

Nazwa  
kwalifikacji:  
Oznaczenie  
kwalifikacji:

Numer zadania:

Kod arkusza:

Wersja arkusza:

## Projektowanie, programowanie i testowanie aplikacji

**INF.04**

**01**

**INF.04-01-24.01-SG**

**SG**

Lp.	Elementy podlegające ocenie/kryteria oceny
<b>R.1</b>	<b>Rezultat 1: Implementacja, kompilacja, uruchomienie programu</b>
	<i>Uwaga: kryteria należy odnieść do aplikacji konsolowej, jeżeli ta nie istnieje, kryteria 1.1 ÷ 1.6 zastosować do aplikacji desktopowej. Kryteria dotyczą wyłącznie samodzielnie napisanego kodu. Wystarczy, że sprawdzaną cechę zastosowano dla większości przypadków w kodzie</i>
R.1.1	Kod źródłowy zapisano w sposób czytelny: instrukcje w osobnych liniach, stosowane spacje pomiędzy operatorami, konsekwentnie stosowana wybrana konwencja dla nawiasów klamrowych instrukcji blokowej
R.1.2	Kod zapisano z wcięciami dla zagłębień bloków
R.1.3	Użyto polskie lub angielskie nazewnictwo funkcji. Nazewnictwo jest znaczące
R.1.4	Użyto polskie lub angielskie nazewnictwo zmiennych. Nazewnictwo jest znaczące. Wyjątkami od reguły są zmienne bufor, tmp, iteratory pętli, oraz zmienne wynikające z algorytmu: S, M, R. Kryterium <b>nie jest</b> spełnione tylko wtedy, gdy nazwy zmiennych nic nie znaczą, np. x, tab, tablica, foo
R.1.5	Zastosowano typy pasujące do problemu oraz numer PESEL jest przechowywany w zmiennej tekstowej albo tablicy elementów liczbowych lub znakowych. W przypadku porównywania znaku z liczbą stosowano odpowiednią konwersję (np. znak - '0'). W przypadku Python zastosowano jawną konwersję do odpowiednich typów
R.1.6	Podjęto próbę skompilowania kodu lub uruchomienia w interpreterze, co udokumentowano zrzutem przedstawiającym uruchomiony program lub jego kompilację
R.1.7	Program podejmuje komunikację z użytkownikiem: monit o wprowadzenie danych jest znaczący lub wyświetlenie danych opatrzone znaczącym komunikatem, wyświetlana jest płęć w postaci całego słowa "Kobieta" lub "Mężczyzna"
<b>R.2</b>	<b>Rezultat 2: Aplikacja konsolowa</b>
	<i>Uwaga: kryteria 2.1 ÷ 2.7 należy sprawdzić w kodzie programu, sprawdzane elementy muszą być zapisane zgodnie ze składnią Gdy aplikacja nie uruchamia się, a zdający zapisał zrzuty ekranu z uruchomienia aplikacji należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego kryterium 2.8 nie jest spełnione. Jeżeli błędy występują w innych plikach ocenić na podstawie kodu i zrzutu ekranu</i>
R.2.1	W programie zdefiniowano przynajmniej jedną funkcję, która pobiera jako argument zmienną przechowującą PESEL (typ musi być zgodny) oraz zastosowano instrukcję return dla każdej ścieżki decyzyjnej
R.2.2	Funkcja sprawdzająca płęć jest typu znakowego, zwraca 'K' gdy 10. cyfra numeru PESEL jest parzysta (0, 2, 4, 6, 8), zwraca 'M', gdy jest nieparzysta (1, 3, 5, 7, 9) (warunki komplementarne - można zastosować samo else)
R.2.3	Funkcja sprawdzająca sumę kontrolną jest typu logicznego, zwraca <code>true</code> gdy obliczona suma kontrolna jest równa 11. cyfrze PESEL lub <code>false</code> w przeciwnym wypadku
R.2.4	Wykonano pierwszy etap liczenia sumy kontrolnej: dla pierwszych 10 cyfr numeru PESEL zostały obliczone ich iloczyny z wagami (1, 3, 7, 9, 1, 3, 7, 9, 1, 3)
R.2.5	Wykonano 2. etap liczenia sumy kontrolnej: sumowanie obliczonych 10 iloczynów (należy sprawdzić jedynie sumowanie nawet jeśli wagi te byłyby błędne)
R.2.6	Wykonano 3. i 4. etap liczenia sumy kontrolnej: obliczono wynik $M = S \% 10$ oraz gdy $M > 0$ to $R = 10 - M$ ; gdy $M = 0$ to $R = 0$ Mogą być zastosowane inne nazwy zmiennych niż M, S, R

R.2.7	W programie głównym wywołano co najmniej jedną funkcję dla numeru PESEL wczytanego z klawiatury
R.2.8	Program uruchamia w konsoli, co udokumentowano zrzutem ekranu
<b>R.3</b>	<b>Rezultat 3: Aplikacja desktopowa</b>
	<p><i>Uwaga: Kryteria 3.1 ÷ 3.5 sprawdzić w kodzie źródłowym, sprawdzane elementy muszą być zapisane zgodnie ze składnią.</i></p> <p><i>Gdy aplikacja nie uruchamia się, a zdający zapisał zrzuty ekranu z uruchomienia aplikacji należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego, kryteria 3.6 ÷ 3.10 nie są spełnione. Jeżeli błędy występują w innych plikach lub bibliotekach, sprawdzić w kodzie oraz na zrzutach ekranu.</i></p> <p><i>Dopuszcza się literówki w wyświetlanych napisach</i></p>
R.3.1	Dla okna aplikacji nadano nazwę "Wprowadzenie danych do paszportu. Wykonał: " wraz z numerem zdającego oraz umieszczono w oknie trzy pola edycyjne, trzy pola opcji, dwa obrazy i przycisk
R.3.2	Nadano dla okna tło koloru CadetBlue (#5F9EA0) oraz dla pól edycyjnych i przycisku tło koloru Azure (#F0FFFF)
R.3.3	Pola opcji zgrupowano: jednocześnie można wybrać tylko jedno pole, w stanie początkowym wybrane jest pole "niebieskie"
R.3.4	Po opuszczeniu pola edycyjnego do wpisywania numeru lub po wciśnięciu przycisku wywoływana jest funkcja
R.3.5	Obrazom nadano tą samą wysokość
R.3.6	Po wypełnieniu pola edycyjnego do wpisania numeru danymi: 111, 333, 000 są wyświetlone oba obrazy, nazwy plików graficznych zgadzają się z podanym numerem np. dla numeru 111 są to <i>111-zdjecie.jpg</i> oraz <i>111-odcisk.jpg</i> (sprawdzić w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
R.3.7	W przeciwnym do R.3.6 przypadku, żaden obraz nie został wyświetlony (sprawdzić w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
R.3.8	Jeżeli dane zostały wprowadzone do wszystkich pól edycyjnych, po wybraniu przycisku zatwierdzającego, wyświetlane jest okienko komunikatu o treści: „<imie> <nazwisko> kolor oczu <kolor>", gdzie pola w nawiasach <> zostały pobrane z kontrolek (sprawdzić w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
R.3.9	W przeciwnym do R.3.8 przypadku wyświetlane jest okienko komunikatu „Wprowadź dane” (sprawdzić w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
R.3.10	Aplikacja uruchamia się, co udokumentowano zrzutem ekranu, jej układ po uruchomieniu jest zgodny z obrazem 1 w arkuszu egzaminacyjnym. Widoczne są obrazy <i>000-zdjecie.jpg</i> oraz <i>000-odcisk.jpg</i> (sprawdzić w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
<b>R.4</b>	<b>Rezultat 4: Dokumentacja aplikacji</b>
	<p><i>Uwaga: nagłówek oceniany w kryteriach 4.1 ÷ 4.5 musi być zgodny ze stanem faktycznym z kodu źródłowego, nawet jeżeli w kodzie są błędy logiczne (liczba pól, typy).</i></p> <p><i>Zrzuty ekranu oceniane w kryteriach 4.6 i 4.7 muszą zawierać cały obszar ekranu z widocznym paskiem zadań. Dokumentacja z kryterium 4.8 zapisana jest w pliku egzamin</i></p>
R.4.1	Dla dowolnej funkcji z aplikacji konsolowej zapisano nagłówek w postaci komentarza zgodny z Listingiem 1 z arkusza egzaminacyjnego (nie należy brać pod uwagę liczby gwiazdek), komentarz może być wieloliniowy lub kilka jednoliniowych, lub Docstrings (potrójny cudzysłów) - w ostatnim przypadku opis znajduje się pod słowem class (na początku definicji klasy)
R.4.2	W komentarzu ujęto nazwę funkcji oraz opisano jej działanie
R.4.3	W komentarzu ujęto wszystkie argumenty funkcji: nazwy i krótki opis
R.4.4	W komentarzu ujęto nazwę typu zwracanego i krótki opis co jest zwracane
R.4.5	W komentarzu ujęto numer zdającego
R.4.6	Zapisano przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji konsolowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja

R.4.7	Zapisano przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji desktopowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja
R.4.8	Dokumentacja zawiera: nazwę systemu operacyjnego, nazwy środowisk, nazwy języków programowania