

# Część I. Aplikacja internetowa

Zadanie w React polega na stworzeniu funkcjonalności dodawania produktu do tabelki. Produkt składa się z dwóch informacji: nazwy i ceny. Po dodaniu produktu, tabela powinna wyświetlać wszystkie dodane produkty. Pod tabelką znajduje się sekcja podsumowania, która wyświetla informacje o ilości produktów, maksymalnej cenie, minimalnej cenie oraz sumie cen wszystkich produktów.

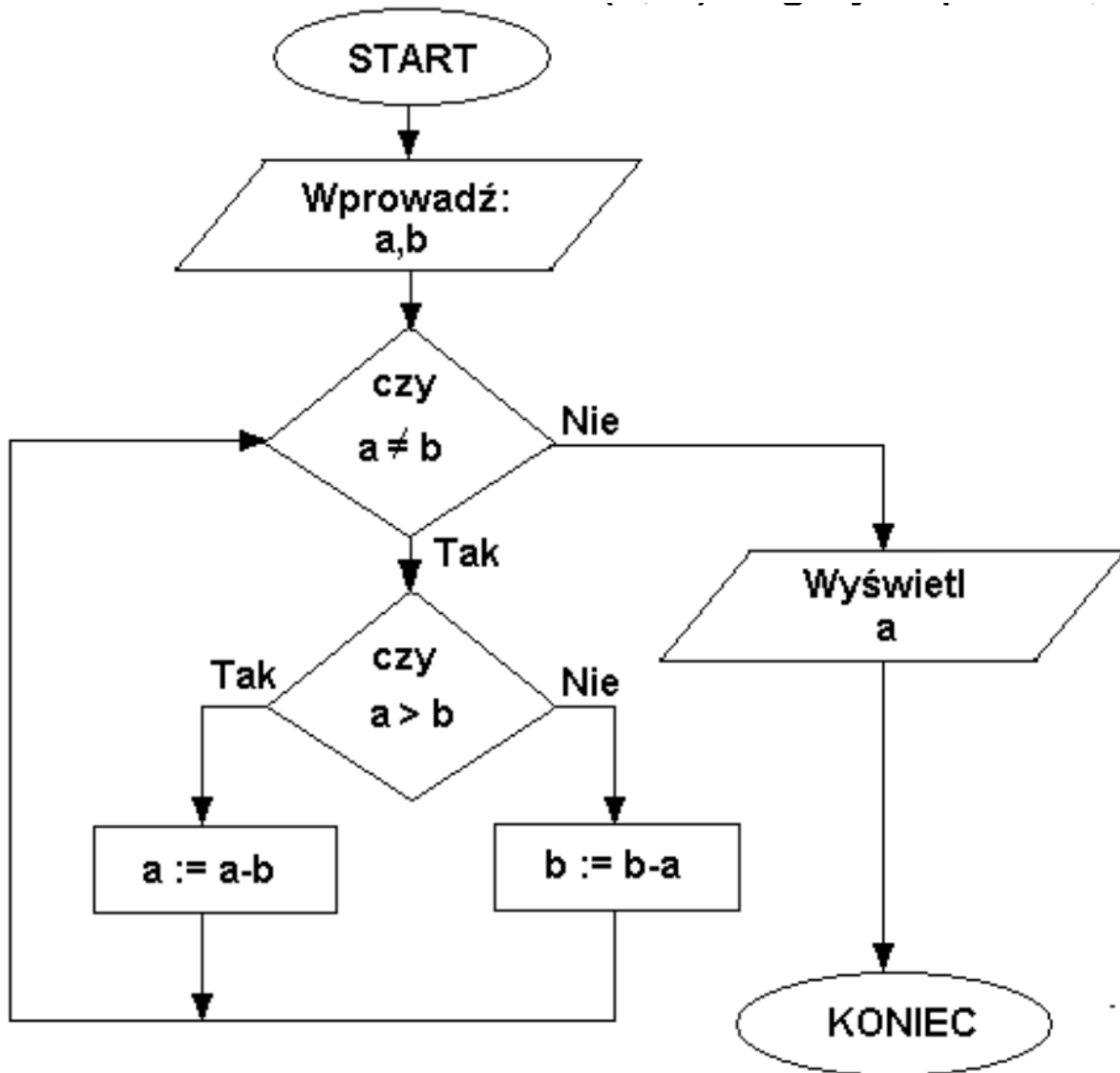
Oto opis wizualnego przedstawienia zadania:

1. Górna sekcja zawiera nagłówek "Product Table".
2. Poniżej nagłówek znajduje się tabela z dwiema kolumnami: "Name" (Nazwa) i "Price" (Cena). Każdy dodany produkt jest reprezentowany jako wiersz w tabeli, zawierający odpowiednie dane.
3. Pod tabelą znajduje się sekcja "Summary" (Podsumowanie) z nagłówkiem "Summary" oraz czterema punktami danych: "Total Products" (Łącznie produktów), "Max Price" (Najwyższa cena), "Min Price" (Najniższa cena) i "Total Price" (Łączna cena). Każdy punkt danych jest wyświetlany w osobnym wierszu.
4. Na samym dole znajduje się sekcja "Add Product" (Dodaj produkt) z nagłówkiem "Add Product", formularzem zawierającym dwa pola tekstowe ("Name" i "Price") oraz przyciskiem "Add" (Dodaj), który pozwala użytkownikowi dodać produkt do tabeli.

Dostosuj wygląd i układ graficzny według własnych potrzeb (css).

## Część II. Aplikacja konsolowa

Za pomocą narzędzi do tworzenia aplikacji konsolowych zaimplementuj algorytm Euklidesa do szukania największego wspólnego dzielnika z dwóch liczb – NWD( $a$ ,  $b$ ). Algorytm przedstawiono na schemacie.



### Algorytm Euklidesa

Założenia aplikacji:

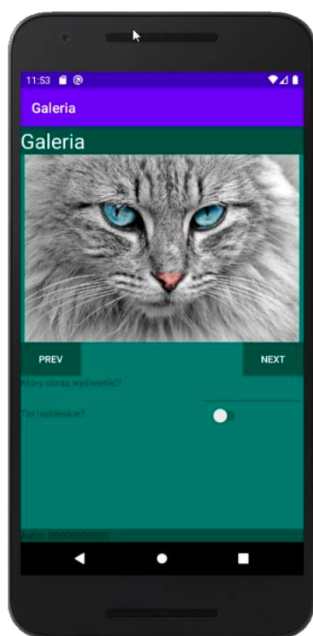
- Obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C#, lub Java, lub Python
- Implementacja algorytmu w pełni zgodna z przedstawionym na schemacie algorytmem

- Liczby  $a$  i  $b$  należą do zbioru liczb całkowitych dodatnich (odpowiedni typ lub kontrola poprawności wpisanej liczby)
- Szukanie NWD zaimplementowane w funkcji o dwóch argumentach i zwracanej wartości największego wspólnego dzielnika. Funkcja nie może zawierać operacji wejścia - wyjścia
- Program główny testuje działanie funkcji i zawiera operacje wejścia - wyjścia
- Program powinien być zapisany czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować nazwy zmiennych zgodne z nazwami zastosowanymi na schemacie blokowym, pozostałe nazwy muszą być znaczące

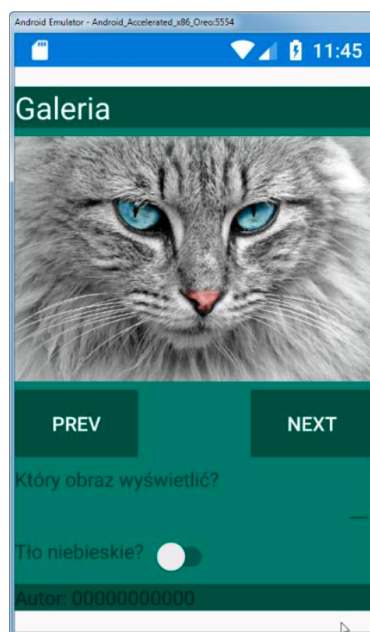
Podjmij próbę kompilacji i uruchomienia aplikacji. Kod aplikacji przygotuj do nagrania. W podfolderze *konsolowa* powinno znaleźć się archiwum całego projektu o nazwie *konsolowa.zip*

## Część III. Aplikacja mobilna

Wykonaj aplikację mobilną za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Wykorzystaj zasoby zamieszczone na w archiwum *obrazy.zip*



Obraz 1a.



Obraz 1b.

Emulacja na systemie Android Oreo.

Na obrazach 1a i 1b przedstawiono stan początkowy aplikacji mobilnej. W zależności od użytego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie się różnić.

Opis wyglądu i działania aplikacji

- Napis „Galeria”
- Poniżej obszar do przeglądania zdjęć z domyślnie załadowanym obrazem *kot1.jpg*. Kolejność wyświetlania zdjęć to kot1, kot2, kot3, kot4, kot1 (zdjęcia wyświetlają się w karuzeli)
- Poniżej przycisk PREV, którego wybranie powoduje zamianę zdjęcia na poprzednie, np. 1 na 4; 4 na 3 i tak dalej
- Obok przycisk NEXT, którego wybranie powoduje zamianę zdjęcia na następne, np. 1 na 2; 4 na 1
- Poniżej pole edycyjne poprzedzone napisem: „Który obraz wyświetlić?”. Wpisanie liczby 1-4 powoduje wyświetlenie odpowiedniego obrazu: 1 dla kot1, 2 dla kot2 i tak dalej. Wpisanie dowolnego innego znaku jest ignorowane – aplikacja nie wywołuje błędu działania
- Poniżej przełącznik (switch) poprzedzony napisem: „Tło niebieskie?”. Zaznaczenie jego powoduje zmianę tła aplikacji na kolor niebieski. Odznaczenie – powrót do koloru zielonego
- Na dole podpis o treści: „Autor: ”, dalej wstawiony numer PESEL zdającego

### Założenia aplikacji

- Obrazy kotów przechowywane w zasobach aplikacji
- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML)

- Zastosowany typ rozkładu: liniowy (Linear / Stack) lub względny (Relative), elementy rozkładu nie mogą na siebie nachodzić
- Zastosowane kolory tła: #00796B, #004C40, #1565c0, według wzoru na obrazie 1a
- Zastosowane kolory czcionki: czarny i biały, według wzoru na obrazie 1a
- Obraz jest rozciągnięty (wypełnia) cały obszar ekranu w poziomie
- Czcionka napisu „Galeria” jest wizualnie większa od pozostałych
- Przyciski są umieszczone poziomo obok siebie i przylegają do krańców ekranu po lewej lub prawej stronie
- Do pola edycyjnego można wpisać jedynie liczby
- Działanie kontrolerek obsługiwane za pomocą zdarzeń lub techniką Data Binding
- Aplikacja powinna być zapisana czytelnie, z zasadami czystego formatowania kodu, stosować znaczące nazwy zmiennych i funkcji
- Dokumentacja do programu powinna być wykonana zgodnie z wytycznymi z części III zadania egzaminacyjnego

Podejmij 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*

## 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 konsolowa – dowolna liczba zrzutów nazwanych *console1*, *console2* ...
- 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