Arkusz INF.04-02-23.01-SG

CENTRALNA KOMISJA EGZAMINACYJNA

Arkusz zawiera informacje prawnie chronione do momentu rozpoczęcia egzaminu

Nazwa kwalifikacji: **Projektowanie, programowanie i testowanie aplikacji**
Oznaczenie kwalifikacji: **INF.04**
Numer zadania: **02**
Wersja arkusza: **SG**

Wypełnia zdający

Numer PESEL zdającego*

Miejsce na naklejkę z numerem PESEL i z kodem ośrodka

Czas trwania egzaminu: **180** minut. INF.04-02-23.01-SG

EGZAMIN ZAWODOWY
Rok 2023
CZĘŚĆ PRAKTYCZNA

PODSTAWA PROGRAMOWA 2019

Instrukcja dla zdającego

1. Na pierwszej stronie arkusza egzaminacyjnego wpisz w oznaczonym miejscu swój numer PESEL i naklej naklejkę z numerem PESEL i z kodem ośrodka.
2. Na KARCIE OCENY w oznaczonym miejscu przyklej naklejkę z numerem PESEL oraz wpisz:
   - swój numer PESEL*,
   - oznaczenie kwalifikacji,
   - numer zadania,
   - numer stanowiska.
3. Sprawdź, czy arkusz egzaminacyjny zawiera 5 stron i nie zawiera błędów. Ewentualny brak stron lub inne usterki zgłoś przez podniesienie ręki przewodniczącemu zespołu nadzorującego.
4. Zapoznaj się z treścią zadania oraz stanowiskiem egzaminacyjnym. Masz na to 10 minut. Czas ten nie jest wliczany do czasu trwania egzaminu.
5. Czas rozpoczęcia i zakończenia pracy zapisze w widocznym miejscu przewodniczący zespołu nadzorującego.
6. Wykonaj samodzielnie zadanie egzaminacyjne. Przestrzegaj zasad bezpieczeństwa i organizacji pracy.
7. Po zakończeniu wykonania zadania pozostaw arkusz egzaminacyjny z rezultatami oraz KARTĘ OCENY na swoim stanowisku lub w miejscu wskazanym przez przewodniczącego zespołu nadzorującego.
8. Po uzyskaniu zgody zespołu nadzorującego możesz opuścić salę/miejsce przeprowadzania egzaminu.

Powodzenia!

** w przypadku braku numeru PESEL – seria i numer paszportu lub innego dokumentu potwierdzającego tożsamość*

Zadanie egzaminacyjne

UWAGA: katalog z rezultatami pracy oraz płytę należy opisać numerem zdającego, którym został podpisany arkusz, czyli numerem PESEL lub w przypadku jego braku numerem paszportu. Dalej w zadaniu numer ten jest nazwany numerem zdającego.

Wykonaj aplikację konsolową oraz mobilną według wskazań. Wykonaj dokumentację do obu aplikacji zgodnie z opisem w części III instrukcji do zadania. Wykorzystaj konto **Egzamin** bez hasła.

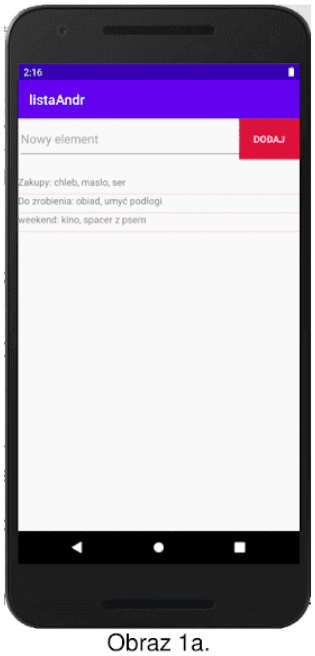
Utwórz folder i nazwij go numerem zdającego. W folderze utwórz podfoldery: *konsola*, *mobilna*, *dokumentacja*. Po wykonaniu każdej aplikacji, jej pełny kod (cały folder projektu) **spakuj do archiwum**. Następnie pozostaw w podfolderze jedynie pliki źródłowe, których treść była modyfikowana, plik uruchomieniowy, jeśli jest to możliwe oraz spakowane archiwum.

Część I. Aplikacja konsolowa
Napisz program implementujący klasę do obsługi notatek.

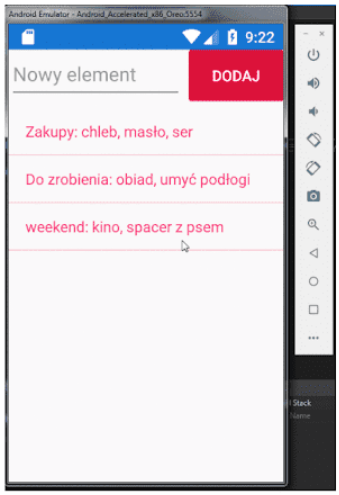
Założenia do programu:
- Program wykonywany w konsoli
- Obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C#, lub Java, lub Python
- Klasa *notatka* zawiera pola:
  - statyczne numeryczne licznika notatek do zliczania utworzonych notatek
  - numeryczne do zapisu unikalnego identyfikatora
  - dwa tekstowe do zapisu tytułu notatki i treści notatki
- Dostęp do wszystkich pól jest ograniczony do wnętrza klasy *notatka*, przy czym pola identyfikatora i licznika nie są dostępne dla klas potomnych, a pola tekstowe są dostępne dla klas potomnych
- Klasa *notatka* zawiera jeden konstruktor o parametrach wejściowych dla tytułu i treści. Ma on za zadanie kolejno:
  - inkrementować licznik notatek
  - ustawić pole identyfikatora równe licznikowi notatek, czyli pierwsza utworzona notatka ma id równe 1, druga – 2, itd.
  - ustawić pola tytułu i treści równe parametrom
- Klasa *notatka* zawiera dwie metody bezparametrowe i niezwracające wartości, które mogą być wołane w programie głównym:
  - metodę wyświetlenia tytułu i treści notatki
  - metodę diagnostyczną wypisującą zawartości wszystkich pól oddzielone od siebie średnikami
- Program powinien podejmować jasną komunikację z użytkownikiem, wyświetlane informacje powinny być zrozumiałe
- Program powinien być zapisany czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji. Wielkość liter np. dla nazwy klasy może być realizowana zgodnie z przyjętą konwencją nazewnictwa w danym języku programowania
- Program główny powinien zawierać test działania aplikacji polegający na utworzeniu dwóch notatek z dowolnymi (znaczącymi) danymi (źródło danych jest dowolne: stała napisowa, literał lub pobrane z klawiatury) oraz uruchomieniu dla każdej z nich obu metod
- Dokumentację aplikacji należy utworzyć zgodnie z opisem w części III treści zadania.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *konsola* powinno znaleźć się archiwum całego projektu o nazwie *konsola.zip*, plik z kodem źródłowym programu oraz plik uruchomieniowy, jeżeli istnieje.

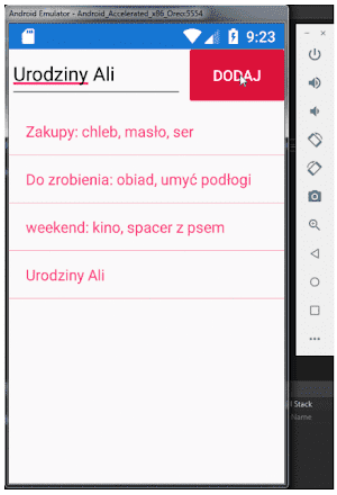
Część II. Aplikacja mobilna
Wykonaj aplikację mobilną obsługującą proste notatki tekstowe za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Aplikacja umożliwia wpisywanie prostych notatek. Dane początkowe aplikacji zapisano w pliku *dane.txt*, zawartym w archiwum *zad2.zip*, które znajduje się na pulpicie konta **Egzamin** i jest zabezpieczony hasłem: **!D@Ne$**



Obraz 1a.
Aplikacja Android Studio, stan początkowy. Emulacja Nexus 5X API 29 x86.



Obraz 1b.
Aplikacja Xamarin w MS Visual Studio, stan początkowy. Emulacja Android Oreo.



Obraz 2.
Aplikacja Xamarin, dodano element do listy. Emulacja Android Oreo.

Na obrazach 1a i 1b przedstawiono stan początkowy aplikacji mobilnej. W zależności od zastosowanego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie się różnić od przedstawionego. Obraz 2 przedstawia zachowanie aplikacji, dodano notatkę „Urodziny Ali", jest ona widoczna jako ostatni element listy

Elementy aplikacji:
- Pole edycyjne z podpowiedzią „Nowy element".
- Przycisk o treści: „DODAJ".
- Lista (element widoku listy).
- Rozmieszczenie elementów zgodne z obrazami 1a i 1b

Działanie aplikacji:
- W stanie początkowym wyświetlone są 3 notatki. Treść notatek można skopiować z pliku *dane.txt* lub ręcznie wpisać do kodu źródłowego z obrazu 1b
- Po wpisaniu do pola edycyjnego treści i wybraniu przycisku, jest ona zapisywana jako ostatni element widoku listy (obraz 2)

Założenia aplikacji:
- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML).
- Zastosowany rozkład liniowy wertykalny (Linear / Stack lub inny o tej idei) z zagłębionym rozkładem liniowym horyzontalnym dla pola edycyjnego i przycisku.
- Cechy przycisku: Kolor tła Crimson (#DC143C), biały kolor czcionki, zgodnie z Obrazem 1a.
- Cechy elementów listy: domyślny kolor czcionki, kolor separatora: Crimson, separator jest widoczny (np. poprzez ustawienie wysokości separatora)
- Notatki są zapisane w dowolnej kolekcji z elementami typu napisowego połączonej z widokiem listy jako jej źródło danych

- Wybranie przycisku powoduje dopisanie do kolekcji treści z pola edycyjnego, widok listy jest automatycznie odświeżany. Dla uproszczenia można założyć, że pole jest zawsze wypełnione, jeśli wybrany jest przycisk
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podejmij próbę kompilacji i uruchomienia aplikacji. Informacje dotyczące dokumentacji i zrzutu ekranowego umieszczono w części III zadania.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *mobilna* powinno znaleźć się archiwum całego projektu o nazwie *mobilna.zip*, plik z kodem źródłowym interfejsu użytkownika (XAML lub XML) oraz plik źródłowy kodu skojarzonego z interfejsem użytkownika.

Część III. Dokumentacja aplikacji konsolowej

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej utwórz nagłówek klasy, według wzoru. Nagłówek powinien znaleźć się nad definicją klasy. W miejscu nawiasów <> należy podać odpowiednie opisy. W miejscu autor należy podać numer zdającego.

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 klasy (liczba gwiazdek dowolna większa od 5)

```
**********************************************
  klasa:   <tu wstaw nazwę klasy>
  opis:    <co klasa reprezentuje>
  pola:    <nazwa pola1 - co przechowuje>
           <nazwa pola2 - co przechowuje>
           ... <wpisz tyle pól ile ma klasa wg. powyższego wzoru>
  autor:  <numer 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 zrzucie 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 zrzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wykonać należy co najmniej tyle zrzutów, ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja konsolowa – zrzuty nazwane: *konsola1*, *konsola2* ...
- Aplikacja mobilna – zrzuty nazwane: *mobile1*, *mobile2* ... (np. stan początkowy, w trakcie pisania, po dodaniu elementu)

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

- Nazwę systemu operacyjnego, na którym pracował zdający,
- Nazwy środowisk programistycznych, z których zdający korzystał na egzaminie,
- Nazwę emulatora dla aplikacji mobilnej,
- Nazwy języków programowania,
- Opcjonalnie komentarz do wykonanej pracy.

Zrzuty ekranu i dokument umieść w podfolderze *dokumentacja*.

UWAGA: Nagraj płytę z rezultatami pracy. W folderze z numerem zdającego powinny się znajdować podfoldery: konsola, mobilna, dokumentacja. W folderze dokumentacja: pliki ze zrzutami oraz plik egzamin. W folderze konsola: spakowany cały projekt aplikacji konsolowej, pliki źródłowe, opcjonalnie plik uruchomieniowy. W folderze mobilna: spakowany cały projekt aplikacji mobilnej, pliki z kodem źródłowym interfejsu i logiki. Po nagraniu płyty sprawdź poprawność nagrania. Opisz płytę numerem zdającego i pozostaw na stanowisku, zapakowaną w pudełku wraz z arkuszem egzaminacyjnym.

Czas przeznaczony na wykonanie zadania wynosi 180 minut.

Ocenie będą podlegać 4 rezultaty:
- implementacja, kompilacja, uruchomienie programu,
- aplikacja konsolowa,
- aplikacja mobilna,
- dokumentacja aplikacji.

Automatyczne odświeżanie widoku listy dla Xamarin z https://docs.microsoft.com

If you want the ListView to automatically update as items are added, removed and changed in the underlying list, you'll need to use an ObservableCollection. ObservableCollection is defined in System.Collections.ObjectModel and is just like List, except that it can notify ListView of any changes:

```
ObservableCollection<Type> CollectionName = new ObservableCollection<Type>();
// add elements to collection
// assign collection to ListView ItemsSource property

//to add element to ListView just add it to collection
CollectionName.Add(collectionElement);
```

Typy, metody i pola wykorzystywane do odświeżenia widoku listy dla AndroidStudio

String[]
ArrayList<String>
ArrayAdapter<String>

Arrays.asList(*tablica_napisow*)

Konstruktor i metody klasy ArrayAdapter
ArrayAdapter(Context context, int resource, int textViewResourceId, List<T> objects)
np. ArrayAdapter<String>(*this, R.layout.ListViewItemTemplate, R.id.txtItem, ArrayListObj*);
notifyDataSetChanged(); // Notifies the attached observers that the underlying data has been changed and any View reflecting the data set should refresh itself.

Metody klasy ListView
setAdapter(*AdapterObj*)