

Nazwa
kwalifikacji:
Oznaczenie

Projektowanie, programowanie i testowanie aplikacji

INF.04

kwalifikacji:

Numer zadania: 02

Kod arkusza: INF.04-02-25.01-SG

Wersja arkusza: SG

Lp.	Elementy podlegające ocenie/kryteria oceny
R.1	Rezultat 1: Implementacja, kompilacja, uruchomienie programu
Uwaga: kryteria należy odnieść do aplikacji konsolowej, jeżeli ta nie istnieje, zastosować do aplikacji mobilnej. Kryteria dotyczą wyłącznie samodzielnie napisanego kodu. Wystarczy, że sprawdzana cecha zastosowana dla większości (90%) przypadków w kodzie	
R.1.1	Kod źródłowy zapisany w sposób czytelny: instrukcje w osobnych liniach, stosowane spacje pomiędzy operatorami, konsekwentnie stosowana wybrana konwencja dla nawiasów klamrowych
R.1.2	Kod zapisany z wcięciami dla zagnieżdżeń bloków
R.1.3	Użyte polskie lub angielskie, znaczące nazewnictwo klas, metod, obiektów
R.1.4	Użyte polskie lub angielskie, znaczące nazewnictwo pól i zmiennych. Wyjątkami od reguły są zmienne: bufor, tmp, iteratory pętli itp. Kryterium <u>nie jest</u> spełnione tylko wtedy, gdy nazwy zmiennych nic nie znaczą, np.: x, fun, foo, tab, tablica
R.1.5	Zastosowane typy zmiennych pasujące do problemu (np. typ boolowski jako stan, typ liczb całkowitych jako program pralki, typ tekstowy dla komunikatu) W przypadku języka Python typ wyniku z przypisanych danych
R.1.6	Podjęta próba uruchomienia kodu, co udokumentowano zrzutem ekranowym przedstawiającym uruchomiony program lub jego kompilację
R.1.7	Program podejmuje komunikację z użytkownikiem, wyświetlane komunikaty są znaczące
R.2	Rezultat 2: Aplikacja konsolowa
Uwaga: kryteria 2.1 ÷ 2.6 należy sprawdzić w kodzie programu, sprawdzane elementy muszą być zapisane zgodnie ze składnią. Gdy aplikacja nie uruchamia się, a są zapisane zrzuty ekranu z uruchomienia aplikacji, należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego, kryteria 2.7 ÷ 2.10 nie są spełnione. Jeżeli błędy występują w innych plikach należy ocenić na podstawie kodu i zrzutu ekranu W przypadku języka Python argument self nie jest wliczany do liczby parametrów oraz deklaracja pól w konstruktorze przez self. np. self.pole	
R.2.1	Program składa się z programu głównego oraz trzech klas, jednej bazowej i dwóch dziedziczących po niej. W programie głównym są powoływane obiekty obu klas potomnych oraz wywołano na ich rzecz co najmniej jedną metodę
R.2.2	Klasa bazowa implementuje metodę o zasięgu public, która wyświetla treść komunikatu. Komunikat jest argumentem metody. Metoda nie zwraca wartości
R.2.3	Klasa pralki definiuje pole numeru programu typu całkowitego o zakresie private, inicjowane wartością 0 oraz klasa odkurzacza definiuje pole stanu odkurzacza typu logicznego o zasięgu private, inicjowane wartością fałsz. (W języku Python, dla pola typu private, dwa podkreślniki przed nazwą)

R.2.4	Klasa implementuje metodę ustawienia numeru programu o zakresie public. Argumentem metody jest numer programu. Gdy numer programu jest liczbą z przedziału od 1 do 12 metoda ustawia na tą wartość pole numeru programu. W przeciwnym wypadku pole jest ustawione na wartość 0. Metoda zwraca wartość pola numeru programu
R.2.5	Klasa odkurzacza implementuje przynajmniej jedną metodę o zakresie public: on() lub off(). Metoda nie zwraca wartości i jest bezargumentowa. Ustawiany jest odpowiedni stan odkurzacza

R.2.6	Metody on() lub off() wywołują metodę do wyświetlenia komunikatu lub wyświetlają komunikat o włączeniu lub wyłączeniu odkurzacza, tylko wtedy, gdy jego stan uległ zmianie
R.2.7	Program uruchamia się w konsoli, co jest udokumentowane zrzutem ekranu
R.2.8	Program pyta o numer prania i wczytuje liczbę. Dla liczb 1..12 wyświetlony jest komunikat o ustawieniu programu, dla pozostałych o niepoprawnym numerze programu
R.2.9	Program wyświetla komunikat o wyładowaniu odkurzacza
R.2.10	Niezależnie od następujących po sobie kilku wywołań metody on() tylko raz wyświetlany jest komunikat o włączeniu odkurzacza lub w przypadku metody off() o wyłączeniu odkurzacza
R.3	Rezultat 3: Aplikacja mobilna
<p>Uwaga: należy uwzględnić różnice pomiędzy emulacjami - nie należy brać pod uwagę takich cech jak: marginesy, wielkości bloków, itp. Kryteria 3.1 ÷ 3.7 sprawdzić w kodzie źródłowym, sprawdzane elementy muszą być zapisane zgodnie ze składnią.</p> <p>Gdy aplikacja nie uruchamia się, a są zapisane zrzuty ekranu z uruchomienia aplikacji, należy sprawdzić powód braku kompilacji. Jeśli występują błędy w plikach źródłowych zdającego kryteria 3.8 ÷ 3.10 nie są spełnione. Jeżeli błędy występują w innych plikach lub bibliotekach sprawdzić w kodzie oraz na zrzucie ekranu. Dla Android Studio dopuszcza się także rozwiązanie w języku Kotlin</p>	
R.3.1	Zastosowany język znaczników XML/XAML lub inny do opisu interfejsu użytkownika oraz zastosowany układ rozkładów liniowych wertykalnych i horyzontalnych (LinearLayout lub StackLayout, lub VerticalStackLayout i HorizontalStackLayout, lub inne o tej idei) tak, aby układ ten był zgodny z obrazem 3 arkusza egzaminacyjnego (oceniane są tylko znaczniki rozkładu)
R.3.2	Umieszczone: 2 etykiety „Urządzenia Domowe” i "autor: " z numerem zdającego, obrazy pralka.jpg i odkurzacz.jpg, napisy "Pralka", "Odkurzacz", pole edycyjne z podpowiedzią (placeholder lub hint) i możliwością wpisania jedynie liczb, przyciski "Zatwierdź" i "Włącz" oraz pozostałe 3 napisy z opisem stanu urządzeń
R.3.3	Zastosowane kolory tła: dla głównego rozkładu lub strony: LightBlue (#ADD8E6), dla pola edycyjnego: SkyBlue (#87CEEB), dla przycisków: RoyalBlue (#4169E1) oraz kolor czcionki i kolor podpowiedzi Navy (#000080) dla pola edycyjnego (w MAUI nazwy kolorów obowiązkowo wielką literą)
R.3.4	Zastosowana czcionka powiększona dla napisów "Urządzenia Domowe", "Pralka", "Odkurzacz", wyśrodkowane pierwsze dwa napisy oraz nadana wysokość 150 dla przynajmniej jednego obrazu
R.3.5	Ustawione marginesy: 5 dla pierwszych dwóch napisów, 10 dla przycisków oraz dla obrazów lewy, prawy i dolny 20, górny 0
R.3.6	Program obsługuje przynajmniej jedno zdarzenie kliknięcia przycisku w sposób właściwy dla danego środowiska programistycznego. Co najmniej jeden przycisk jest powiązany ze zdarzeniem

R.3.7	Program pobiera wartości z co najmniej jednej kontrolki w sposób właściwy dla danego środowiska programistycznego
R.3.8	Po wciśnięciu przycisku „Zatwierdź” pobierana jest liczba z pola edycyjnego, jeśli jest ona z przedziału od 1 do 12 to zmieniany jest napis poniżej na: „Numer prania: <numer>”, gdzie <numer> oznacza wartość podaną w polu edycyjnym (w uruchomionej aplikacji lub na zrzucie oraz <u>obowiązkowo</u> w kodzie)
R.3.9	Po wciśnięciu przycisku „Włącz”, zmienia się podpis przycisku na „Wyłącz” oraz treść napisu poniżej na: „Odkurzacz włączony”. Jeżeli kolejny raz zostanie wciśnięty przycisk, napisy wracają do stanu początkowego („Włącz” oraz „Odkurzacz wyłączony”) (w uruchomionej aplikacji lub na zrzucie oraz <u>obowiązkowo</u> w kodzie)
R.3.10	Aplikacja uruchamia się w emulatorze, co jest udokumentowane zrzutem ekranu, jej układ jest zgodny z obrazem 3 w arkuszu egzaminacyjnym
R.4	Rezultat 4: Dokumentacja aplikacji

<p>Uwaga: nagłówek z kryteriów 4.1 ÷ 4.5 musi być zgodny ze stanem faktycznym z kodu źródłowego, nawet jeżeli w kodzie są błędy logiczne (liczba pól, typy). Sprawdzany jest tylko jeden dowolny nagłówek. Zrzuty ekranu z kryteriów 4.6 i 4.7 muszą zawierać cały obszar ekranu z widocznym paskiem zadań. Dokumentacja z kryterium 4.8 zapisana jest w pliku egzamin</p>	
R.4.1	Dla metody klasy bazowej z aplikacji konsolowej zapisany nagłówek w postaci komentarza zgodny z Listingiem 1 z arkusza egzaminacyjnego (nie liczymy gwiazdek), komentarz może być wieloliniowy lub kilka jednoliniowych lub Docstrings (potrójny cudzysłów) - w tym przypadku opis znajduje się pod nazwą metody
R.4.2	W komentarzu podana: nazwa i opis działania metody
R.4.3	W komentarzu podana: nazwa i opis wszystkich parametrów metody
R.4.4	W przypadku gdy metoda zwraca wartość, podany w komentarzu opis tej wartości. W przeciwnym przypadku zapisany void, brak itp.
R.4.5	W komentarzu podany numer zdającego
R.4.6	Zapisany przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji konsolowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja
R.4.7	Zapisany przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji mobilnej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja
R.4.8	Dokumentacja zawiera wykorzystane oprogramowanie: nazwa systemu operacyjnego, środowiska programistycznych, emulator, języki programowania