mail.
5. **Pole edycji tekstu „Nr telefonu”** – szerokość elementu powinna być dopasowana do rodzica, pole powinno być typu „phone” oraz powinno zawierać margines górny 16 jednostek.
6. **Obiekt TextView z walidacją numeru telefonu** - będzie wyświetlał tekst w kolorze czerwonym, który informuje o niepoprawnym numerze telefonu.
7. **Pola edycji tekstu „Hasło” i „Powtórz hasło”** - szerokość elementu powinna być dopasowana do rodzica, pole powinno być typu „password”. Pierwsze pole edycji powinno zawierać margines górny 16 jednostek.
8. **Obiekt TextView z walidacją numeru telefonu** - będzie wyświetlał tekst w kolorze czerwonym, który informuje o niepasujących do siebie hasłach.
9. **Wszystkie pola edycyjne powinny zawierać podpowiedź pokazującą się przed wprowadzeniem tekstu.**
10. **Przycisk „Wyślij”** – domyślny styl, tekst wewnętrzny przycisku „Wyślij”. Przycisk powinien mieć ustawioną szerokość do zawartości.
11. **Obiekt TextView z danymi rejestracji** – rozmiar czcionki 16 jednostek, szerokość dopasowana do rodzica, wyrównanie tekstu do lewej.

Logika walidacji formularza

Domyślnie żaden obiekt `TextView` nie powinien zawierać tekstu wewnątrz. Po naciśnięciu przycisku „Wyślij”, który ma przypięte zdarzenie `onClickListener`, wywoływane są po kolei 3 metody walidujące:

- I. Walidacja adresu e-mail – należy sprawdzić, z wykorzystaniem metody z klasy `String`, czy podany tekst zawiera znak '@'. Jeżeli go nie zawiera to w odpowiednim obiekcie `TextView` wyświetlana jest ta informacja. Jeżeli zawiera to walidacja przechodzi poprawnie (trzeba zapisać wynik walidacji).
- II. Walidacja numeru telefonu – należy sprawdzić, czy numer telefonu jest liczbą oraz czy długość jest równa 9. Do tej metody walidacyjnej należy utworzyć poboczną metodę, która będzie zwracać prawdę lub fałsz zależnie od tego, czy podany ciąg znaków jest liczbą. Do tej metody należy użyć metody z klasy `Integer`, która próbuje parsować tekst na liczbę. Parsowanie powinno być owinięte w blok `try . . . catch`, który po poprawnej walidacji zwraca prawdę, a po wyłapaniu wyjątku `NumberFormatException` zwraca fałsz. Jeżeli numer telefonu nie spełnia warunków walidacji to w odpowiednim obiekcie `TextView` jest wyświetlana ta informacja. Jeżeli walidacja przebiegła poprawnie to należy zapisać jej wynik.
- III. Walidacja hasła – należy sprawdzić, czy z pola edycyjne z hasłami zawierają ten sam ciąg znaków. Nie należy używać operatora porównania (`==`) tylko metody z klasy `String`. Jeżeli walidacja przebiegła niepoprawnie to należy wyświetlić

odpowiedni komunikat w odpowiednim obiekcie `TextView`. Jeżeli przebiegła poprawnie to należy zapisać jej wynik.

Dopiero gdy wszystkie 3 testy walidacji przebiegną poprawnie to pod przyciskiem „Wyślij” wyświetlają się dane z formularza w formacie podanym na zrzucie ekranu.

Ważne informacje

Wszystkie nazwy metod, zmiennych oraz identyfikatorów elementów UI powinny być znaczące oraz powinny zawierać inicjały z Twojego imienia i nazwiska. Na przykład ID dla przycisku wysyłania: `sendButton_sj`, zmienna przechowująca tekst z pola email: `mailText_sj`.

Każda metoda walidacyjna powinna być osobną metodą, przypisywanie elementów UI do zmiennych powinno odbyć się wewnątrz metody `onCreate`.

Aplikację należy utworzyć na podstawie nowego, pustego projektu „Empty Views Activity”.

Co podlega ocenie

- Wygląd aplikacji zbliżony do tego, co na zrzucie ekranu.
- Prawidłowo zaimplementowany układ liniowy.
- Odpowiednie rozmiary oraz kolory tekstu.
- Odpowiednio ustawione marginesy.
- Prawidłowe nazewnictwo metod, zmiennych oraz ID.
- Prawidłowy podział na metody w kodzie.
- Zastosowanie metod z klas `String` i `Integer`, opisanych powyżej.
- Czytelność kodu.
- Działanie aplikacji

Ocenianie będzie zbliżone do oceniania podczas egzaminu zawodowego – wszystkie kryteria w jednym punkcie muszą być spełnione aby uzyskać punkty za tą część.

Za wykonanie wszystkiego prawidłowo można uzyskać maksymalnie ocenę **bardzo dobrą**.

Na ocenę celującą należy zmienić sposób walidacji hasła na taki, że hasło jest sprawdzane przy każdym naciśnięciu klawisza. Do tego należy użyć zdarzenia `TextChangedListener` wraz z obiektem `TextWatcher`, który nasłuchuje zmian w polu edycyjnym.