

Część I. Aplikacja internetowa

Zadanie: aplikacja webowa (formularz)

W ramach zadania należy zaimplementować aplikację webową, która będzie umożliwiała dodawanie nowych kart z danymi osobowymi. Formularz dodawania składa się z 4 pól (imię, nazwisko, numer telefonu, płeć), które będą wyświetlane w formie listy (w kolumnie). Należy dodać przycisk, który wyczyści formularz, a dane zostaną zapamiętane (jeśli wszystkie dane są prawidłowe). Kolejny przycisk, który wyczyści cały formularz. Obok formularza ma znajdować się kolejny fragment strony, który zawiera dropdown list. Dzięki niemu, można sprawdzić ile danych zostało zapamiętanych. Dropdown list wyświetla opcje (imię + nazwisko), po wybraniu osoby z listy, wyświetlą się dane w postaci ramki, gdzie pojawią się dane (mini tabelka lub labelki).

Aplikację należy zaimplementować w jednym z dostępnych frameworków ReactJS lub Angular.

Do formularza należy dodać walidację, która będzie sprawdzać poprawność danych:

- Imię i nazwisko -> pierwsza litera duża, reszta same małe
- nr telefonu -> składa się z 9 cyfr (tylko cyfry)
- płeć (jest to dropdown lista: kobieta, mężczyzna, inna)

Wygląd elementów (wygląd strony ma być zbliżony do obrazków referencyjnych):

Formularz

Imię:

Nazwisko:

Telefon:

Płeć:

Wyczyść

Dodaj

Database

Imię: Imię

Nazwisko: Nazwisko

Telefon: 111222333

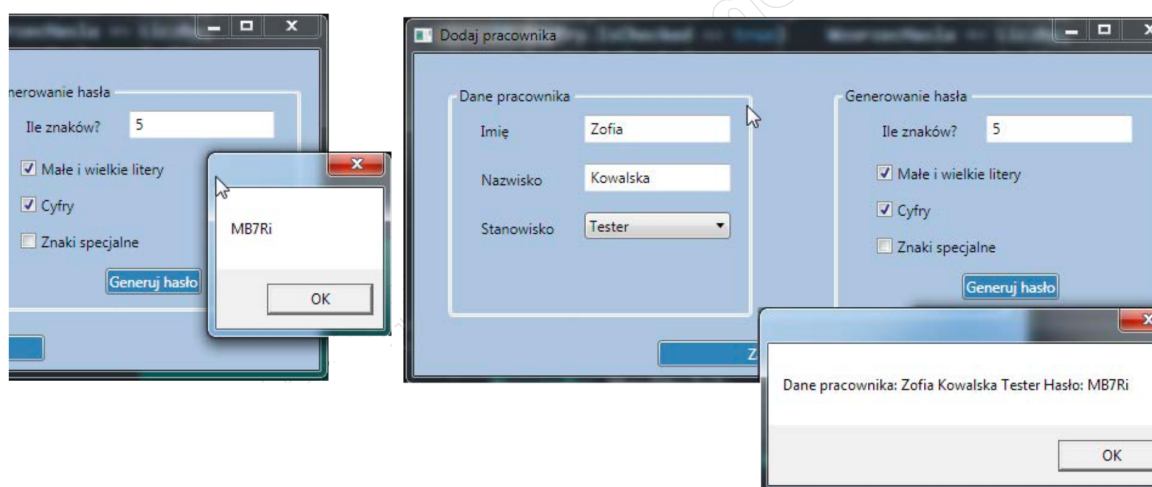
Płeć: Kobieta

Część II. Aplikacja okienkowa

Za pomocą dostępnego na stanowisku egzaminacyjnym środowiska programistycznego wykonaj aplikację desktopową do wprowadzania danych pracownika z możliwością generowania hasła.



Obraz 1. Stan początkowy aplikacji



Obraz 2. Fragment okna po wybraniu przycisku „Generuj hasło”

Obraz 3. Po wybraniu przycisku „Zatwierdź”

Na obrazie 1 przedstawiono ideę aplikacji desktopowej. W zależności od użytego środowiska programistycznego wygląd może nieznacznie się różnić.

Opis wyglądu aplikacji

- Okno o nazwie „Dodaj pracownika”.
- Kontrolki rozmieszczone zgodnie z obrazem 1
- Kontrolki w grupie „Dane Pracownika”:
 - pole edycyjne poprzedzone etykietą o treści „Imię”

- pole edycyjne poprzedzone etykietą o treści „Nazwisko”
- lista rozwijalna podpisana „Stanowisko” z elementami: Kierownik, Starszy programista, Młodszy programista, Tester
- Kontrolki w grupie „Generowanie hasła”:
 - pole edycyjne poprzedzone etykietą o treści „Ile znaków?”
 - trzy pola wyboru podpisane: „Małe i wielkie litery”, „Cyfry”, „Znaki specjalne”.
Pierwsze pole jest domyślnie zaznaczone
 - przycisk o treści „Generuj hasło”
- Na dole okna znajduje się przycisk „Zatwierdź”, jest on wyraźnie dłuższy niż przycisk do generowania hasła
- Okno ma tło koloru LightSteelBlue (#B0C4DE)
- Przyciski mają tło koloru SteelBlue (#4682B4) i biały kolor czcionki

Działanie aplikacji po wybraniu przycisku „Generuj hasło”:

- Generowane jest hasło o liczbie znaków określonej w polu edycyjnym
- Poszczególne znaki hasła są wybierane losowo z zestawu małych liter
- Jeżeli zaznaczono checkbox „Małe i wielkie litery”, jeden ze znaków jest losowany z zestawu wielkich liter
- Jeżeli zaznaczono checkbox „Cyfry”, jeden ze znaków hasła jest losowany z zestawu cyfr
- Jeżeli zaznaczono checkbox „Znaki specjalne”, jeden ze znaków hasła jest losowany z zestawu znaków specjalnych
- Dla uproszczenia zadania można przyjąć zawsze określony znak (np. pierwszy), który jest losowany z cyfr. Podobnie należy postąpić ze znakami specjalnymi (np. drugi znak)
- Hasło jest wypisywane jako komunikat zgodnie z obrazem 2 po wybraniu przycisku „Generuj hasło”
- Za zestaw liter przyjmuje się wszystkie litery z klawiatury małe i wielkie alfabetu łacińskiego
- Za zestaw cyfr przyjmuje się kolejne cyfry od 0 do 9
- Za zestaw znaków specjalnych przyjmuje się znaki !@#\$%^&*()_+ -=
- Hasło oraz zestawy znaków są przechowywane w zmiennych typu napisowego

Działanie aplikacji po wybraniu przycisku „Zatwierdź”:

- Jeśli hasło nie zostało wcześniej wygenerowane (nie wciśnięto przycisku „Generuj hasło”) to generowane jest nowe hasło zgodnie z wybranymi kryteriami.
- Wyświetlany jest komunikat z wypełnionymi danymi pracownika oraz wygenerowanym wcześniej hasłem, zgodnie z obrazem 3.

UWAGA: dla uproszczenia aplikacji nie trzeba sprawdzać czy pole edycyjne jest wypełnione i czy liczba znaków w hasle jest wystarczająca dla wstawienia małych, wielkich liter, liczb i znaków specjalnych

Aplikacja powinna być zapisana czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podjmij próbę kompilacji i uruchomienia aplikacji. Kod aplikacji przygotuj do nagrania. W podfolderze *okienkowa* powinno znaleźć się archiwum całego projektu o nazwie *okienkowa.zip*

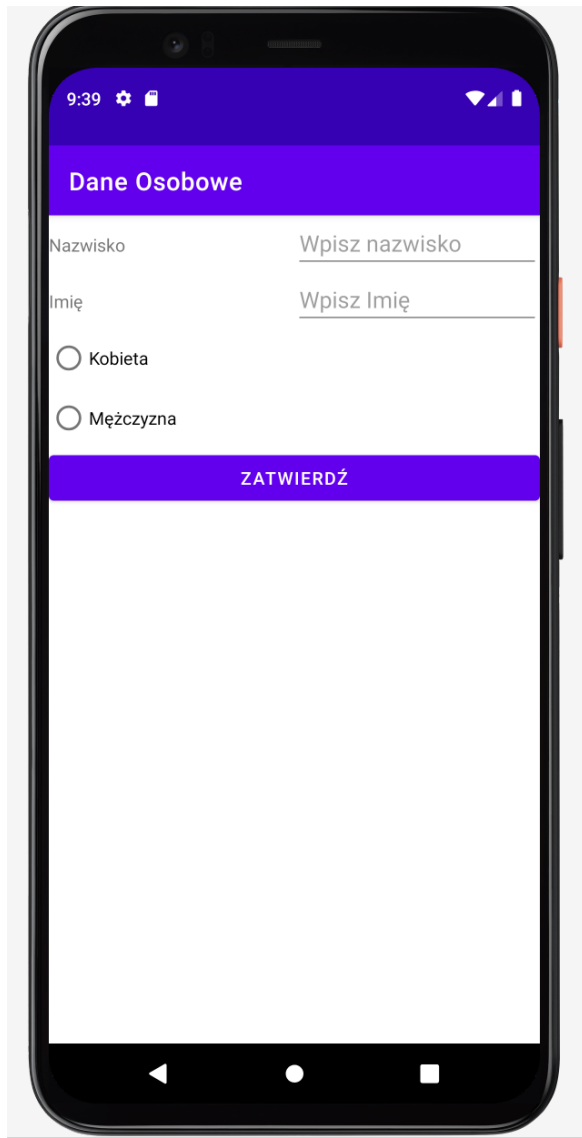
Przydatne rzeczy:

```
Random rnd = new Random();  
rnd.Next(1, 11)); //Liczby losowe od 1 do 10
```

```
Random rnd = new Random();  
string[] malePetNames = { "Rufus", "Bear", "Dakota", "Fido",  
                           "Vanya", "Samuel", "Koani", "Volodya",  
                           "Prince", "Yiska" };  
// Generate random indexes for pet names.  
int mIndex = rnd.Next(malePetNames.Length);
```

Część III. Aplikacja mobilna

Wykonaj aplikację mobilną za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Aplikacja umożliwia wpisywanie prostych notatek.



Na obrazie przedstawiono stan po uruchomieniu aplikacji mobilnej. W zależności od zastosowanego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie się różnić od przedstawionego.

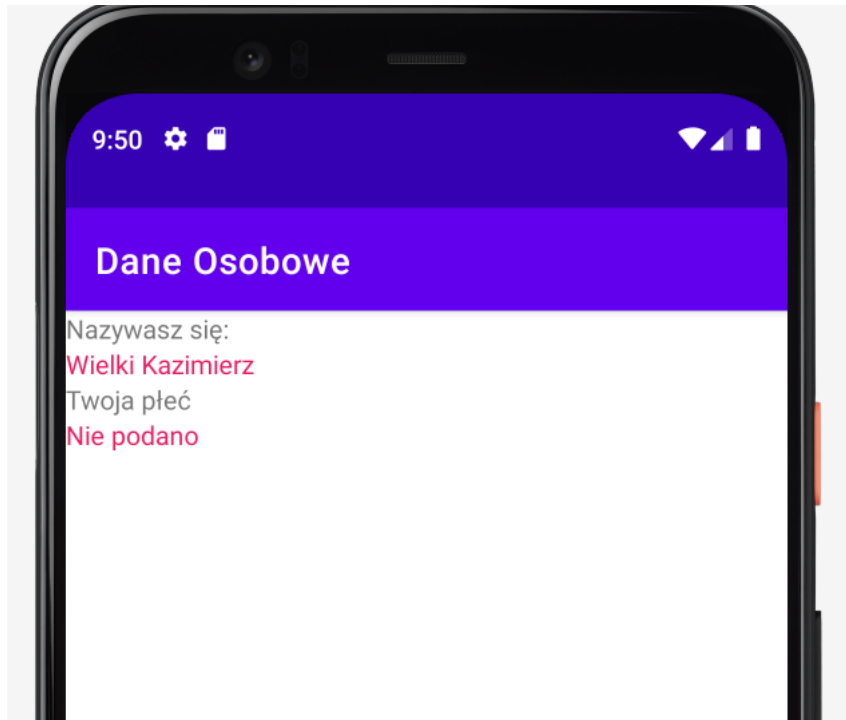
Elementy Activity:

- Pole tekstowe: Nazwisko
- Pole edycyjne z podpowiedzą: „Wpisz Nazwisko”
- Pole tekstowe: Imię
- Pole edycyjne z podpowiedzą: „Wpisz Imię”
- Zgrupowane dwa RadioButton'y z wartościami: Kobieta, Mężczyzna
- Przycisk o treści: „Zatwierdź”
- Rozmieszczenie elementów takie jak na powyższym obrazku

Działanie aplikacji:

- Po wciśnięciu przycisku „Zatwierdź” program przekazuje wprowadzone dane (imię, nazwisko oraz informację o płci) do drugiego Activity.

Drugie activity ma wygląd jak na poniższym obrazku:

**Elementy Activity:**

- Pole tekstowe: Nazywasz się
- Pole tekstowe z otrzymanym od pierwszego Activity imieniem i nazwiskiem
- Pole tekstowe: Twoja płeć
- Pole tekstowe z wypisanym jednym z trzech komunikatów („Kobieta”, „Mężczyzna”, „Nie podano”. Komunikat „Nie podano” jest wypisywany gdy użytkownik w pierwszym Activity nie wybrał żadnej płci (nie zaznaczył żadnego RadioButton’a).

Założenia aplikacji:

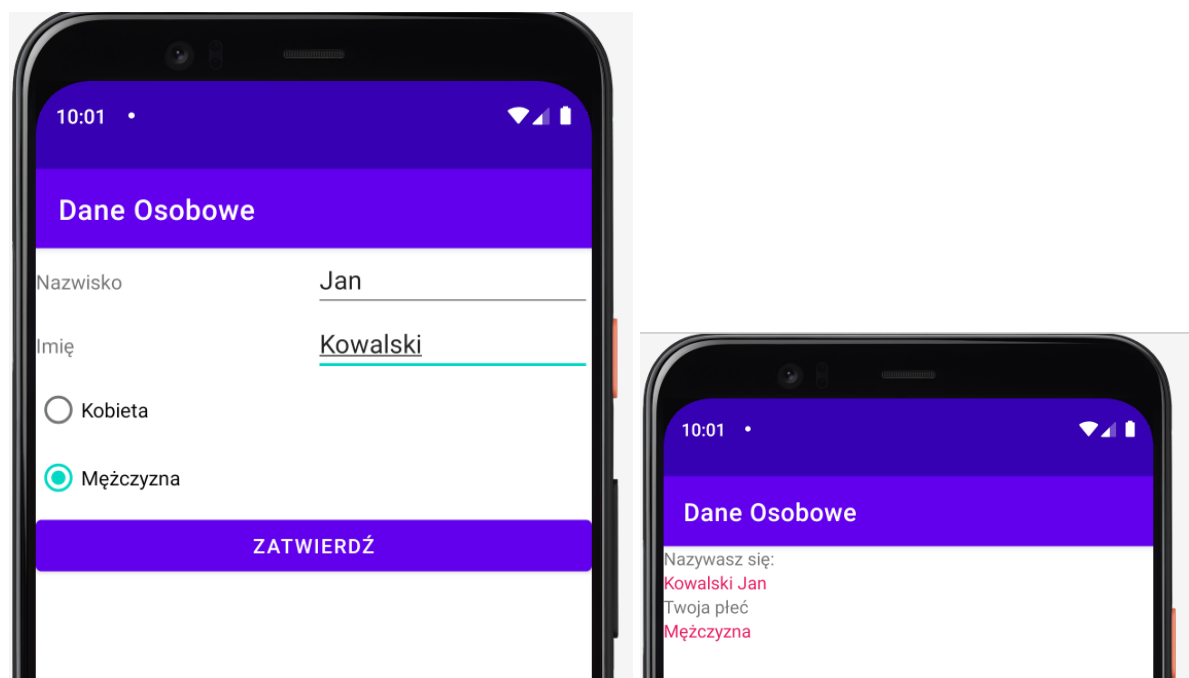
- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML).
- Zastosowany typ rozkładu liniowy wertykalny (Linear / Stack lub inny o tej idei) z zagłębionym rozkładem liniowym horyzontalnym dla pola tekstowego i edycyjnego
- Color czcionki dla wypisywanego w drugim Activity imienia i nazwiska oraz informacji o płci #E91E63
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

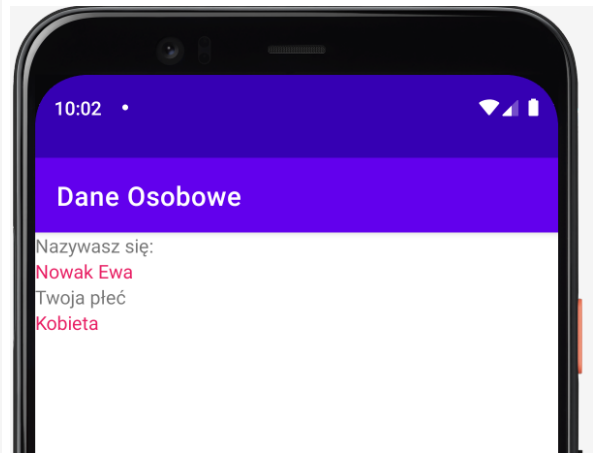
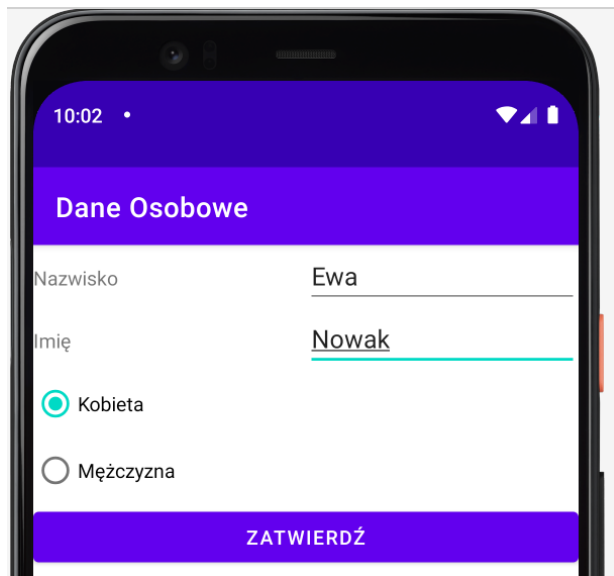
Podjmij próbę kompilacji i emulacji. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części IV zadania. Kod aplikacji przygotuj do nagrania. W podfolderze *mobilna* powinno znaleźć się archiwum całego projektu o nazwie *mobilna.zip*

Przydatne rzeczy dla AndroidStudio:

```
radioButton.isChecked()  
Intent intent = new Intent( . . . )  
intent.putExtra( . . . )  
startActivity(intent);  
getIntent().getIntExtra( . . . )  
getIntent().getStringExtra( . . . )
```

Przykładowe działanie aplikacji:





Część IV. Dokumentacja aplikacji

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej utwórz nagłówek funkcji przeszukującej, według wzoru zgodnie z listingiem 1. Nagłówek powinien znaleźć się w kodzie źródłowym nad funkcją. W miejscu znaków <> należy podać nazwę funkcji, nazwy argumentów wraz z krótkim opisem, typ zwracany wraz z krótkim opisem, krótki opis zawierający przynajmniej nazwę algorytmu. Gdy funkcja nie ma argumentów – zapisać „brak”. W miejscu autor należy podać swoje nazwisko i imię.

UWAGA: Dokumentację umieścić w komentarzu (wieloliniowym lub kilku jednoliniowych). Znajdujący się w listingu 1 wzór dokumentacji jest bez znaków początku i końca komentarza, gdyż te są różne dla różnych języków programowania

Listing 1. Wzór dokumentacji funkcji

```
*****
nazwa funkcji: <tu wstaw nazwę funkcji>
argumenty: <nazwa argumentu> - <co przechowuje>
           < nazwa argumentu> - <co przechowuje>
typ zwracany: <nazwa typu>, <co jest zwracane>
informacje: <opis>
autor: <Nazwisko i imię zdającego>
*****
```

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji utworzonych podczas egzaminu. Zrzuty powinny obejmować cały obszar ekranu monitora z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na rzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem lub okno terminala z kompilacją projektu. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na rzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wykonać należy tyle zrzutów ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja internetowa – dowolna liczba zrzutów nazwanych *internetowa1*, *internetowa2* ...
- Aplikacja okienkowa – dowolna liczba zrzutów nazwanych *window1*, *window2* ...
- Aplikacja mobilna – dowolna liczba zrzutów nazwanych *mobile1*, *mobile2* ...

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać podpisane zrzuty ekranu oraz zapisane informacje:

- o Nazwę systemu operacyjnego, na którym pracował zdający.
- o Nazwy środowisk programistycznych, z których zdający korzystał na egzaminie.
- o Nazwy języków programowania.
- o Nazwa emulatora systemu mobilnego.

Zrzuty ekranu i dokument umieścić w podfolderze *dokumentacja*.

Nagraj folder z rezultatami pracy. W folderze z Nazwiskiem i imieniem zdającego powinny się znajdować podfoldery: internetowa, okienkowa, mobilna, dokumentacja.

W folderze dokumentacja: pliki ze zrzutami oraz plik egzamin.

W folderze internetowa: spakowany cały projekt aplikacji internetowej, część serwerowa oraz klientowa

W folderze okienkowa: spakowany cały projekt aplikacji okienkowej, pliki z kodem źródłowym, opcjonalnie plik uruchomieniowy

W folderze mobilna: spakowany cały projekt aplikacji mobilnej, pliki ze źródłami interfejsu i logiki.

Upewnij się, że wszystko co stworzyłeś zostało prawidłowo nagrane