

Nazwa
kwalifikacji:
Oznaczenie
kwalifikacji:

Projektowanie, programowanie i testowanie aplikacji

INF.04

Numer zadania: 01

Kod arkusza: INF.04-01-25.01-SG

Wersja arkusza: SG

| Lp. | Elementy podlegające ocenie/kryteria oceny |
|-------|---|
| R.1 | Rezultat 1: Implementacja, kompilacja, uruchomienie programu |
| | <i>Uwaga: kryteria należy odnieść do aplikacji konsolowej, jeżeli ta nie istnieje, zastosować do aplikacji webowej. Kryteria dotyczą wyłącznie samodzielnie napisanego kodu. Wystarczy, że sprawdzana cecha zastosowana dla większości (90%) przypadków w kodzie</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 znaczące polskie lub angielskie nazewnictwo metod |
| R.1.4 | Użyte polskie lub angielskie nazewnictwo zmiennych i obiektów. Nazewnictwo jest znaczące. Wyjątkami od reguły są zmienne: bufor, tmp, iteratory pętli. Kryterium nie jest spełnione tylko wtedy, gdy nazwy zmiennych nic nie znaczą, np.: x, foo (<i>Uwaga! klasa i pole tablicowe klasy może mieć nazwę ogólną np tab, tablica itp</i>) |
| R.1.5 | Zastosowane typy pól pasujące do problemu, np. tablica przechowuje liczby całkowite, rozmiar i wartość szukana są liczbami całkowitymi, średnia jest liczbą. W przypadku Python, typ wynika z przypisanych danych |
| R.1.6 | Podjęta próba skompilowania kodu, udokumentowana zrzutem przedstawiającym uruchomiony program lub jego kompilację |
| R.1.7 | Program podejmuje komunikację z użytkownikiem, wyświetlane komunikaty są znaczące |
| R.2 | Rezultat 2: Aplikacja konsolowa |
| | <i>Uwaga: kryteria 2.1 ÷ 2.5 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 2.6 ÷ 2.10 nie są spełnione. Jeżeli błędy występują w innych plikach, należy ocenić na podstawie kodu i zrzutów ekranu. W przypadku języka Python argument self nie jest wliczany do liczby parametrów</i> |
| R.2.1 | W programie zdefiniowana klasa z dwoma polami private, jedno jest typu tablicowego, drugie typu całkowitego, w programie głównym jest powołany obiekt klasy z liczbą > 20 (<i>W języku Python, dla pola typu private, dwa podkreślniki przed nazwą</i>) |
| R.2.2 | Konstruktor klasy ma jeden parametr i wypełnia tablicę wartościami losowymi od 1 do 1000 (<i>tolerancja +/-1</i>) oraz ustawia pole liczby elementów tablicy wartością parametru konstruktora |
| R.2.3 | Metoda szukająca pobiera argument typu liczbowego. Metody są typu: wyświetlająca - bez typu (np. void) wyszukująca - typ całkowity wyświetlająca nieparzyste - typ całkowity licząca średnią - typ liczbowy (wystarczy, że zastosowane dla trzech metod) |
| R.2.4 | Przynajmniej jedna pętla działająca na tablicy jest ograniczona do liczby wypełnionych elementów w tablicy |
| R.2.5 | W metodzie wyszukiwającej, gdy odnaleziono pierwsze wystąpienie wartości szukanej w tablicy, działanie pętli jest zakończone lub została zastosowana wbudowana metoda lub funkcja |

| | |
|------------|--|
| R.2.6 | Program uruchamia się w konsoli, co jest udokumentowane zrzutem ekranu |
| R.2.7 | Program wyświetla wszystkie elementy tablicy wg. wzoru "<index_tablicy>: <wartość>" |
| R.2.8 | Program wyszukuje liczbę w tablicy i wyświetla jej indeks, gdy nie odnaleziono liczby nic nie wyświetla |
| R.2.9 | Program wyświetla na ekranie liczby nieparzyste oraz liczbę ich wystąpień |
| R.2.10 | Program liczy średnią arytmetyczną (suma / rozmiar) oraz wyświetla ją na ekranie |
| R.3 | Rezultat 3: Aplikacja webowa |
| | <p><i>Uwaga: Kryteria 3.1 ÷ 3.7 sprawdzić w kodzie źródłowym, sprawdzane elementy muszą być zapisane zgodnie ze składnią.</i></p> <p><i>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 zrzutach ekranu. Cały rezultat nie jest spełniony, gdy zdający zapisał kod bez użycia Angular lub React, stosował metody DOM w tym getElementById i inne podobne, przypisywał bezpośrednio metody do zdarzeń np. onclick="fun()" zamiast (click)="fun()" czy onClick={fun}</i></p> |
| R.3.1 | Zapisany w aplikacji jeden komponent, który zawiera nagłówek h1, trzy pola switch lub checkbox zaznaczone, przynajmniej jeden obraz, nagłówek h4 i przycisk "Pobierz" oraz tablicę obiektów z pliku <i>dane.txt</i> |
| R.3.2 | Do projektu dołączona biblioteka Bootstrap oraz sformatowany obraz: <code>margin: 5px;</code> oraz zaokrąglone rogi (<i>border-radius lub klasa rounded</i>) |
| R.3.3 | <p>Przycisk ma przypisaną klasę <code>btn btn-success</code> oraz przynajmniej jedno pole switch zdefiniowane za pomocą biblioteki Bootstrap jako blok formatowany stylami <code>form-check form-switch</code> (<i>form-check-inline opcjonalnie</i>) i złożony z kontrolki input i przypisanej do niej etykiety label, np.:</p> <pre><div class="form-check form-switch"> <input class="form-check-input" type="checkbox" id="chk3" ...> <label class="form-check-label" for="chk3">Samochody</label> </div></pre> <p>W React.js zastosowane <code>className</code> zamiast <code>class</code> oraz domknięte wszystkie znaczniki</p> |
| R.3.4 | <p>Do generowania obrazów zastosowana pętla działająca na elementach galerii np. Angular: <code>*ngFor = "let zdjecie of galeria"</code></p> <p>React.js: <code>{this.state.galeria.map(zdjecie => ... (lub bez this.state) }</code> oraz pętla generuje obraz i nagłówek h4 z odwołaniem do pól obiektu np.</p> <pre> <h4>Pobrań: {{ zdjecie.downloads }}</h4></pre> <p>W React.js pojedyncze nawiasy klamrowe (<i>uwaga! może poprawnie działać również bez nazwy katalogu assets</i>)</p> |
| R.3.5 | <p>Do sprawdzenia czy switch jest zaznaczony lub czy kategoria obrazu zgadza się z zaznaczeniem zastosowany warunek np.:</p> <p>Angular: <code>*ngIf = "kwiaty == true", *ngIf = "zdjecie.category == 1"</code></p> <p>React.js: funkcja renderująca, w której znajduje się warunek i instrukcja return, która odpowiednio zwraca kod wyświetlający obrazek</p> |

| | |
|------------|--|
| R.3.6 | <p>Zapisane zdarzenie kliknięcia przycisku i obsłużone w kodzie np. Angular: <code><button ... (click)="onPobierzClick(i)">Pobierz</button></code> <code>onPobierzClick(i: any):void ...</code> zmiennej <i>i</i> z przykładu jest indeksem w pętli (<code>let i=index</code>)</p> <p>React.js: <code>onClick={this.OnPobierzClick(this.state.i)}</code> <code>OnPobierzClick() {...}</code></p> |
| R.3.7 | <p>W kodzie zdefiniowana przynajmniej jedna funkcja (lub operacja), która jest uruchamiana po zmianie stanu switch np. <code>(change)="flowersChanged()</code> lub <code>onChange={this.flowersChanged() }</code></p> |
| R.3.8 | <p>Aplikacja jest interpretowana bez błędów. W stanie początkowym są zaznaczone wszystkie pola switch i wyświetlone wszystkie obrazy co jest udokumentowane zrzutem ekranu</p> |
| R.3.9 | <p>Zdjęcia wyświetlają się w zależności od stanu pól switch (<i>w uruchomionej aplikacji lub na zrzucie oraz obowiązkowo w kodzie</i>)</p> |
| R.3.10 | <p>Po wciśnięciu przycisku zwiększany jest o jeden licznik pobrań, zmiany są widoczne na stronie, są modyfikowane w tablicy dla odpowiedniego obiektu (<i>w uruchomionej aplikacji lub na zrzucie oraz obowiązkowo w kodzie</i>)</p> |
| R.4 | Rezultat 4: Dokumentacja aplikacji |
| | <p><i>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</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 | <p>Dla metody z aplikacji konsolowej zapisany nagłówek w postaci komentarza zgodny z Listingiem 1 z arkusza egzaminacyjnego (nie należy brać pod uwagę liczby gwiazdek), komentarz może być wieloliniowy lub kilka jednoliniowy lub Docstrings (potrójny cudzysłów)</p> |
| R.4.2 | <p>W komentarzu podana nazwa metody oraz opisane jej działanie</p> |
| R.4.3 | <p>W komentarzu podane wszystkie argumenty metody: nazwy i krótki opis lub zapisane "brak"</p> |
| R.4.4 | <p>W komentarzu podana nazwa typu zwracanego i krótki opis co jest zwracane lub zapisane "brak"</p> |
| R.4.5 | <p>W komentarzu podany numer zdającego</p> |
| R.4.6 | <p>Zapisany przynajmniej jeden zrzut ekranu z uruchomienia lub kompilacji aplikacji konsolowej, na zrzucie widoczne jest środowisko, w którym powstała aplikacja</p> |
| R.4.7 | <p>Zapisany przynajmniej jeden zrzut ekranu z uruchomienia aplikacji webowej lub z błędami interpretacji, na zrzucie widoczne jest środowisko, w którym powstała aplikacja</p> |
| R.4.8 | <p>Dokumentacja zawiera wykorzystane oprogramowanie: nazwa systemu operacyjnego, nazwy środowisk programistycznych, nazwy języków programowania</p> |