



Podstawowe struktury danych: rekord

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



Podstawowe struktury danych: rekord

Źródło: Kelly Sikkema, domena publiczna.

Analizując zbiór zainteresowań mieszkańców danego miasta, musimy wziąć pod uwagę wiele czynników, takich jak ich wiek, płeć czy stan majątkowy. Dzięki wykorzystaniu **rekordu**, jesteśmy w stanie w łatwy sposób operować danymi, zachowując ich wewnętrzną ustaloną strukturę. Stanowią one wtedy pewną logiczną całość, na której możemy pracować.

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

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

Twoje cele

- Dowiesz się, czym jest rekord.
- Poznasz podstawowe pojęcia związane z rekordem.
- Przeanalizujesz przykładowe zastosowania rekordów.

Przeczytaj

Rekord a struktura

Rekord pozwala na przechowywanie danych z zachowaniem pewnej abstrakcyjnej struktury. Rekord składa się z **różnych typów danych** połączonych w całość. Przykładowo, wyobraźmy sobie, że jesteśmy właścicielami firmy i chcemy napisać program, który ułatwi nam zarządzanie pracownikami. Program będzie działał, wykorzystując dane osobowe, takie jak imię, nazwisko, PESEL czy adres zamieszkania.

Moglibyśmy wszystkie potrzebne informacje przechowywać w oddzielnych tablicach. Wtedy w *i*-tej komórce każdej tablicy znajdowałyby się dane *i*-tego pracownika. Jednak taki sposób przechowywania danych jest niewygodny i mało intuicyjny. Skorzystajmy zatem z **rekordu**, często nazywanego również **strukturą**.

Ważne!

Struktura i rekord koncepcyjnie odpowiadają temu samemu modelowi danych. Przyjęło się jednak, że słowo *rekord* jest powiązane głównie z bazami danych, natomiast słowo *struktura* - z programowaniem.

Przeanalizujmy poniższy pseudokod:

```
1 Struktura Pracownik {
2     imie           // 1. składowa
3     nazwisko       // 2. składowa
4     pesel          // 3. składowa
5     wynagrodzenie  // 4. składowa
6 }
7
8 Pracownik p
9 p.imie ← "Jan"
10 p.nazwisko ← "Kowalski"
11 p.pesel ← "99122819456"
12 p.wynagrodzenie ← 5600
13
14 wypisz(p.wynagrodzenie * 0.5) // wypisze 2800
```

Definiując strukturę, musimy na samym początku zaznaczyć, jakie będą jej składowe. Każda z nich powinna mieć unikalną nazwę, za pomocą której będziemy mogli się odwoływać do jej konkretnej zawartości. Mówimy wtedy, że składowe są **etykietowalne**.

Firma może zatrudniać bardzo dużą liczbę pracowników, dlatego deklarowanie każdego z nich oddzielnie nie jest dobrym pomysłem. Możemy jednak utworzyć tablicę stuelementową o typie struktury Pracownik. Przeanalizujmy zatem jeszcze jeden pseudokod:

```
1 Pracownik pracownicy[100] = {}
2
3 pracownicy[0].wynagrodzenie = 10000
4 pracownicy[1] = pracownicy[0]
5
6 wypisz(pracownicy[0].wynagrodzenie + pracownicy[1].wynagrodzenie)
7 // wypisze 20000
```

Dzięki takiemu rozwiązaniu możemy w łatwy sposób utworzyć dużą liczbę zmiennych o typie wymyślonej struktury. Jest to bardzo wygodny i czytelny sposób pracy nad danymi, które posiadają pewną wewnętrzną abstrakcyjną strukturę.

Zagnieżdżanie struktur i tablice jako składowe

Struktury są bardzo elastyczne, jeśli chodzi o możliwe typy ich składowych. Mogą nimi być nawet tablice czy choćby inne struktury. Dzięki temu możemy tworzyć bardziej zaawansowane obiekty, które ułatwią nam pracę nad danymi.

Przeanalizujmy przykład takiej struktury:

```
1 Struktura Dane_osobowe {
2     imie
3     nazwisko
4     pesel
5 }
6
7 Struktura Pracownik {
8     Dane_osobowe dane // struktura jako składowa
9     godziny_pracy[50] // tablica jako składowa
10    wynagrodzenie
11 }
12
13 Pracownik p
14 p.dane.imie ← "Jan"
15 p.godziny_pracy[2] ← 8
```

Aby dostać się do konkretnej składowej zagnieźdzonej struktury, najpierw musieliśmy sprecyzować, że odwołujemy się do pola `dane` o typie `Dane_osobowe`, a następnie do składowej `imie`. Osiągnęliśmy to za pomocą instrukcji `p.dane.imie <-- "Jan"`.

Za to w przypadku tablicy musieliśmy podać konkretny indeks komórki. W powyższym pseudokodzie skorzystaliśmy z tablicy statycznej, zatem w definicji struktury musimy podać jej rozmiar. Dzięki temu każda zmienna o typie `Pracownik` będzie zawierać 50-elementową tablicę.

Przykładowe zastosowania

Struktura jest uniwersalna i może być wykorzystywana do różnych celów. Tak jak w zaprezentowanym wcześniej przykładzie, nadaje się ona do pracy na obiektach mających pewne etykietowalne własności.

Zastanów się, jakie inne zastosowania może mieć struktura.

Inne zastosowania

Dzięki temu możemy w wygodny sposób reprezentować pracowników firmy, sprzedawane samochody czy chociażby zawodników ulubionej drużyny sportowej.

Struktury mogą być także wykorzystywane do implementacji bardziej zaawansowanych struktur danych, takich jak lista czy drzewo. W liście dwukierunkowej każdy element przechowuje swoje wartości oraz wskaźnik do poprzednika i następnika. Wykorzystując strukturę, jesteśmy w stanie zaimplementować to w dosyć wygodny sposób.

```
1 Struktura Element_listy_dwukierunkowej {
2     wartosc_1
3     wartosc_2
4     poprzednik
5     nastepnik
6 }
```

Natomiast w przypadku **drzewa binarnego** każdy węzeł przechowuje swoją wartość, wskaźnik do ojca oraz synów – lewego i prawego. Tutaj również wykorzystanie struktury wydaje się dosyć intuicyjne i naturalne.

```
1 Struktura Wezel_drzewa_binarnego {  
2     wartosc  
3     lewy_syn  
4     prawy_syn  
5     ojciec  
6 }
```

Zastanów się, czy widzisz jakieś zastosowanie struktury dla siebie.

Słownik

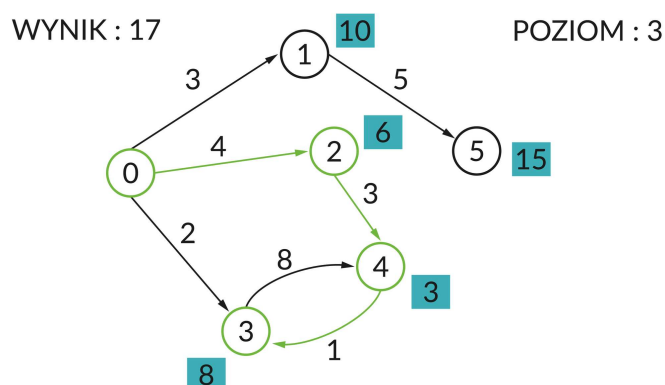
drzewo binarne

specjalny rodzaj drzewa, w którym każdy węzeł może mieć maksymalnie dwóch synów

Prezentacja multimedialna

Polecenie 1

Przeanalizuj prezentację przedstawiającą użycie struktur do implementacji bardziej skomplikowanych programów.



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

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Chcielibyśmy zaimplementować logikę do pewnej prostej gry komputerowej. Na każdym poziomie gracz będzie poruszał się po skierowanym grafie z wagami.

2

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Waga na danym połączeniu między dwoma wierzchołkami będzie oznaczać, ile czasu musimy poświęcić na podróż między nimi. Dodatkowo każdy wierzchołek ma swoją unikatową wartość, która zostaje dodana do

wyniku gracza za każdym razem, gdy wierzchołek zostanie przez niego odwiedzony.

Zadaniem gracza na każdym poziomie jest znalezienie takiej trasy z określonego punktu startowego, aby zdobyć jak najwięcej punktów w określonym czasie.

3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Implementację zapiszemy w pseudokodzie, aby zachować uniwersalność kodu. Do reprezentacji poszczególnych obiektów w grze użyjemy tablic i struktur. Są to:

- Gracz,
- Poziom,
- Graf,
- Połączenie.

4

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Zacznijmy od implementacji połączeń. Jako że graf jest ważony, to każde połączenie zawiera swoją własną wagę. Dodatkowo jest on skierowany, więc musimy uwzględnić kierunek połączenia.

```
1 Struktura Polaczenie {  
2     od  
3     do  
4     waga  
5 }
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Następnie przechodzimy do implementacji grafu. Nasza struktura powinna zawierać liczbę miast oraz tablice połączeń i wartości dla konkretnych miast. Zakładamy, że punktem startowym w grafie jest zawsze wierzchołek o numerze 0 oraz że maksymalna liczba miast wynosi 10, a połączeń 20.

```
1 Struktura Graf {
2     liczba_miast
3     wartosci_miast[10]
4     Polaczenie
5     polaczenia[20]
6 }
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Pozostało nam jeszcze zaimplementować strukturę reprezentującą gracza. Będzie ona przechowywała uzyskany przez niego wynik na danym poziomie, wybraną trasę, czas potrzebny na jej pokonanie oraz jej długość, czyli liczbę wykorzystanych połączeń.

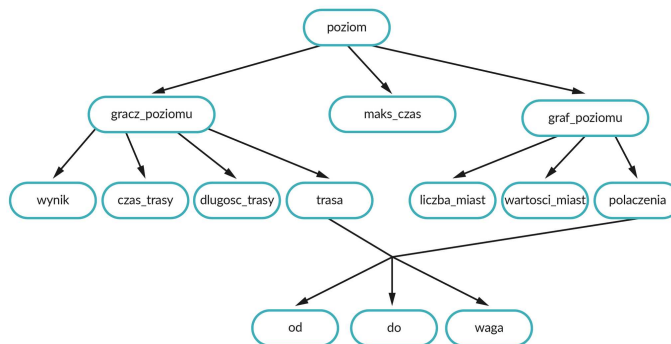
```
1 Struktura Gracz {
2     wynik
3     czas_trasy
4     dlugosc_trasy
5     Polaczenie trasa[20]
6 }
```

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Dodatkowo nasza gra będzie składała się z kilku poziomów. Każdy będzie miał swój własny maksymalny czas do wykorzystania na pokonanie trasy.

```
1 Struktura Poziom {
2     Gracz gracz_poziomu
3     maks_czas
4     Graf graf_poziomu
5 }
```



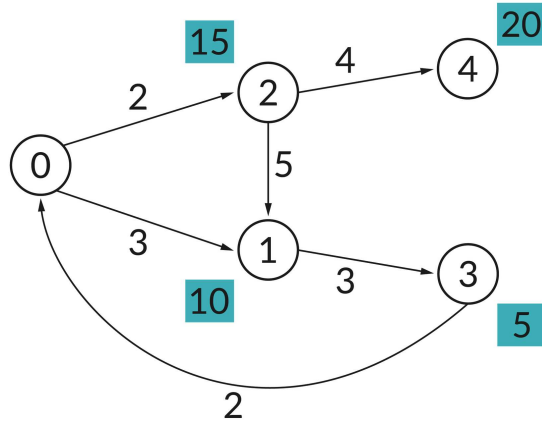
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Mamy zatem zdefiniowane podstawowe obiekty oraz utworzone relacje między nimi. Pozostaje zadeklarować odpowiednią liczbę poziomów naszej gry.

```
1 Poziom poziomy_gry[5]
```



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

Materiał audio dostępny pod adresem:

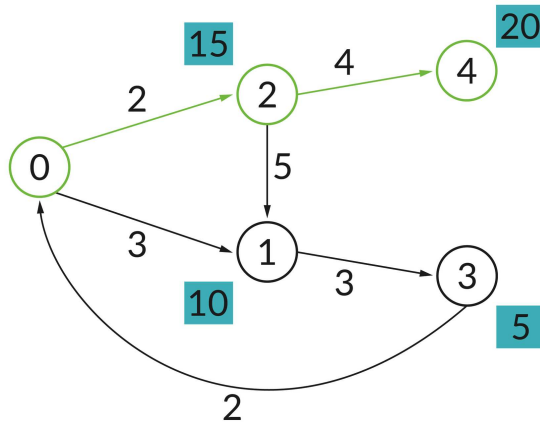
<https://zpe.gov.pl/b/P6JZn9oqd>

Zaimplementujemy teraz pierwszy poziom w naszej grze.

```

1 // najpierw utworzymy graf
2 Graf graf
3 graf.liczba_miast ← 5
4 graf.wartosci_miast ← {0,
5   10, 15, 5, 20}
6 graf.polaczenia ← {
7   {0, 1, 3},
8   {0, 2, 2},
9   {2, 1, 5},
10  {2, 4, 4},
11  {1, 3, 3},
12  {3, 0, 2}
13 }
14 // nastepnie poziom
15 poziomy_gry[0].maks_czas ←
16   15
17 poziomy_gry[0].graf_poziom
18   u ← graf

```



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

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P6JZn9oqd>

Założmy, że gracz odwiedził wierzchołki w następującej kolejności: 2, 4. Napiszmy pseudokod liczący czas potrzebny na przebycie jego trasy oraz uzyskany przez niego wynik.

```

1 Gracz gracz
2 gracz.trasa ← {{0, 2, 2},
  {2, 4, 4}}
3 gracz.dlugosc_trasy ← 2
4
5 i ← 0
6 wynik ← 0
7 czas_trasy ← 0
8
9 Dopóki i <
  gracz.dlugosc_trasy
  wykonuj
10   wynik ← wynik +
    graf.wartosci_miast[gracz.
      trasa[i].do]
11
12   czas_trasy ←
    czas_trasy +
    gracz.trasa[i].waga
13   i ← i + 1
14
15 gracz.wynik ← wynik
  
```

```
16 gracz.czas_trasy ←  
   czas_trasy  
17 poziomy_gry[0].gracz_pozio  
   mu ← gracz  
18 wypisz("Wynik: ",  
   poziomy_gry[0].gracz_pozio  
   mu.wynik)  
19 wypisz("Dlugosc trasy: ",  
20  
   poziomy_gry[0].gracz_pozio  
   mu.dlugosc_trasy)
```

Polecenie 2

Zastanów się samodzielnie nad możliwym rozbudowaniem zaprezentowanego przykładu.



1

1

Polecenie 3

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

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, ilu maksymalnie synów może mieć węzeł drzewa binarnego.

☐ czterech

☐ jednego

☐ trzech

☐ dwóch

Ćwiczenie 2



Wskaż, które stwierdzenie jest prawdziwe.

☐ Strukturę możemy wykorzystać jedynie do implementacji elementu listy dwukierunkowej.

☐ Strukturę możemy wykorzystać jedynie do implementacji elementu listy dwukierunkowej i węzła drzewa binarnego.

☐ Strukturę możemy wykorzystać zarówno do implementacji elementu listy jednokierunkowej, jak i dwukierunkowej.

Ćwiczenie 3



Wskaż, z czym głównie powiązane jest słowo „rekord”.

- ☐ z programowaniem
- ☐ ze złożonością obliczeniową
- ☐ z bazami danych
- ☐ z algorytmami

Ćwiczenie 4



Wskaż, czy podane stwierdzenie jest prawdziwe, czy fałszywe.

Struktura i rekord są koncepcyjnie tą samą strukturą danych.

- ☐ Fałsz
- ☐ Prawda

Ćwiczenie 5



Wskaż, co oznacza etykietowalność składowych struktury.

- ☐ Składowe struktury nie muszą mieć unikatowych nazw. Do konkretnej zawartości możemy się odwoływać dzięki indeksowaniu.
- ☐ Składowe struktury nie mogą mieć dwóch różnych nazw - jest to niezgodne z ich koncepcją.
- ☐ Składowe struktury mają unikatowe nazwy, dzięki czemu możemy się odwoływać do konkretnej zawartości tych składowych.

Ćwiczenie 6



Zaznacz wszystkie niezbędne składowe struktury reprezentującej element listy dwukierunkowej.

☐ następnik

☐ początek

☐ syn

☐ wartość

☐ koniec

☐ poprzednik

☐ ojciec

Ćwiczenie 7



Korzystając z zaproponowanych poniżej elementów, uzupełnij tekst tak, aby jego treść była prawdziwa.

Używanie struktur jest bardzo przydatne w pracy z danymi statystycznymi. Zazwyczaj mają one wiele właściwości, które mogą być struktury. Jeżeli potrzebujemy dużej liczby obiektów naszej struktury, warto skorzystać z .

struktury tablic

tablicy struktur

tagowalnych

właściwościami

etykietowalnych

składowymi

Ćwiczenie 8



Dopasuj elementy do odpowiednich grup.

słownictwo związane ze strukturą

słownictwo niezwiązane ze strukturą

dziecko

pesel

następnik

składowe

rekord

poprzednik

etykiety

Dla nauczyciela

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

Przedmiot: Informatyka

Temat: Podstawowe struktury danych: rekord

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum

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

1) w zależności od problemu rozwiązuje go, stosując metodę wstępującą lub zstępującą;

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 rekord.
- Poznasz podstawowe pojęcia związane z rekordem.
- Przeanalizujesz przykładowe zastosowania rekordów.

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-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- telefony z dostępem do internetu.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Podstawowe struktury danych: rekord”. Nauczyciel prosi uczniów o zapoznanie się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla i odczytuje temat lekcji oraz cele zajęć. Prosi uczniów o sformułowanie kryteriów sukcesu.

2. Prowadzący prosi uczniów, aby zgłaszali swoje propozycje pytań do tematu. Jedna osoba może zapisywać je na tablicy. Gdy uczniowie wyczerpią swoje pomysły, a pozostały jakieś ważne kwestie do poruszenia, nauczyciel je dopowiada.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Przeczytaj”.
2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie zapoznają się z treściami przedstawiającymi użycie struktur do implementacji bardziej skomplikowanych programów. Następnie samodzielnie zastanawiają się nad możliwym rozbudowaniem przedstawionego przykładu.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-6 z sekcji „Sprawdź się”, a następnie dzielą się wynikami swojej pracy z kolegą lub koleżanką.

Faza podsumowująca:

1. Nauczyciel ponownie wyświetla na tablicy temat lekcji zawarty w sekcji „Wprowadzenie” i inicjuje krótką rozmowę na temat zrealizowanych celów (czego uczniowie się nauczyli).
2. Na koniec zajęć nauczyciel prosi uczniów o rozwinięcie zdania: „Na dzisiejszych zajęciach nauczyłam/łem się jak...”.

Praca domowa:

1. Uczniowie wykonują ćwiczenia 7-8 z sekcji z sekcji „Sprawdź się”.

Wskazówki metodyczne:

- Treści w sekcji „Prezentacja multimedialna” można wykorzystać jako materiał służący powtórzeniu materiału.