

Nazwa kwalifikacji: **Projektowanie, programowanie i testowanie aplikacji**
Symbol 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.04-02-24.01-SG

EGZAMIN ZAWODOWY

Rok 2024

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 6 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. Do pracy w systemie operacyjnym 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 z narzędziami dla typu łańcuchowego (string).

Założenia do programu:

- Program wykonywany w konsoli
- Zastosowany obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C#, lub Java, lub Python
- Klasa ma charakter biblioteki narzędziowej, nie jest przewidziane powoływanie obiektów tej klasy, wszystkie jej metody są statyczne i są widzialne poza klasą
- Klasa zawiera dwie metody:
 - Metoda licząca wszystkie samogłoski:
 - Jako parametr przyjmuje zmienną typu tekstowego
 - Zwraca liczbę samogłosek znajdujących się w parametrze
 - Przez samogłoski rozumie się znaki: „aąęiouóyAĄĘİOUÓY”
 - W przypadku, gdy parametrem jest pusty łańcuch znaków lub `null`, metoda zwraca 0
 - Metoda usuwająca powtórzenia znaków występujące obok siebie:
 - Jako parametr przyjmuje łańcuch, w którym będą szukane powtórzenia
 - Zwraca łańcuch bez powtórzeń
 - W przypadku, gdy parametrem jest pusty łańcuch lub `null`, metoda zwraca pusty łańcuch
 - Wszystkie znaki łańcucha są sprawdzane, czy znak następny jest tym samym znakiem. Jeśli tak, jest on pomijany w wyjściowym łańcuchu znaków. Przykłady:

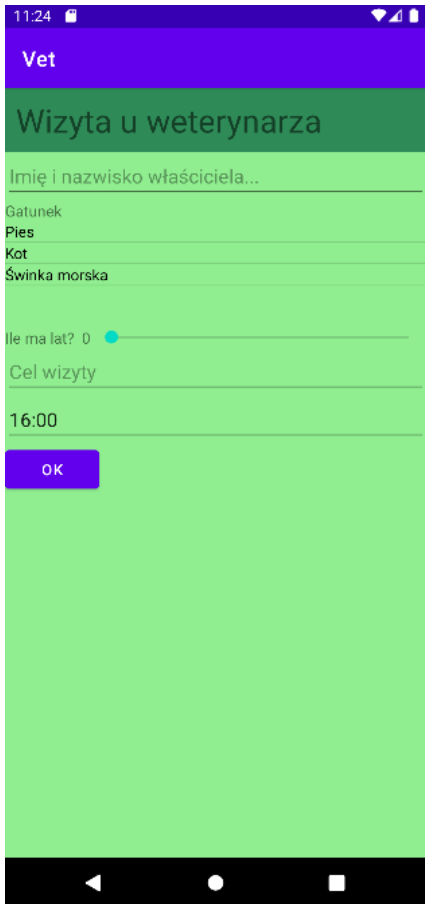
Łańcuch wejściowy	Łańcuch wyjściowy	Co zostało zmienione?
Przykład tekstu	Przykład tekstu	podwójna spacja → pojedyncza spacja
Abba ; ; ; to zespół	Aba ; to zespół	podwójne bb → pojedyncze b potrójne ; ; ; → pojedyncze ;

- W programie głównym należy przetestować obie metody, wczytując łańcuch z klawiatury i wyświetlając liczbę samogłosek oraz łańcuch po eliminacji duplikatów
- Program powinien podejmować zrozumiałą komunikację z użytkownikiem
- W programie może być zastosowane angielskie lub polskie nazewnictwo
- Program powinien być zapisany czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych, klasy i metod
- Dokumentację aplikacji należy utworzyć zgodnie z opisem w części III treści zadania.

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

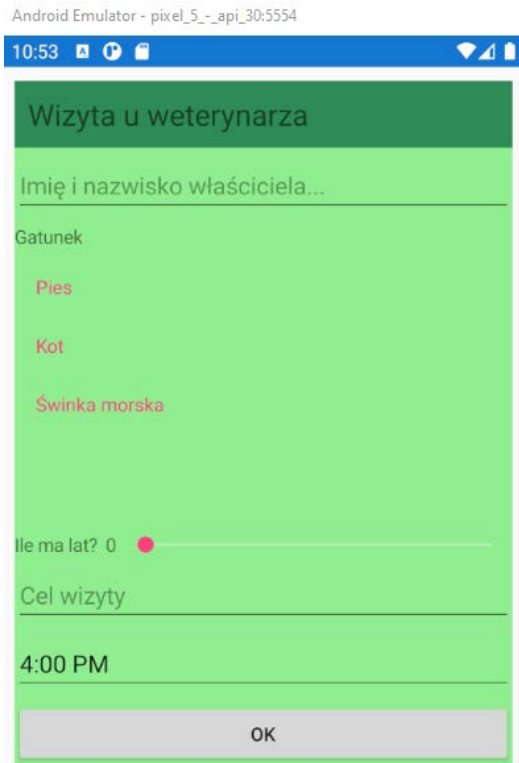
Część II. Aplikacja mobilna

Za pomocą środowiska programistycznego dostępnego na stanowisku egzaminacyjnym wykonaj aplikację mobilną obsługującą zapisy na wizytę u weterynarza.



Obraz 1a.

Środowisko AndroidStudio, stan początkowy.
Emulacja Pixel 5

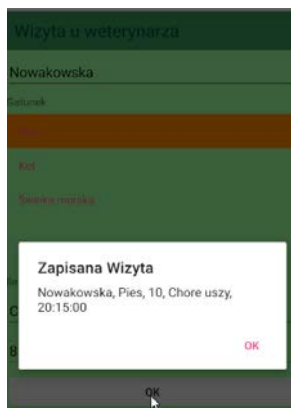


Obraz 1b.

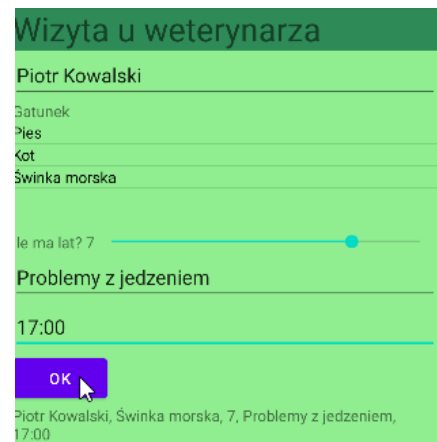
Środowisko MAUI, stan początkowy.
Emulacja Pixel 5



Obraz 2a.
Środowisko MAUI
Zmiana wartości
maksymalnej suwaka



Obraz 2b.
Środowisko MAUI
Po zatwierdzeniu,
komunikat w oknie



Obraz 2c.
Środowisko Android Studio
Po zatwierdzeniu, komunikat
pod przyciskiem

Stan początkowy aplikacji został przedstawiony na obrazach 1a i 1b. Interakcje w aplikacji zostały przedstawione na obrazach 2a, 2b, 2c. W zależności od zastosowanego środowiska programistycznego oraz emulowanego systemu wynik końcowy może nieznacznie różnić się od przedstawionego.

Elementy aplikacji:

- Pole tekstowe z tytułem „Wizyta u weterynarza”
- Pole edycyjne z podpowiedzią „Imię i nazwisko właściciela...”
- Pole tekstowe „Gatunek”
- Lista z elementami tekstowymi: „Pies”, „Kot”, „Świnka morska”
- Jedno lub dwa pola tekstowe o treści „Ile ma lat?” oraz „0” (wartość reprezentująca wiek zwierzęcia)
- Suwak, w postaci kontrolki Slider lub SeekBar o wartości początkowej 0 i wartości maksymalnej 20
- Pole edycyjne z podpowiedzią „Cel wizyty”
- Pole dedykowane do zapisu czasu z wartością początkową 4:00PM lub 16:00
- Przycisk o treści: „OK”
- Rozmieszczenie elementów zgodne z obrazami 1a lub 1b

Działanie aplikacji:

- Po kliknięciu gatunku z listy, wartość maksymalna suwaka zmienia się na:
 - 18 dla psa
 - 20 dla kota
 - 9 dla świnki morskiej
- W momencie przesuwania suwaka zmienia się odpowiednio wartość wieku w polu tekstowym obok suwaka (obraz 3). Wartość wieku jest zawsze liczbą całkowitą
- Po kliknięciu przycisku, wartości z formularza: imię i nazwisko właściciela, gatunek, wiek, cel wizyty, czas są wyświetlane w postaci napisu pod przyciskiem albo w postaci okna z komunikatem. Poszczególne wartości są rozdzielone przecinkiem (obraz 2b i 2c)
- Formularz nie musi być walidowany



Obraz 3. Interakcje suwaka z polem tekstowym dla gatunku Pies

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 zagnieżdżonym rozkładem liniowym horyzontalnym dla kontrolki dotyczących wieku (napisy i suwak)
- Kolor głównego rozkładu: LightGreen (#90EE90)
- Cechy tytułu: zapisany czcionką większą niż pozostałe kontrolki, kolor tła: SeaGreen (#2E8B57), marginesy wewnętrzne 10
- Cechy listy: wysokość dopasowana do liczby elementów
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podjmij próbę kompilacji i emulacji aplikacji. Informacje dotyczące dokumentacji i zrzutów ekranowych umieszczono w części III zadania.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *mobilna* zapisz archiwum całego projektu o nazwie *mobilna.zip*, plik (lub pliki) 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 komentarz do klasy, według wzoru z listingu 1. Komentarz powinien znaleźć się nad klasą lub pod jej nagłówkiem. W miejscu nawiasów <> należy podać odpowiednie opisy. W miejscu autor należy podać numer zdającego.

UWAGA: Dokumentację należy 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)

```
*****
klasa: <nazwa klasy>
opis: <co klasa reprezentuje i wykonuje?>
metody: <nazwa metody1 - co zwraca>
        <nazwa metody2 - co zwraca>
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 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. Należy wykonać tyle zrzutów ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja konsolowa – dokumentująca interakcje dowolna liczba zrzutów o nazwach *konsola1*, *konsola2*, ...
- Aplikacja mobilna – dokumentująca interakcje dowolna liczba zrzutów o nazwach *mobile1*, *mobile2*, ... (co najmniej stan początkowy, zmiana wartości maksymalnej suwaka, np. dla psa, zmiany suwaka, po zatwierdzeniu przyciskiem)

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać podpisane zrzuty ekranu oraz zapisane informacje dotyczące narzędzi, z których korzystał zdający podczas egzaminu:

- Nazwę systemu operacyjnego
- Nazwy środowisk programistycznych
- Nazwy języków programowania
- Nazwę emulatora dla aplikacji mobilnej

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 swoim 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.

Pomoc dla środowiska Xamarin lub MAUI

Wypełnienie listy

```
listView1.ItemsSource = lista;
```

Gdzie:

listView1 jest kontrolką ListView

lista jest obiektem typu List<string>

Zdarzenia klasy ListView: Focused, ItemTapped, ItemSelected,

Zdarzenia klasy Slider: ValueChanged, SizeChanged, Focused

Pomoc dla środowiska AndroidStudio

Wykorzystanie adaptera do wypełnienia listy

```
ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,  
R.layout.list_item, R.id.textView1, tablica_elementow);
```

```
listView1.setAdapter(arrayAdapter);
```

gdzie:

listView1 jest kontrolką ListView

list_item jest rozkładem dla elementu listy

textView1 jest id kontrolki w tym rozkładzie

tablica_elementow jest tablicą elementów, które powinny być wstawione do listy. Typ tablicy powinien być zgodny z typem elementów ArrayAdapter

Zdarzenie wybrania elementu ListView

```
listView1.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> myAdapter, View myView, int  
myItemInt, long mylng) {  
        elementWybrany = (String) listView1.getItemAtPosition(myItemInt);  
    }  
});
```

Gdzie listView1 jest kontrolką ListView

Szkielet do obsługi zdarzenia zmiany suwaka

```
seekBar1.setOnSeekBarChangeListener(new OnSeekBarChangeListener() {  
    @Override  
    public void onStopTrackingTouch(SeekBar seekBar) {}  
    @Override  
    public void onStartTrackingTouch(SeekBar seekBar) {}  
    @Override  
    public void onProgressChanged(SeekBar seekBar, int progress, boolean  
fromUser) {}  
});
```

Gdzie seekBar1 jest kontrolką SeekBar

www.EgzaminZawodowy.info

