

Nazwa  
kwalifikacji:

## Projektowanie, programowanie i testowanie aplikacji

Oznaczenie  
kwalifikacji:

INF.04

Numer zadania:

01

Kod arkusza:

INF.04-02-23.06-SG

Wersja arkusza:

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, zastosować 1.1 ÷ 1.6 do aplikacji desktopowej</i> <i>Wystarczy, że sprawdzaną cechę zastosowano dla większości przypadków w kodzie</i> <i>Kryteria z R.1 dotyczą kodu własnego zdającego</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	Zastosowano znaczące nazewnictwo funkcji lub metod
R.1.4	Zastosowano znaczące nazewnictwo zmiennych. Wyjątkami od reguły są zmienne bufor, tmp, iteratory pętli itp. Kryterium <b>nie jest</b> spełnione tylko wtedy, gdy nazwy zmiennych nic nie znaczą, np. Użyto znaczące nazewnictwo zmiennych. Wyjątkami od reguły są zmienne bufor, tmp, iteratory pętli itp. Kryterium nie jest spełnione tylko wtedy, gdy nazwy zmiennych nic nie znaczą, np. x, a, tab, tablica, fun
R.1.5	Zastosowano typy zmiennych pasujące do problemu (np. dowolny typ całkowity dla tablicy; typ całkowity dla iteratora pętli) w przypadku Python, tam gdzie jest to wymagane zastosowano jawną konwersję do odpowiednich typów
R.1.6	Podjęto próbę uruchomienia kodu, co udokumentowano zrzutem ekranowym przedstawiającym wykonywany program lub jego kompilację
R.1.7	Program po uruchomieniu wyświetla liczby rozdzielone dowolnym separatorem, widoczny jest znaczący komunikat opisujący co zostało wyświetlone. Jeżeli kod nie uruchamia się - sprawdzić w kodzie aplikacji
<b>R.2</b>	<b>Rezultat 2: Aplikacja konsolowa</b>
	<i>Uwaga: R.2.1 ÷ R.2.4 należy sprawdzić w kodzie programu, sprawdzane elementy muszą być zapisane zgodnie ze składnią.</i> <i>Gdy aplikacja nie uruchamia się, a zdający zapisał zrzut ekranu z uruchomienia aplikacji należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego kryterium 2.5 nie jest spełnione. Jeżeli błędy występują w innych plikach ocenić na podstawie kodu i zrzutu ekranu</i>
R.2.1	Program składa się z programu głównego oraz przynajmniej jednej funkcji lub metody do sortowania. Zawartość funkcji jest zgodna z przeznaczeniem. Dopuszcza się aby funkcja była niedokończona, ale zapisy w niej wskazywały na proces sortowania
R.2.2	Utworzono funkcję sortującą która jako argument przyjmuje tablicę i nie zwraca żadnej wartości (np. void)
R.2.3	Zastosowano algorytm sortowania (wystarczy że elementy kodu będą na to wskazywać)
R.2.4	Na proces sortowania składają się dwie pętle operujące na wszystkich wymaganych do prawidłowego posortowania elementach tablicy. Pętle są sterowane różnymi zmiennymi
R.2.5	W pętli wewnętrznej znajduje się porównanie dwóch sąsiednich elementów tablicy gwarantujące sortowanie bąbelkowe rosnące
R.2.6	Zapisano zamianę elementów miejscami przy pomocy zmiennej buforowej lub w inny sposób, który nie powoduje utraty danych lub w Python zapis typu a, b = b, a
R.2.7	Program uruchamia się w konsoli, co udokumentowano zrzutem ekranu
<b>R.3</b>	<b>Rezultat 3: Aplikacja mobilna</b>

*Uwaga: jeżeli jest to możliwe uruchomić aplikację na tym samym urządzeniu, na którym uruchamiał zdający. Należy uwzględnić różnice pomiędzy emulacjami - takie cechy jak marginesy, wielkości bloków itp. nie bierzemy pod uwagę. Na emulacji może nie być widoczny cały ekran, przewinąć lub sprawdzić w kodzie, czy zawarto wszystkie wymagane elementy. Kryteria R.3.1 ÷ R.3.6 sprawdzić w kodzie źródłowym, 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 kryteria 3.7 ÷ 3.10 nie są spełnione. Jeżeli błędy występują w innych plikach lub bibliotekach sprawdzić w kodzie oraz na zrzucie ekranu*

*Dopuszcza się literówki w napisach i komunikatach*

R.3.1	Zastosowano język znaczników XML/XAML lub inny do opisu interfejsu użytkownika oraz kod zawiera przynajmniej jedną kontrolkę wynikającą z treści zadania
R.3.2	Zastosowano rozkład liniowy (np. StackLayout, LinearLayout lub inny o tej idei), pola edycyjne są rozciągnięte na całą szerokość okna (np. fill lub match_parent), ustawiono marginesy dla całej strony lub rozkładu na 20
R.3.3	Zastosowano kontrolki: tekstu, suwaka, przycisku. Tekst "Właściwości czcionki" oraz tekst przycisku mają cechy: biały kolor czcionki, kolor tła #558B2F, pogrubione. Tekst "Właściwości czcionki" jest zapisany czcionką o największym rozmiarze, "Rozmiar: " - średnią
R.3.4	Suwak ma wartość maksymalną 40
R.3.5	Obsłużono przycisk za pomocą zdarzenia kliknięcia lub obsłużono suwak za pomocą zdarzenia zmiany wartości. Po wystąpieniu zdarzenia wywoływana jest funkcja obsługująca zdarzenie lub zapisano obsługę suwaka za pomocą powiązania (Binding)
R.3.6	Napisy "Dzień dobry", "Good morning", "Buenos dias" zakodowano w tablicy 3-elementowej typu napisowego, np. String[]
R.3.7	Aplikacja kompiluje się i uruchamia w emulatorze, co udokumentowano zrzutem ekranu, jej wygląd jest zgodny z obrazem 1a lub 1c w arkuszu egzaminacyjnym. Zawiera wszystkie elementy. Przycisk jest wyśrodkowany oraz napis cytatu brzmi "Dzień dobry"
R.3.8	Po przesunięciu suwaka jego aktualna wartość jest wyświetlana w napisie "Rozmiar: "
R.3.9	Po przesunięciu suwaka jest zmieniany rozmiar czcionki napisu cytatu na wartość pobraną z suwaka (tylko wartości całkowite)
R.3.10	Po wybraniu przycisku, zmienia się napis cytatu w karuzeli, czyli 1-2-3-1...
<b>R.4</b>	<b>Rezultat 4: Testy aplikacji konsolowej</b>
<i>Uwaga: zrzuty ekranu z kryteriów 4.4 i 4.5 powinny zawierać cały obszar ekranu z widocznym paskiem zadań. Nie sprawdzamy wykadrowanych zrzutów</i>	
R.4.1	Test funkcji sortującej zawiera odwołanie do tej funkcji z prawidłowo przekazany parametrem
R.4.2	Tablica jest wypełniana liczbami pseudolosowymi z zakresu 0..1000
R.4.3	Działanie programu testującego polega na wyświetleniu posortowanych rosnąco elementów tablicy
R.4.4	Zapisano zrzut ekranu z uruchomienia lub kompilacji aplikacji konsolowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja oraz wypisany jest wynik działania programu lub błędy kompilacji
R.4.5	Zapisano zrzut ekranu z uruchomienia lub kompilacji aplikacji mobilnej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja