



Podstawowe struktury danych: lista

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



Źródło: Kelly Sikkema, domena publiczna.

Cechą charakteryzującą dobrego programistę jest znajomość różnych struktur danych. Szybkość działania programów wzrasta, gdy przetwarzają one informacje zorganizowane w odpowiedni sposób.

Należy zatem poznać właściwości podstawowych struktur danych, aby później – jak najlepiej – korzystać z nich podczas rozwiązywania konkretnych problemów. W tym e-materiale omówimy jedną z takich struktur: listę.

Listy są preferowaną strukturą danych w sytuacjach, gdy nie wiemy, ile elementów docelowo będzie przechowywanych. Pozwalają one na szybkie dodawanie i usuwanie elementów. Lista wykorzystywana jest m.in. przy mailingu. Osoby odpowiedzialne za rozsyłanie np. promocji twojego ulubionego sklepu z grami nie są w stanie stwierdzić, ile osób zapisze się do newslettera. W związku z tym nie mogą użyć tablicy, by przechować adresy e-mail swoich użytkowników. Zastosowanie listy ma więcej sensu – swobodnie można dodawać adresy, kasować je lub zamieniać miejscami.

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

- [Podstawowe struktury danych,](#)
- [Podstawowe struktury danych: tablica,](#)
- [Podstawowe struktury danych: rekord,](#)

- Podstawowe struktury danych: stos i kolejka.

Twoje cele

- Wyjaśnisz, czym charakteryzuje się lista.
- Wskażesz różnice między listami a tablicami.
- Wymienisz różne typy list.
- Przeanalizujesz operacje wykonywane na listach.

Przeczytaj

Listy: podstawowe informacje

Lista jest strukturą danych zbudowaną z elementów tego samego typu, w której elementy są ułożone w porządku liniowym. W tym e-materiale omówimy dwa rodzaje list: jednokierunkowe oraz dwukierunkowe.

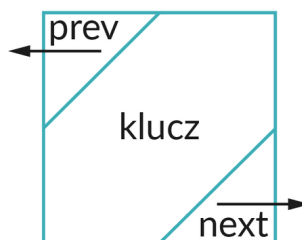
Lista to dynamiczna struktura danych, dlatego pamięć na bieżąco jest przydzielana jej nowym elementom. Pozwala to na budowanie list o dowolnej długości, zmieniającej się w czasie działania programu.

Jak jest tworzona lista? Otóż wykorzystuje się przy tym [wskaźniki](#). Każdy element listy jednokierunkowej poza polem przechowującym klucz zawiera pole wskazujące na następny element listy (*next*). Element listy dwukierunkowej zawiera dodatkowo wskaźnik do elementu poprzedniego (*prev*).

Elementy należące do listy nie muszą zajmować ciągłego obszaru w pamięci – mogą być rozmieszczone w różnych jej częściach. Z tego właśnie powodu są wykorzystywane wskaźniki. W przypadku gdy chcemy odwołać się do któregośkolwiek elementu, musimy po kolei odczytywać wskaźniki przechowywane w elementach poprzednich.

Zamieszczone niżej rysunki pokazują schematyczną budowę elementu

- listy dwukierunkowej:



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

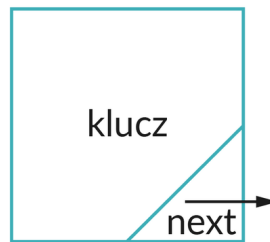
Ważne!

Każdy element listy dwukierunkowej musi się składać co najmniej z trzech pól – *key* (pole zawierające klucz elementu), *next* oraz *prev*.

Dla danego elementu listy *a* atrybut *next[a]* wskazuje na jego następnika, *prev[a]* – poprzednika.

Warto zaznaczyć, że jeśli *prev[a] = NIL*, element *a* nie ma poprzednika – to znaczy, że jest pierwszym elementem listy; jeśli *next[a] = NIL*, element nie ma następnika – jest ostatnim elementem listy.

- listy jednokierunkowej.



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

Ważne!

Każdy element listy jednokierunkowej składa się z minimum dwóch pól – *key* (pole zawierające klucz elementu) oraz *next*.

Dla danego elementu *a* na liście atrybut *next [a]* wskazuje na jego następnik.

Pamiętajmy, że jeśli *next [a] = NIL*, element nie ma następnika – jest ostatnim elementem listy.

Zazwyczaj, tworząc listę, rezerwuje się w pamięci dodatkowo dwie zmienne wskaźnikowe do jej obsługi: *head* (głowa – wskazuje na pierwszy element listy) i *tail* (ogon – wskazuje na ostatni element listy).

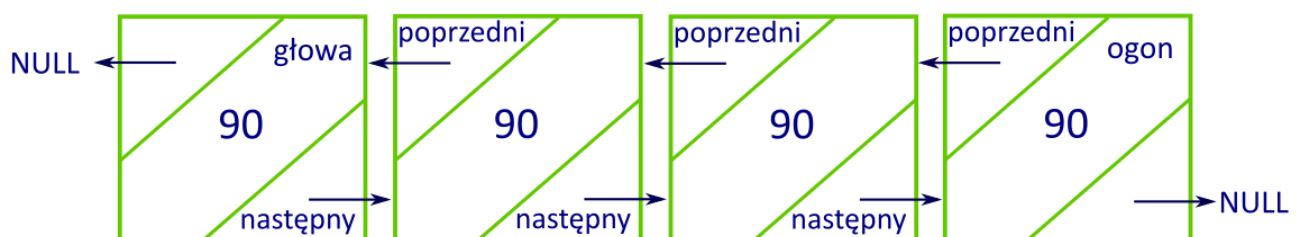
Przyjrzyjmy się dwóm szczególnym elementom listy tak jedno-, jak i dwukierunkowej.

- Gdy element nie ma poprzednika, jest on pierwszym elementem. Wskazująca na niego zmienna *head* po polsku jest nazywana głową listy.
- Gdy element nie ma następnika, będzie on ostatnim elementem listy. Wskazująca na niego zmienna *tail* po polsku nazywa się ogonem listy.

Rodzaje list

Rozróżniamy następujące rodzaje list:

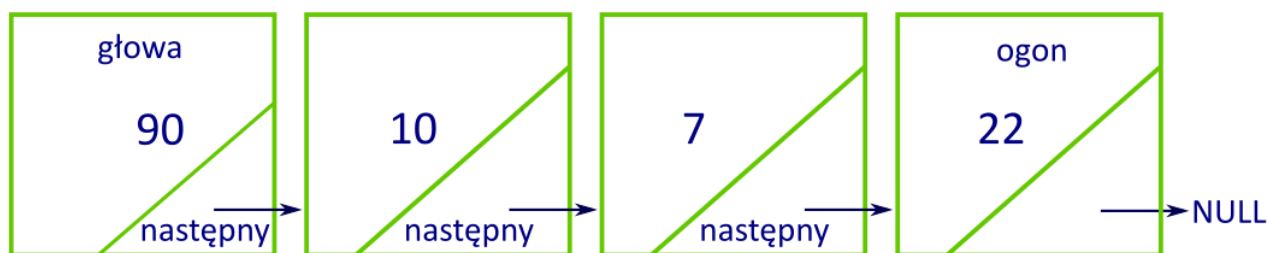
- **Lista dwukierunkowa** – lista, w której możemy poruszać się w dwóch kierunkach. Każdy element zawiera co najmniej trzy pola: *klucz*, *prev* oraz *next*. W rezultacie elementy wskazują zarówno swoje poprzedniki, jak i następniki.



Lista dwukierunkowa

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

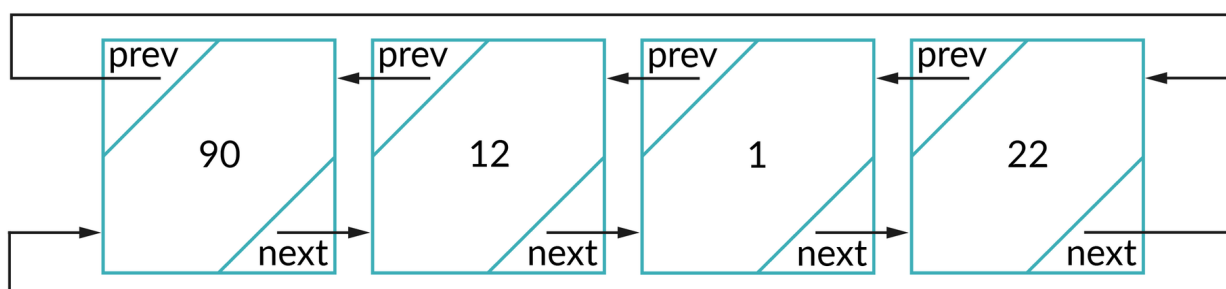
- **Lista jednokierunkowa** – lista, w której możemy poruszać się po elementach tylko w jednym kierunku. Każdy element zawiera minimum dwa pola (*klucz*, *next*). W takim przypadku nie da się przejść do poprzedniego elementu.



Lista jednokierunkowa

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

- **Lista cykliczna dwukierunkowa** – lista, w której wskaźnik *next* ostatniego elementu przechowuje adres pierwszego elementu, a wskaźnik *prev* pierwszego elementu wskazuje ostatni element, jest listą cykliczną dwukierunkową. W jej przypadku możemy przechodzić w kółko po elementach w obu kierunkach. Natomiast w liście cyklicznej jednokierunkowej poruszamy się tylko w jedną stronę.



Lista cykliczna dwukierunkowa

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

Różnice między listą a tablicą

W związku z tym, że elementy listy mogą być umieszczone w różnych miejscach pamięci, nie musimy rezerwować dla niej z góry ustalonego, ciągłego obszaru (tak jak dzieje się w przypadku tablic). W rezultacie do listy można dodawać nowe elementy.

Aby odwołać się do któregośkolwiek elementu tablicy, wystarczy podać jego indeks. W przypadku list jest inaczej. Gdy chcemy odwołać się do wybranego elementu, musimy przejść kolejno przez wszystkie elementy go poprzedzające. Cecha ta jest rezultatem sekwencyjnej budowy listy, a zatem jej elementy nie są oznaczone za pomocą indeksów (tak jak w przypadku tablic), lecz odwołują się do siebie kolejno (obiekt A wskazuje obiekt B, obiekt B wskazuje obiekt C i tak dalej).

Słownik

wskaźnik

zmienna przechowująca adres innej zmiennej

złożoność czasowa

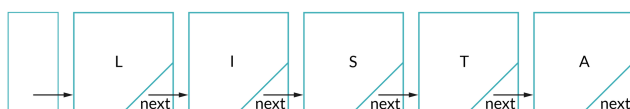
czas niezbędny do rozwiązania określonego problemu, podawany zazwyczaj w liczbach operacji podstawowych/dominujących (zależy ona od ilości danych wejściowych)

Aplet

Polecenie 1

Zapoznaj się z operacjami wyszukiwania, wstawiania oraz usuwania elementów listy jednokierunkowej.

Następnie postaraj się samodzielnie odtworzyć wszystkie czynności.

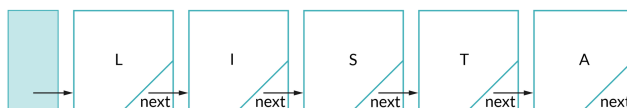


Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Pierwszą operacją będzie wyszukiwanie elementu. W liście jednokierunkowej przedstawionej na rysunku wyszukajmy element o kluczu „I”

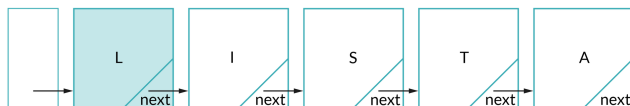
2



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Zaczynamy od wskaźnika *head*. To on poda nam adres pierwszego elementu listy.



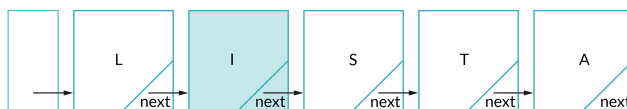
3

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Znajdujemy się w pierwszym elemencie listy. Jego klucz nie zgadza się z poszukiwanym kluczem, więc przechodzimy do następnego elementu. Korzystamy w tym celu z wartości pola next.

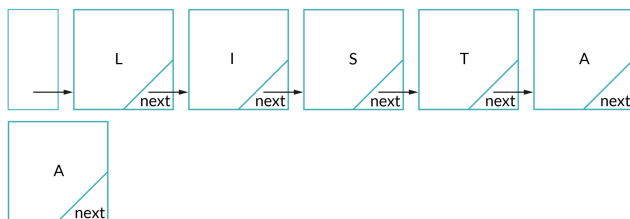
4



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Tym razem klucz elementu zgadza się z wartością poszukiwaną. Znaleźliśmy interesujący nas element, więc kończymy algorytm. Pamiętaj, że musieliśmy sprawdzić poprzedni element ciągu.

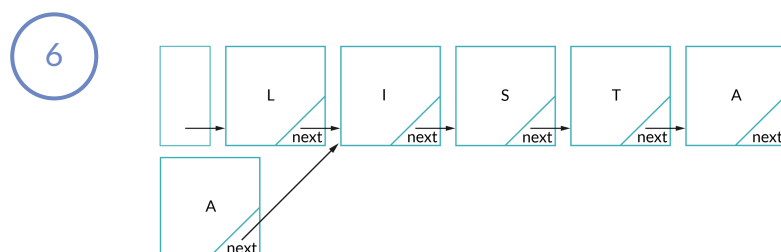


5

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

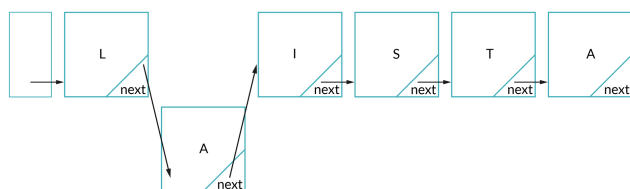
Następnym zadaniem jest wstawienie elementu do wnętrza listy. Na drugim miejscu umieścimy wartość „A”. Zaczniemy od utworzenia nowej komórki oraz zapisania pola *klucz*.



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

W polu wskaźnika *next* podajemy adres następnika, przed którym ma być wstawiony nasz element.

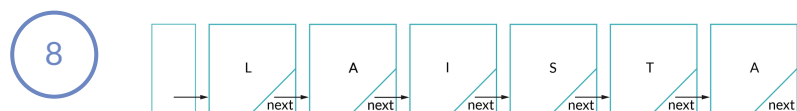


7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

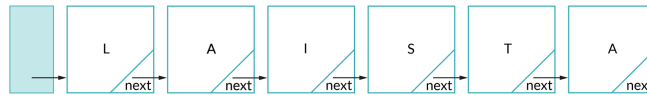
Następnie wskaźnik poprzednika ustawiamy na adres wstawianego elementu.



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Otrzymaliśmy listę ze wstawionym elementem.



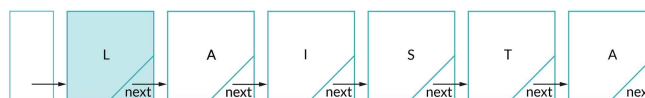
9

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Ostatnią operacją, którą wykonamy, będzie usuwanie elementu. Usuniemy element o wartości „L”. Musimy ją odszukać.

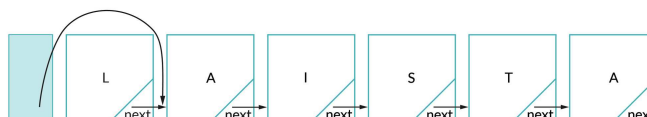
10



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Jesteśmy w pierwszym elemencie listy.
Porównujemy wartość *klucza* z usuwaną wartością. Okazuje się, że od razu znaleźliśmy element przeznaczony do usunięcia.



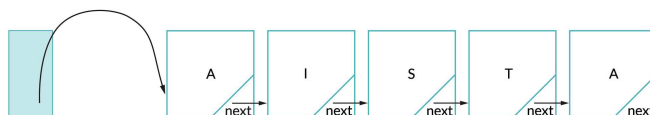
11

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Wskaźnik *head* ustawiamy na następnik kasowanego elementu.

12



Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/Puhk7mD92>

Usuujemy wartość „L” z pamięci.



13

Materiał audio dostępny pod adresem:

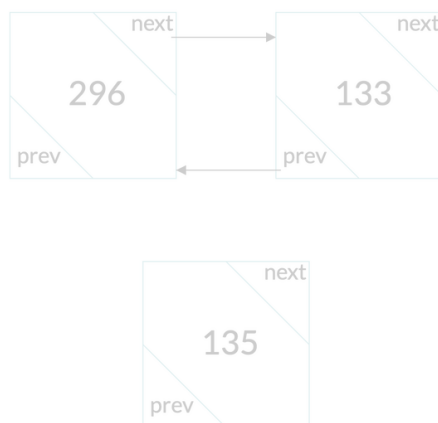
<https://zpe.gov.pl/b/Puhk7mD92>

W rezultacie otrzymujemy nową listę.

Polecenie 2

Uruchom aplet prezentujący, w jaki sposób do listy dwukierunkowej dodawany jest nowy element pomiędzy dwa już istniejące. Sprawdź, jakie odwołania są przypisywane wskaźnikom węzłów w kolejnych krokach.

Prezentacja kroków dodawania elementu do listy dwukierunkowej



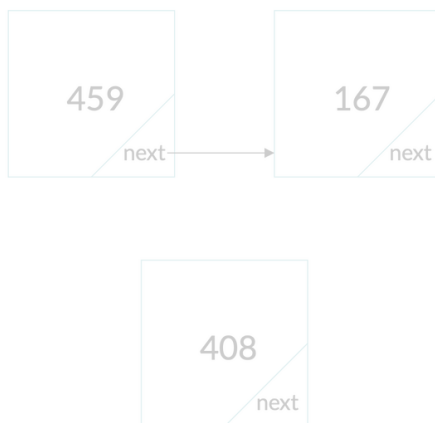
Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/D68RkABS7>

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

Polecenie 3

Uruchom aplet przedstawiający wstawianie elementu do listy jednokierunkowej w miejscu pomiędzy dwoma elementami. Porównaj zmiany odwołań wskaźników dla tej samej operacji na liście dwukierunkowej.

Prezentacja kroków dodawania elementu do listy jednokierunkowej



Zasób interaktywny dostępny pod adresem <https://zpe.gov.pl/a/D68RkABS7>

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Zaznacz zdania poprawnie opisujące listy.

- ☐ Aby dostać się do wybranego elementu listy, należy podać jego indeks.
- ☐ Lista jest sekwencyjną strukturą danych.
- ☐ W listach dwukierunkowych z jednego elementu listy możemy przenieść się do następnego lub poprzedniego elementu.
- ☐ Lista jest zbudowana z elementów różnych typów.

Ćwiczenie 2



Wybierz zdanie, które nie opisuje list jednokierunkowych.

- ☐ Elementy mogą być umieszczone w różnych miejscach w pamięci.
- ☐ Do wybranego elementu listy docieramy przez przechodzenie po wszystkich jego poprzednikach.
- ☐ Elementy muszą być umieszczone w ciągłym obszarze pamięci.

Ćwiczenie 3



Uzupełnij zdanie.

Element listy dwukierunkowej, poza polem danych () , zawiera pole wskazujące element listy (*next*), oraz element wskazujący element listy ().

następny

prev

klucz

poprzedni

Ćwiczenie 4



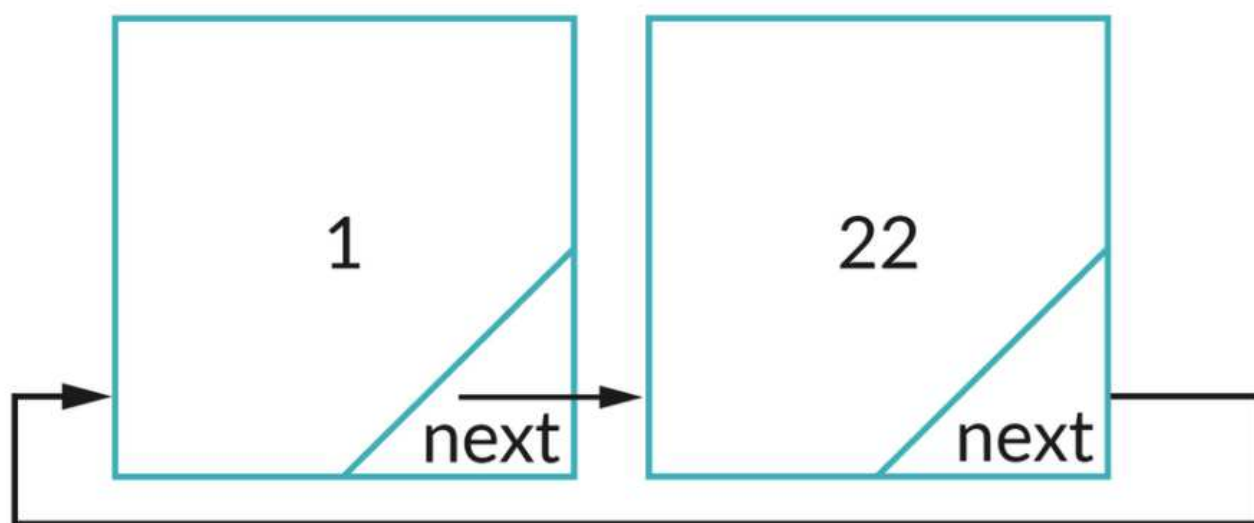
Wskaż zdania, które poprawnie opisują listę cykliczną dwukierunkową.

- ☐ Jest to lista, której elementy zawierają dwa pola.
- ☐ Jest to lista, w której możemy przejść zarówno do poprzednika, jak i następnika elementu.
- ☐ Jest to lista, w której elementy zawierają wskaźniki *prev* oraz *next*.
- ☐ Jest to lista, w której możemy poruszać się po elementach tylko w jednym kierunku.

Ćwiczenie 5



Wskaż, która lista została przedstawiona na ilustracji.



☐ lista cykliczna dwukierunkowa

☐ lista cykliczna jednokierunkowa

☐ lista jednokierunkowa

☐ lista dwukierunkowa

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

Ćwiczenie 6



Zapisz specyfikację problemu wstawiania elementu X do wnętrza listy jednokierunkowej za elementem A. Przygotuj listę kroków.

Specyfikacja problemu:

Dane:

Wynik:

Lista kroków:

Ćwiczenie 7



Uporządkuj kolejne kroki algorytmu usuwania elementu z listy jednokierunkowej.

Wartość wskaźnika poprzednika usuwanego elementu ustaw na wartość przechowywaną w polu wskaźnika usuwanego elementu.



Zapamiętaj adres poprzednika usuwanego elementu.



Usuń element z pamięci.



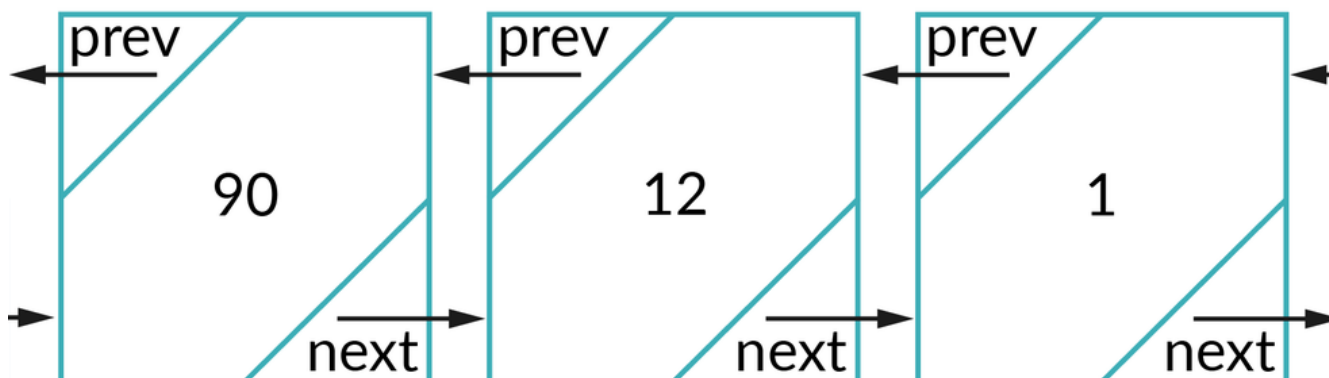
Wyszukaj element, który chcesz usunąć w liście.



Ćwiczenie 8

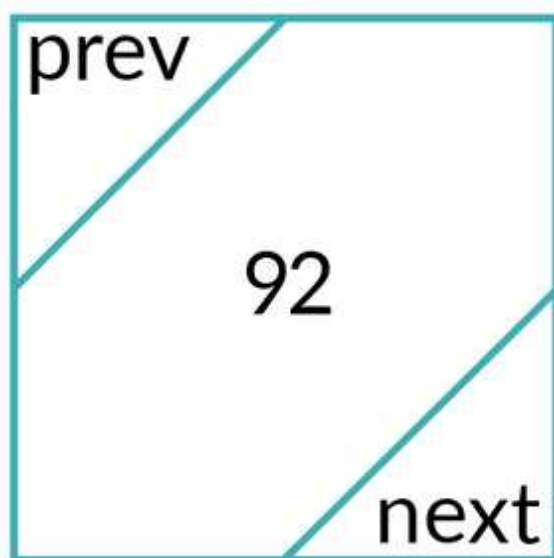


Założmy, że posiadamy listę dwukierunkową, której fragment jest pokazany poniżej.



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

Do listy chcemy dodać kolejny element, którego wartość wynosi 92. Powinien on znaleźć się między elementami o wartościach 90 oraz 12.



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

Połącz w pary wskaźniki oraz to na co wskazują, **po** wykonaniu operacji dodania nowego elementu.

wskaźnik poprzednika elementu 92

element 12

wskaźnik następnika elementu 12

element 92

wskaźnik poprzednika elementu 12

element 1

wskaźnik następnika elementu 92

element 90

Dla nauczyciela

Autor: Zespół autorski Contentplus.pl sp. z o.o.

Przedmiot: Informatyka

Temat: Podstawowe struktury danych: lista

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres 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 rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

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

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

3) objaśnia, a także porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przy tym przykłady problemów i algorytmów, w szczególności:

i) struktury dynamiczne: stos, kolejka, lista (do realizacji algorytmu: ONP, symulacji problemu Flawiusza, sortowania leksykograficznego),

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):

- Wyjaśnisz, czym charakteryzuje się lista.
- Wskażesz różnice między listami a tablicami.
- Wymienisz różne typy list.
- Przeanalizujesz operacje wykonywane na listach.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

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

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.

Przebieg lekcji

Przed lekcją:

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

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów zajęć, przejście do wspólnego ustalenia kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel wyświetla na tablicy pytania zawarte w sekcji „Wprowadzenie”:
 - czym różni się lista od tablicy?
 - czym jest złożoność czasowa?
 - jak nazywamy zmienną, która przechowuje adres innej zmiennej?
 Ochotnicy na nie odpowiadają. Pozostali uczniowie uzupełniają ich wypowiedzi lub przedstawiają swoje propozycje.

Faza realizacyjna:

1. **Praca z multimedium.** Uczniowie w zespołach dwuosobowych zapoznają się z operacjami wyszukiwania, wstawiania oraz usuwania elementów listy. Następnie starają się odtworzyć wszystkie czynności.
2. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-8. Po wykonaniu każdego z nich następuje omówienie rozwiązania przez nauczyciela.

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).

Praca domowa:

1. Uczniowie wykonują polecenie nr 2 z sekcji „Aplet”.

Wskazówki metodyczne:

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