

3.4 Przykład zadania do części praktycznej egzaminu

Część praktyczna egzaminu z kwalifikacji INF.04 jest przeprowadzana według modelu dk i trwa 180 minut.

Przykład zadania do części praktycznej egzaminu:

Wykonaj aplikację konsolową oraz mobilną według wskazań. Udokumentuj obie aplikacje zrzutami ekranu i komentarzami zgodnie z opisem w części III Dokumentacja utworzonych aplikacji.

Utwórz folder i nazwij swoim numerem PESEL. W folderze utwórz podfoldery: *konsola*, *mobilna*, *dokumentacja*. Po wykonaniu każdej aplikacji, jej pełny kod (cały projekt) skopiuj do odpowiedniego podfolderu. Dokumentację obu aplikacji w postaci zrzutów ekranu i dokumentu umieść w podfolderze *dokumentacja*.

Część I. Aplikacja konsolowa

Napisz program, który szyfruje podany przez użytkownika tekst szyfrem podstawieniowym zwanym GADERYPOLUKI. Klucz ten zawiera pary zamienników: GA-DE-RY-PO-LU-KI. Pierwsza litera w parze jest zastępowana na drugą, druga na pierwszą. Dla przykładu litera G zastępowana jest literą A, litera A literą G. Litery, których nie ma w kluczu (inne niż GADERYPOLUKI) pozostają bez zmian. Np. słowo PROGRAM po zaszyfrowaniu brzmi OYPAYGM, bo:

P	R	O	G	R	A	M
O	Y	P	A	Y	G	M

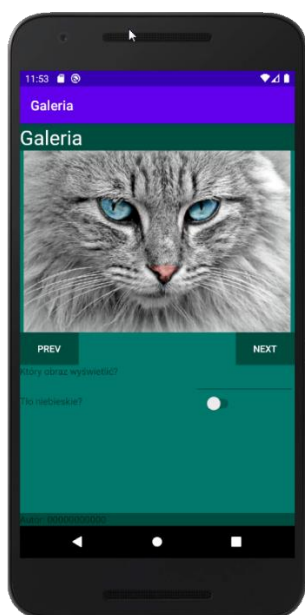
Założenia programu

- Program wykonywany w konsoli
- Język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C# lub Java lub Python
- Jeżeli język programowania tego wymaga, można założyć dla uproszczenia, że tekst do zaszyfrowania ma maksymalnie 20 liter i zapisany jest małymi literami oraz nie ma w nim innych znaków, cyfr, spacji
- Program zawiera funkcję szyfrującą, która przyjmuje jako argument wprowadzony tekst
- Funkcja zwraca zaszyfrowany tekst
- W programie głównym występuje wczytanie tekstu z klawiatury po uprzednim wyświetleniu stosownego komunikatu dla użytkownika, a po zaszyfrowaniu wyświetlenie zaszyfrowanej jego wersji
- Program powinien być zapisany czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji
- Dokumentacja do programu powinna być wykonana zgodnie z wytycznymi z części III zadania egzaminacyjnego

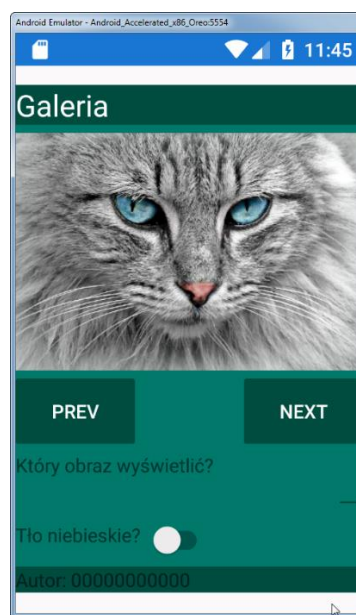
Kod aplikacji (cały projekt) przygotuj do nagrania na płytę (skopiuj do podfolderu *konsola*).

Część II. 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 pulpicie w archiwum *obrazy.zip* z hasłem: **Galeri@Zdjec#**



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 kontrolki 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

Kod aplikacji (cały projekt) przygotuj do nagrania na płytę (skopiuj do podfolderu *mobilna*).

Część III. Dokumentacja utworzonych aplikacji

Wykonaj dokumentację aplikacji utworzonych podczas egzaminie. W kodzie źródłowym aplikacji szyfrującej utwórz nagłówek funkcji szyfrującej według wzoru. Nagłówek powinien znaleźć się w kodzie źródłowym nad funkcją. W miejscu nawiasów <> należy podać nazwę funkcji, nazwy parametrów oraz zwięzłe informacje (kilka słów) – zgodnie ze wzorcem. W miejscu autor należy podać swój numer PESEL

Listing 1. Wzór dokumentacji funkcji

```
*****
* nazwa funkcji:    <tu wstaw nazwę funkcji>
*
* parametry wejściowe: <nazwa parametru> - <co przechowuje parametru>
* wartość zwracana:  <co zwraca funkcja – opis>
* opis funkcji:     <zwięzły opis>
*
* autor:            <numer PESEL zdającego>
* *****/
```

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji. Zrzuty powinny obejmować cały obszar ekranu z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na zrzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem pod spodem. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na zrzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wymagane zrzuty ekranu:

- z aplikacji konsolowej – *konsola.jpg*
- z aplikacji mobilnej – *mobilna.jpg*

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

- nazwę systemu operacyjnego, na którym pracował zdający
- nazwy środowisk programistycznych, z których zdający korzystał na egzaminie
- nazwy języków programowania użytych podczas tworzenia aplikacji
- nazwę emulowanego urządzenia lub systemu, na którym uruchomiono aplikację mobilną
- opcjonalnie komentarz do wykonanej pracy.

Całą dokumentację umieścić w podfolderze *dokumentacja*.

UWAGA: Nagraj płytę z rezultatami pracy. W folderze z numerem PESEL powinny się znajdować podfoldery dokumentacja, konsola, mobilna. W folderze dokumentacja pliki: konsola.jpg, mobilna.jpg, egzamin. W folderze konsola: cały projekt aplikacji konsolowej. W folderze mobilna: cały projekt aplikacji mobilnej, ewentualnie inne przygotowane pliki. Po nagraniu płyty sprawdź poprawność nagrania. Opisz płytę swoim numerem PESEL i zapakowaną w pudełku pozostaw na stanowisku, 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.

Komentarz do zadania

Zadanie w kwalifikacji INF.04 składa się z trzech części. W części pierwszej są sprawdzane umiejętności pisania aplikacji obiektowych i implementacji algorytmów. W części drugiej programowanie aplikacji okienkowych lub mobilnych lub internetowych. W części trzeciej dokumentacja aplikacji lub testy.

Ocena zadania dotyczy nie tylko uruchomienia i przetestowania aplikacji, ale też stylu pisania kodu. Kod sprawdzany jest pod względem czytelności, zasad pisania czystego kodu takich jak wcięcia, konwencja zapisu nawiasów klamrowych, spacje przy operatorach, nazewnictwo zmiennych i funkcji. Wymagane też jest udokumentowanie rzutem ekranu próby kompilowania i uruchomienia programu.

Efekty kształcenia sprawdzane przykładowym zadaniem praktycznym wraz z kryteriami weryfikacji:

<i>Jednostka efektów kształcenia:</i> INF.04.3 Projektowanie oprogramowania	
<i>Efekty kształcenia</i>	<i>Kryteria weryfikacji</i>
Uczeń (zdający):	Uczeń (zdający):
1) posługuje się prostymi typami danych	4) posługuje się typami liczbowymi stałoprzecinkowe i zmiennoprzecinkowymi, typem logicznym, typem znakowym i łańcuchowym
2) posługuje się złożonymi typami danych	9) projektuje zestawy danych dla problemu programistycznego
6) projektuje aplikację	1) analizuje wymagania klienta i tworzy zgodny z nimi projekt 4) projektuje interfejs użytkownika i wygląd aplikacji

<i>Jednostka efektów kształcenia:</i> INF.04.4 Programowanie obiektowe	
<i>Efekty kształcenia</i>	<i>Kryteria weryfikacji</i>
Uczeń (zdający):	Uczeń (zdający):
1) wykorzystuje środowisko programistyczne dla obiektowych aplikacji konsolowych	6) kompiluje i uruchamia programy
2) przestrzega zasad programowania	1) dzieli program na funkcje (metody) 3) implementuje algorytmy w programie

4) stosuje wyrażenia, instrukcje i biblioteki	4) stosuje instrukcje pętli
6) definiuje klasy	3) definiuje metody klasy

Jednostka efektów kształcenia:

INF.04.6 Programowanie aplikacji mobilnych

<i>Efekty kształcenia</i>	<i>Kryteria weryfikacji</i>
Uczeń (zdający):	Uczeń (zdający):
2) programuje aplikacje mobilne	3) programuje przechowywanie danych i preferencji użytkownika w aplikacjach mobilnych 5) programuje interfejs użytkownika za pomocą języka XAML 6) programuje aplikacje mobilne przy wykorzystaniu elementów UI 11) uruchamia aplikacje mobilne

Jednostka efektów kształcenia:

INF.04.8 Testowanie i dokumentowanie aplikacji

<i>Efekty kształcenia</i>	<i>Kryteria weryfikacji</i>
Uczeń (zdający):	Uczeń (zdający):
2) dokumentuje tworzoną aplikację	1) stosuje komentarze w kodzie źródłowym programu 2) pisze dokumentację kodu 3) pisze dokumenty pomocy programu 6) pisze dokumentację wdrożenia projektu

Inne zadania z kwalifikacji INF.04 mogą dotyczyć

W części I:

- implementacji algorytmów sortowania
- implementacji algorytmów poszukiwań, szyfrowania
- implementacji algorytmów operujących na wybranych typach danych (np. operacje na zmiennych napisowych)
- implementacji różnych algorytmów zapisanych w postaci opisu, kroków, schematu blokowego
- projektowania zestawu danych i typów do zadanego problemu
- projektowania klasy z zestawem pól i metod obsługującej konkretne zagadnienie

W części II:

- implementacji aplikacji okienkowej obsługującej okno dialogowe do wpisania danych, system menu aplikacji
- projektowania zestawu okien dialogowych dla aplikacji okienkowej
- implementacji aplikacji mobilnej obsługującej różne rozkłady (layouts) i zawierającej kontrolki takie jak: pola edycyjne, edytor, pole wyboru (spinner), przełącznik (switch), pasek postępu (slider), stepper, picker, data i czas, przyciski i inne
- implementacji aplikacji mobilnej obsługującej obrazy
- implementacji aplikacji mobilnej z systemem menu i przechodzeniem do podstron
- projektowania interfejsu użytkownika z wykorzystaniem języków znaczników (np. XAML, XML)
- implementacji aplikacji internetowych przy wykorzystaniu Angular lub ReactJS (do wyboru) obejmujących wykonanie komponentu z kontrolkami i ich walidacją oraz obsługą zdarzeń
- implementacji aplikacji internetowych przy wykorzystaniu Angular lub ReactJS (do wyboru) złożonych z kilku podstron
- obsługi komunikacji części front-end z serwerem za pomocą JSON
- implementacji części serwerowej aplikacji internetowej w środowisku Node.js lub Django lub ASP.NET (do wyboru) i zarządzania danymi z bazy

W części III:

- dokumentacji aplikacji, kodu
- utworzenia instrukcji dla użytkownika
- utworzenia testów jednostkowych dla aplikacji przy wykorzystaniu frameworka testowania do wybranego języka programowania
- utworzenia dokumentacji do scenariuszy testowania danej aplikacji
- zaproponowanie serii testów dotyczących np. badania wydajności systemu, badania wartości brzegowych dla zmiennych i innych