**5 JS Zadania Złożone struktury danych**

**Zad 1.**

Wymień poznane w szkoleniu struktury danych i krótko opisz różnice między każdym z nich.

**Zad 2.**

Mając do dyspozycji poniższą listę imion:

let names = [‘Jan’, ‘Grzegorz’, ‘Krzysztof’, ‘Maciej’, ‘Joanna’, ‘Izabela’, ‘Magdalena’, ‘Kinga’, ‘Kacper’]

, odwołaj się kolejno do:

* pierwszego,
* ostatniego,
* przedostatniego,
* środkowego

elementu.

Następnie dodaj na koniec takiej listy dwa nowe imiona, a na początek jedno. Ponów operacje. Czy otrzymałeś różne wyniki niż poprzednio? Jak zaprogramować ww. funkcjonalność tak, aby była niezależna od ilości elementów w liście.

**Zad 3.**

Aby przećwiczyć możliwe do wywołania metody na tablicy, wykonaj poniższe kroki (realizację każdego z zestawów umieść w oddzielnych funkcjach):

Zestaw I

* Stwórz pustą tablicę
* Dodaj do niej 5 nowych elementów
* Odczytaj długość takiej tablicy
* Umieść nowy element na początku tablicy
* Odczytaj pierwszy, środkowy oraz ostatni element na takiej liście

Zestaw II 🡪 **DOPYTAJ**

* Zadeklaruj pustą tablicę i umieść w niej 5 dowolnych nazw firm IT, które poda użytkownik. Użyj pętli.
* Wyświetl tylko te firmy, które nie mają w nazwie litery ‘o’
* Sprawdź czy użytkownik podał firmę o nazwie “Devs-Mentoring.pl”. Na podstawie tego wyświetl odpowiedni komunikat.
* Usuń drugą nazwę firmy z listy
* Wyświetl tylko te nazwy, których długość przekracza 8 znaków
* Skopiuj tablicę do companies\_copy
* Odwróć oryginalną listę, korzystając z funkcji reverse()

Zestaw III

* Stwórz listę 10-ciu dowolnie wygenerowanych kodów o długości 5 lub 6 (długość ma być losowana)
* Wyświetl tylko te wyrazy, których długość jest parzysta

**Zad 4.**

Mając do dyspozycji poniższą tablicę:

const ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

* Posortuj listę i znajdź wartość minimalną oraz maksymalną
* Oblicz średnią wieku
* Oblicz przedział wieków (różnica między max a min)

**Zad 5.**

Zaprojektuj mapę, w której przechowywać będziesz pary imię : wiek\_osoby. Pobierz z niej:

* Tylko klucze
* Tylko wartości
* Pary wszystkich elementów (kluczy i wartości)

**Zad 6.**

Stwórz mapę, która będzie przechowywała 5 par: kluczami będą nazwy użytkowników (*Dagger, Dev, JS-coder, Python-Coder, Mike*), a wartości do nich przypisane - stringami reprezentującymi dowolne daty rejestracji w formacie DD-MM-YYYY.

* Po inicjalizacji, dodaj do takiej listy 2 nowych użytkowników
* Usuń dowolnego usera
* Wyświetl wszystkie daty przechowywane w mapie, ale w formacie DD/MM/YYYY (nie DD-MM-YYYY)
* Usuń z mapy użytkownika o nickname JS-coder

**Zad 7.**

Mając do dyspozycji poniższy zestaw informacji (jest to informacja o językach i ilości kursantów przypisanego do każdego z nich; zestaw ten umieść w Mapie)...

["French:", 45],

["Arabic: ", 25],

["Spanish:", 24],

["Russian:", 9],

["Portuguese:", 9],

["Dutch:", '8'],

["German:", 7],

["Chinese:", 5],

["Swahili:", 4],

["Serbian:", 4],

["English:", 91],

… odczytaj nazwę państwa, z którym skojarzona jest największa ilość uczestników. Wykorzystaj forEach().

**Zad 7.**

Wyszukaj, co znaczy, że elementy w zbiorze są hashowane.

**Zad 8.**

Odszukaj, jaką złożonością wyszukiwania charakteryzuje się mapa/zbiór/tablica. Czym jest **notacja dużego O**?

**Zad 9.**

Utwórz zbiór uniqueNumbers i dodaj do niego następujące wartości:

* 1231112
* 1231113
* 1231114
* 1221112
* 1231119
* 1231112
* 1231114

Odczytaj jego rozmiar. Dlaczego jest on równy 5, a nie 7 (czyli ilości liczb, które umieściliśmy w zbiorze)?

**Zad 10.**

Wykonaj poniższe operacje:

* Stwórz pusty zbiór.
* Stwórz zbiór składający się z liczb od 0 do 10 (użyj pętli)
* Usuń element o wartości 5
* Wyczyść zbiór
* Stwórz drugi zbiór - składający się z 5 dowolnych nazw państw
* Stwórz mapę składającą się z kluczy - państw oraz wartości, czyli ilości liter każdego z państw

**Zad 11.**

Wygeneruj tablicę 1000 trójelementowych napisów składających się tylko z liter: ‘a’, ‘b’, ‘c’, ‘d’, ‘e’. Odczytaj rozmiar takiej tablicy. Następnie usuń z niej duplikaty i wyświetl wynik. Jak zmienił się rozmiar?