



Podstawowe struktury danych: tablica

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Podstawowe struktury danych: tablica

Źródło: Kelly Sikkema, domena publiczna.

Tablica jest jedną z najbardziej podstawowych struktur danych. Pozwala ona na łatwe przechowywanie i modyfikację dużej liczby zmiennych, dzięki czemu jest często wykorzystywana. Dodatkowo za jej pomocą jesteśmy w stanie zaimplementować bardziej złożone obiekty, takie jak stos, czy chociażby reprezentować połączenia między wierzchołkami w grafie.

Więcej informacji o podstawowych strukturach danych znajdziesz w e-materiałach:

- [Podstawowe struktury danych,](#)
- [Podstawowe struktury danych: rekord,](#)
- [Podstawowe struktury danych: lista,](#)
- [Podstawowe struktury danych: stos i kolejka.](#)

Twoje cele

- Dowiesz się, czym jest tablica.
- Poznasz podstawowe operacje na tablicach.
- Przeanalizujesz przykładowe zastosowania tablic.

Przeczytaj

Czym jest tablica?

Podstawowym założeniem, na jakim opiera się koncepcja tablicy, jest możliwość przechowywania wielu wartości **tego samego typu**. Przypuśćmy, że chcemy w naszym programie przechowywać wyniki pomiarów temperatury powietrza z ostatnich dwóch tygodni. Pierwsze, co w tej sytuacji przychodzi na myśl, to po prostu zdefiniowanie 14 osobnych zmiennych. A gdybyśmy chcieli ulepszyć nasz program i przechowywać wyniki pobrane o trzech różnych porach dnia? Wtedy potrzebowalibyśmy już 42 zmiennych.

W ten sposób nasz program stawałby się coraz bardziej zagmatwany, a definiowanie kolejnych zmiennych byłoby niepotrzebnie pracochłonne. Natomiast gdybyśmy użyli tablicy, byłibyśmy w stanie zadeklarować dowolną liczbę zmiennych tego samego typu przy pomocy jednej instrukcji. Dodatkowo wykorzystując **indeksowanie**, moglibyśmy wygodnie je modyfikować.

Aby lepiej zrozumieć tę koncepcję, przeanalizujmy poniższy pseudokod. Załóżmy przy tym, że pierwszy element tablicy ma indeks 0.

```
1 tab[5] ← {}           // zadeklarowanie 5-elementowej tablicy
2
3 tab[0] ← 2
4 tab[1] ← 3
5 tab[2] ← tab[0] + tab[1]
6 tab[3] ← 2 * tab[0]
7 tab[4] ← tab[1]
8
9 i ← 0
10
11 Dopóki i < 5 wykonuj
12     wypisz(tab[i])    // wypisze 2 3 5 4 3
```

Tablica jest zazwyczaj strukturą statyczną i jej rozmiar nie zmienia się w trakcie działania programu, jednak istnieją również warianty dynamiczne. Przy deklaracji statycznej tablicy należy określić, ile elementów ma ona przechowywać, czyli podać jej rozmiar. Aby odwołać się do konkretnego elementu, wykorzystujemy indeksowanie, czyli odwołanie postaci `nazwa_tablicy[indeks]`. Dzięki temu jesteśmy w stanie w wygodny sposób modyfikować i wykorzystywać zawartość tablicy.

ZAWARTOŚĆ:

2	3	5	4	3
---	---	---	---	---

INDEKS:

0 1 2 3 4

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Ważne!

Niestety, tablice nie są w pełni zunifikowane – w zależności od wykorzystywanego języka programowania, do pierwszej komórki tablicy odwołujemy się, stosując indeks 0 bądź 1. Co za tym idzie, podczas pisania pseudokodu warto zaznaczyć, z jakiego typu indeksowania korzystamy.

Tablica wielowymiarowa

Założmy, że chcemy w naszym programie zaimplementować symulację gry w szachy. Plansza do gry jest kwadratem zbudowanym z 64 pól, po których mogą poruszać się figury. Do przechowywania informacji o rodzaju figury i jej pozycji na szachownicy moglibyśmy wykorzystywać więc 64-elementową tablicę jednowymiarową, lecz indeksowanie byłoby tu dosyć problematyczne.

O wiele lepszym pomysłem w takiej sytuacji będzie użycie tablicy dwuwymiarowej. Jej zadeklarowanie jest równie proste, co w przypadku jednego wymiaru. Spójrzmy na poniższy przykład:

```
1 tab[2][3] ← {} // zadeklarowanie tablicy rozmiaru 2x3
2 i ← 0
3
4 Dopóki i < 2 wykonuj
5     j ← 0
6
7     Dopóki j < 3 wykonuj
8         tab[i][j] ← i + j
9         wypisz(tab[i][j])
10        j ← j + 1
11
12    wypisz("\n")
13    i ← i + 1
14
```

```
15 // Nasz program wypisze:
16 // 0 1 2
17 // 1 2 3
```

Aby odwołać się do konkretnej wartości, wykorzystujemy następujący zapis: `nazwa_tablicy[indeks_wiersza][indeks_kolumny]`.

		indeksy kolumn							
		0	1	2	3	4	5	6	7
indeksy wierszy	0								
	1								
	2								
	3								
	4								
	5								
	6								

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

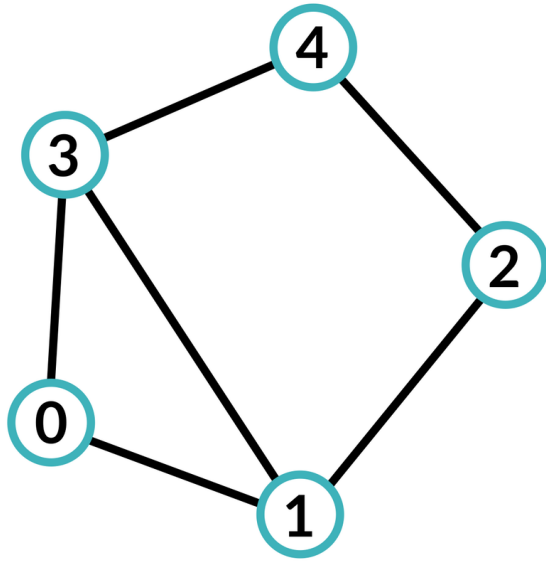
Ciekawostka

Aby zadeklarować tablicę o dowolnej liczbie wymiarów wystarczy dodać odpowiednią liczbę nawiasów. Np. za zadeklarowanie tablicy czterowymiarowej w pseudokodzie odpowiadałaby instrukcja: `tab[3][3][4][4] ← {}`.

Przykładowe zastosowania tablic

Tablice są bardzo uniwersalną strukturą danych, dzięki czemu mogą być wykorzystywane na różne sposoby. Jednym z takich zastosowań jest reprezentacja tekstu w postaci tablicy znaków. Można też za pomocą tablicy dwuwymiarowej reprezentować połączenia między wierzchołkami grafu. Nazywamy ją [macierzą sąsiedztwa](#).

RYSUNEK GRAFU



MACIERZ SĄSIEDZTWA

	0	1	2	3	4
0	0	1	0	1	0
1	1	0	1	1	0
2	0	1	0	0	1
3	1	1	0	0	1
4	0	0	1	1	0

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Poza tym tablice nie ograniczają się do przechowywania podstawowych typów danych takich jak liczby całkowite lub zmiennoprzecinkowe. Możemy również utworzyć tablicę zawierającą struktury albo chociażby listy. Mamy dużą dowolność.

Słownik

macierz sąsiedztwa grafu

dwuwymiarowa tablica kwadratowa, w której wartość przechowywana w wierszu i oraz kolumnie j oznacza liczbę połączeń w grafie wychodzących z wierzchołka i do wierzchołka j

Prezentacja multimedialna

Polecenie 1

Przeanalizuj poniższą prezentację, a następnie wykonaj polecenie.

ZAWARTOŚĆ:	2	5	7	1
INDEKS:	0	1	2	3

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablicą nazywamy zbiór uporządkowanych danych tego samego typu, w którym poszczególne elementy dostępne są za pomocą indeksów.

W zależności od języka programowania indeks pierwszego elementu wynosi 0 lub 1.

```
1 liczby[4] ← {2, 5, 7, 1}
2 liczby[0] ← liczby[1] +
  liczby[2]
3 wypisz(liczby[0])
4 // program wypisze: 12
```

		indeksy kolumn	
		0	1
indeksy wierszy	0	1	2
	1	2	2
	2	9	3

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Dodatkowo tablice jednowymiarowe mogą przechowywać inne tablice, dzięki czemu uzyskuje się tablice wielowymiarowe. Wtedy poszczególne elementy są adresowane poprzez ciąg indeksów.

```

1 liczby[3][2] ← {{1, 2},
  {2, 2}, {9, 3}}
2 suma ← (liczby[0][0] *
  liczby[0][1]) + liczby[2]
  [0]
3 wypisz(suma)
4 // program wypisze: 11

```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablice są bardzo prostymi i uniwersalnymi strukturami. Ich głównym zadaniem jest reprezentowanie pewnego zbioru danych tego samego typu. Mogą zatem być wykorzystywane na wiele różnych sposobów.

3	4	1	2	6	5
---	---	---	---	---	---



1	2	3	4	5	6
---	---	---	---	---	---

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablice jednowymiarowe służą najczęściej do przechowywania, modyfikacji i reprezentacji danych. Może to być np. zestawienie liczb, które chcemy sortować, czy lista naszych wydatków w ciągu ostatniego miesiąca.

```
1 tab[6] ← {3, 4, 1, 2, 6,
2      5}
3 sortuj_rosnaco(tab)
3 wypisz(tab[0], tab[5])
4 // program wypisze: 1, 6
```

A	l	a		m	a		k	o	t	a
0	1	2	3	4	5	6	7	8	9	10

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablice mogą przechowywać m.in.: liczby całkowite, liczby zmiennoprzecinkowe, wartości logiczne oraz znaki.

Dzięki możliwości przechowywania znaków tablice są idealnym narzędziem do reprezentacji wyrazów i zdań.

```

1 slowo[11] ← "Ala ma kota"
2 slowo[1] ← "g"
3 slowo[10] ← "y"
4 wypisz(slowo)
5 // program wypisze: Aga ma
  koty

```

6

$$M = \begin{bmatrix} 1 & 4 \\ 8 & 6 \\ -3 & 3 \end{bmatrix} \rightarrow \begin{array}{|c|c|} \hline 1 & 4 \\ \hline 8 & 6 \\ \hline -3 & 3 \\ \hline \end{array}$$

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablice wielowymiarowe także mają wiele przydatnych zastosowań. W obliczeniach matematycznych mogą służyć za reprezentację komputerową macierzy. Są wykorzystywane m.in. do wyznaczania rozwiązań równań liniowych czy przekształceń liniowych obiektów na płaszczyźnie.

		○						0	0	2	0	0	0
		●	●	○				0	0	1	1	2	0
	○	●	●					0	2	1	1	0	0
	●	○						0	1	2	0	0	0
			●					0	0	0	1	0	0
								0	0	0	0	0	0

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

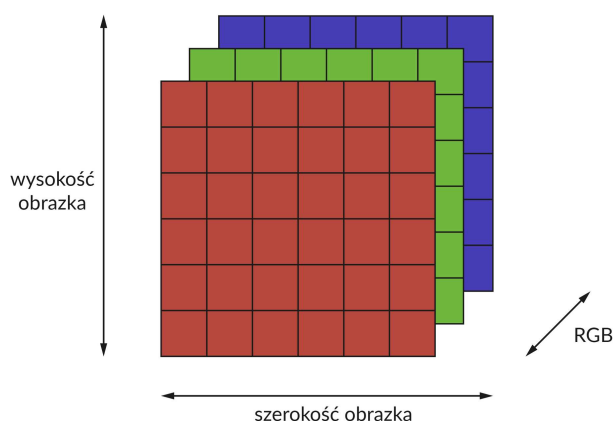
Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

7

Plansza do gry w Go może być reprezentowana przez tablicę dwuwymiarową. Jeżeli komórka w tablicy zawiera liczbę 0, to oznacza, że żaden gracz nie postawił tam swojego kamienia. Kamienie czarne mogą być reprezentowane przez liczbę 1, natomiast białe przez 2.

8



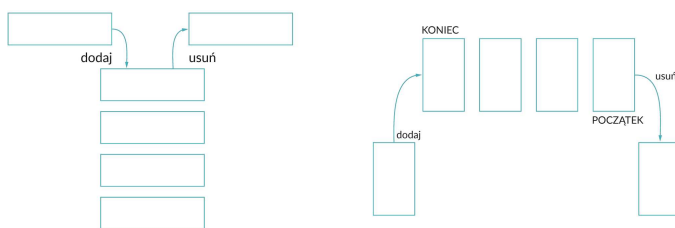
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Tablice trójwymiarowe idealnie nadają się do reprezentacji komputerowej zdjęć i obrazków. Składają się one z pikseli, które bardzo często zapisywane są w pamięci komputera jako trójka liczb reprezentująca ich barwę w modelu RGB.

```
1 obrazek[50][50][3] ← {}  
2  
3 // ustawmy pierwszy piksel  
  obrazka na kolor  
  pomarańczowy  
4 // ma on RGB : (255, 165,  
  0)  
5 obrazek[0][0][0] ← 255  
6 obrazek[0][0][1] ← 165  
7 obrazek[0][0][2] ← 0
```

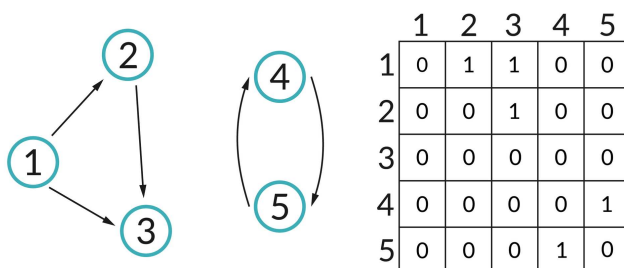


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Dzięki swojej uniwersalności i prostocie tablice jednowymiarowe mogą być także wykorzystywane do implementacji bardziej złożonych i zaawansowanych abstrakcyjnych struktur danych, takich jak stos czy kolejka.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HRyIBlj>

Kolejnym zastosowaniem tablic jest reprezentacja połączeń między wierzchołkami w grafie. Możemy przyjąć, że jeżeli w komórce znajdującej się w wierszu i oraz kolumnie j znajduje się liczba 1, to istnieje połączenie wychodzące z wierzchołka o numerze i do wierzchołka j . Brak połączenia może być reprezentowany przez liczbę 0.

Polecenie 2

Stwórz notatkę na temat innych przykładów zastosowania tablic.

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w prezentacji multimedialnej.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, jaki indeks ma pierwsza komórka jednowymiarowej tablicy.

☐ Zależy to od wykorzystywanego języka programowania.

☐ 0

☐ 1

Ćwiczenie 2



Zaznacz poprawną definicję macierzy sąsiedztwa grafu.

☐ Dwuwymiarowa tablica kwadratowa, w której wartość przechowywana w wierszu i oraz kolumnie j oznacza najmniejszą wagę spośród wszystkich krawędzi w grafie wychodzących z wierzchołka i do wierzchołka j .

☐ Dwuwymiarowa tablica kwadratowa, w której wartość przechowywana w wierszu i oraz kolumnie j oznacza największą wagę spośród wszystkich krawędzi w grafie wychodzących z wierzchołka i do wierzchołka j .

☐ Dwuwymiarowa tablica kwadratowa, w której wartość przechowywana w wierszu i oraz kolumnie j oznacza liczbę krawędzi w grafie wychodzących z wierzchołka i do wierzchołka j .

☐ Dwuwymiarowa tablica kwadratowa, w której wartość przechowywana w wierszu i oraz kolumnie j oznacza sumaryczną wagę krawędzi w grafie wychodzących z wierzchołka i do wierzchołka j ,

Ćwiczenie 3



Wskaż, czy możemy deklarować tablice o dowolnej liczbie wymiarów.

- ☐ Nie, możemy deklarować tablice maksymalnie czterowymiarowe.
- ☐ Nie, możemy deklarować tablice maksymalnie dwuwymiarowe.
- ☐ Tak, możemy.

Ćwiczenie 4



Dokończ zdanie.

Tablice służą do przechowywania danych...

- ☐ różnego typu.
- ☐ uporządkowanych hierarchicznie.
- ☐ uporządkowanych niemalejąco.
- ☐ tego samego typu.

Ćwiczenie 5



Zaznacz poprawną odpowiedź.

Przy deklaracji statycznej tablicy należy określić, ile elementów ma ona przechowywać.

- ☐ Fałsz
- ☐ Prawda

Ćwiczenie 6



Zaznacz wszystkie poprawne odwołania do zawartości tablicy, zakładając, że piszesz pseudokod algorytmu, który wykorzystuje tablicę Tab o rozmiarze 4×4 .

☐ Tab[1][1][1]

☐ Tab[3][2]

☐ Tab[8][2]

☐ Tab[1][1]

☐ Tab[2;3]

Ćwiczenie 7



Uzupełnij poniższy pseudokod algorytmu, który wypisuje na wyjście zawartość dwuwymiarowej tablicy o rozmiarze 4×4 .

Dopóki $i < 4$ wykonuj

Dopóki $j < 5$ wykonuj

$i \leftarrow i + 1$

$j \leftarrow j + 1$

wypisz("\n")

$j \leftarrow 0$

wypisz(tab[i][j])

$i \leftarrow 0$

Ćwiczenie 8



Wstaw elementy do właściwych grup.

słownictwo związane z tablicami

słownictwo niezwiązane z tablicami

wierzchołek

indeks

waga

szczyt

liście

komórka

krawędź

korzeń

Dla nauczyciela

Autor: Zespół autorski [Contentplus.pl](https://contentplus.pl) sp. z o.o.

Przedmiot: Informatyka

Temat: Podstawowe struktury danych: tablica

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy i rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

- I. Rozumienie, analizowanie i rozwiązywanie problemów.

Zakres podstawowy. Uczeń:

- 1) planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

- 2) do realizacji rozwiązania problemu dobiera odpowiednią metodę lub technikę algorytmiczną i struktury danych;

- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Dowiesz się, czym jest tablica.
- Poznasz podstawowe operacje na tablicach.
- Przeanalizujesz przykładowe zastosowania tablic.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiałach;
- tablica interaktywna/tablica, pisak/kreda.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Podstawowe struktury danych: tablica”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla uczniom temat, wskazuje cele zajęć oraz ustala z uczestnikami zajęć kryteria sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie przystępują do cichego czytania tekstu zawartego w sekcji „Przeczytaj”, jeśli nauczyciel – na podstawie raportu na platformie – uważa, że przygotowanie uczniów jest wystarczające, może pominąć tę czynność.
2. **Praca z multimedialnym.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie analizują prezentację, a następnie samodzielnie zastanawiają się nad innymi przykładami zastosowania tablic.
3. **Ćwiczenie umiejętności.** Uczniowie realizują indywidualnie ćwiczenia nr 1-6, po ich wykonaniu porównują otrzymane wyniki z inną osobą.
4. Uczniowie wykonują w parach kolejne ćwiczenia i po rozwiązaniu każdego z nich porównują swoje odpowiedzi z inną parą.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat i cele lekcji zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji następuje omówienie ewentualnych problemów z rozwiązaniem ćwiczeń z sekcji „Sprawdź się”.
2. Nauczyciel prosi uczniów o podsumowanie zgromadzonej wiedzy.

Praca domowa:

1. Uczniowie opracowują FAQ (minimum 3 pytania i odpowiedzi) do tematu lekcji („Podstawowe struktury danych: tablica”).
2. Uczniowie wykonują ćwiczenia 6-8 z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Treści w sekcji „Prezentacja multimedialna” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.