

Nazwa  
kwalifikacji:  
Oznaczenie  
kwalifikacji:

## Projektowanie, programowanie i testowanie aplikacji

INF.04

Numer zadania: 01

Kod arkusza: INF.04-01-25.06-SG

Wersja arkusza: SG

Lp.	Elementy podlegające ocenie/kryteria oceny
R.1	<b>Rezultat 1: Implementacja, kompilacja, uruchomienie programu</b>
	<i>Uwaga: kryteria należy odnieść do aplikacji konsolowej, jeżeli ta nie istnieje, kryteria zastosować do aplikacji desktopowej. Kryteria dotyczą wyłącznie samodzielnie napisanego kodu. Wystarczy, że sprawdzaną cechę zastosowano dla większości przypadków w kodzie (90%)</i>
R.1.1	Kod źródłowy zapisany w sposób czytelny: instrukcje w osobnych liniach, stosowane spacje pomiędzy operatorami, konsekwentnie stosowana wybrana konwencja dla nawiasów klamrowych instrukcji blokowej
R.1.2	Kod zapisany z wcięciami dla zagnieżdżeń bloków
R.1.3	Użyte polskie lub angielskie nazewnictwo funkcji lub metod. Nazewnictwo jest znaczące
R.1.4	Użyte polskie lub angielskie nazewnictwo pól lub zmiennych. Nazewnictwo jest znaczące. Wyjątkami od reguły są zmienne bufor, tmp, iteratory pętli. Kryterium <b>nie jest</b> spełnione tylko wtedy, gdy zmienne nie istnieją lub ich nazwy nic nie znaczą, np. x, tab, tablica, foo, kolekcja
R.1.5	Zastosowane typy pasujące do problemu, np. tablica lub kolekcja liczb całkowitych, dane wczytywane jako typ całkowity
R.1.6	Podjęta próba uruchomienia, co jest udokumentowane zrzutem przedstawiającym uruchomiony program lub jego kompilację
R.1.7	Program podejmuje zrozumiałą komunikację z użytkownikiem: komunikaty są znaczące, np. "ile losowań?" itp.
R.2	<b>Rezultat 2: Aplikacja konsolowa</b>
	<i>Uwaga: kryteria 2.1 ÷ 2.4, 2.6 należy sprawdzić w pliku z kodem źródłowym 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 kryteria 2.5, 2.7, 2.8 nie są 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 co najmniej dwóch funkcji lub metod (mogą być niedokończone lub zawierać błędy). Po wyjściu z funkcji / metody losującej dane nie zostały utracone (w podejściu strukturalnym nie można stosować zmiennych globalnych)
R.2.2	Do przechowywania danych zastosowana tablica dwuwymiarowa lub inna kolekcja o dynamicznym rozmiarze dostosowanym do ilości danych
R.2.3	Do losowania liczb zastosowany generator liczb pseudolosowych, losowane są liczby z przedziału od [1, 49] (+/- 1 z każdej strony przedziału)
R.2.4	Program wypełnia tablicę lub kolekcję danymi, pętla obsługuje rzeczywistą liczbę losowań
R.2.5	Program wyświetla zawartość tablicy lub kolekcji za pomocą pętli. Wyświetlone wszystkie losowania (w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie)
R.2.6	Program implementuje losowanie przynajmniej jednego zestawu sześciu liczb bez powtórzeń
R.2.7	Program zlicza wystąpienia wszystkich liczb od 1 do 49 w losowaniach, są one wyświetlane na ekranie (w uruchomionej aplikacji lub na zrzucie i <b>obowiązkowo</b> w kodzie, dopuszcza się wyświetlenie tylko niezerowych wystąpień)
R.2.8	Program uruchamia się w konsoli, co jest udokumentowane zrzutem ekranu

<b>R.3</b>	<b>Rezultat 3: Aplikacja desktopowa</b>
	<p><i>Uwaga: Kryteria 3.1 ÷ 3.3 sprawdzić w pliku z kodem źródłowym programu, 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 3.4 ÷ 3.9 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	W oknie zdefiniowane wszystkie kontrolki (etykiety, suwaki, przycisk) oraz ich układ jest zgodny z obrazem 2 w arkuszu egzaminacyjnym (prostokąty mały i duży mogą być dowolną kontrolką, która daje efekt przedstawiony na obrazie 2)
R.3.2	<p>W stanie początkowym:</p> <ul style="list-style-type: none"> <li>- prostokąty są koloru białego, mały wyświetla wartość 255, 255, 255,</li> <li>- przynajmniej jeden suwak przyjmuje wartości od 0 do 255,</li> <li>- przynajmniej jeden suwak i odpowiadająca mu etykieta mają wartość 255</li> </ul>
R.3.3	Zdefiniowana metoda obsługująca kliknięcie przycisku
R.3.4	<p>Zmiana położenia przynajmniej jednego suwaka powoduje wyświetlenie aktualnej wartości suwaka po jego prawej stronie</p> <p><i>(w uruchomionej aplikacji lub na zrzucie i <u>obowiązkowo</u> w kodzie)</i></p>
R.3.5	<p>Zmiana położenia każdego z trzech suwaków powoduje zmianę koloru dużego prostokąta. Kolor odpowiada wartościom RGB ustawionym na suwakach</p> <p><i>(w uruchomionej aplikacji lub na zrzucie i <u>obowiązkowo</u> w kodzie)</i></p>
R.3.6	<p>Tylko po kliknięciu przycisku zmienia się kolor małego prostokąta. Kolor odpowiada wartościom RGB ustawionym na suwakach</p> <p><i>(w uruchomionej aplikacji lub na zrzucie i <u>obowiązkowo</u> w kodzie)</i></p>
R.3.7	Po kliknięciu przycisku zmienia się tekst w małym prostokącie, odpowiada on wartościom suwaków <i>(w uruchomionej aplikacji lub na zrzucie i <u>obowiązkowo</u> w kodzie)</i>
R.3.8	Program uruchamia się w oknie, co jest udokumentowane zrzutem ekranu <i>(wymagana przynajmniej jedna kontrolka w oknie)</i>
<b>R.4</b>	<b>Rezultat 4: Dokumentacja aplikacji</b>
	<p><i>Uwaga: komentarz 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 nagłówek.</i></p> <p><i>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</i></p>
R.4.1	Dla dowolnej funkcji lub metody zapisany nagłówek w postaci komentarza zgodnego z Listingiem 1 z arkusza egzaminacyjnego <i>(nie należy brać pod uwagę liczby gwiazdek)</i> , komentarz może być wieloliniowy lub kilka jednoliniowych, lub Docstrings <i>(potrójny cudzysłów)</i> - w tym przypadku opis znajduje się pod nazwą funkcji
R.4.2	W komentarzu umieszczona nazwa funkcji / metody oraz opisane jej działanie
R.4.3	W komentarzu umieszczone wszystkie argumenty funkcji / metody: nazwy i krótki opis lub "brak", gdy nie ma argumentów
R.4.4	W komentarzu umieszczona nazwa typu zwracanego i krótki opis co jest zwracane lub "brak", gdy funkcja nie zwraca wartości
R.4.5	W komentarzu umieszczony 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 desktopowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja
R.4.8	Dokumentacja zawiera: nazwę systemu operacyjnego, nazwy środowisk programistycznych, nazwy języków programowania <i>(stosowanych przez zdającego)</i>
R.4.9	Na płycie wszystkie wykonane projekty są spakowane oraz umieszczone pliki źródłowe i pliki wykonywalne <i>(jeśli istnieją)</i> na zewnątrz archiwów