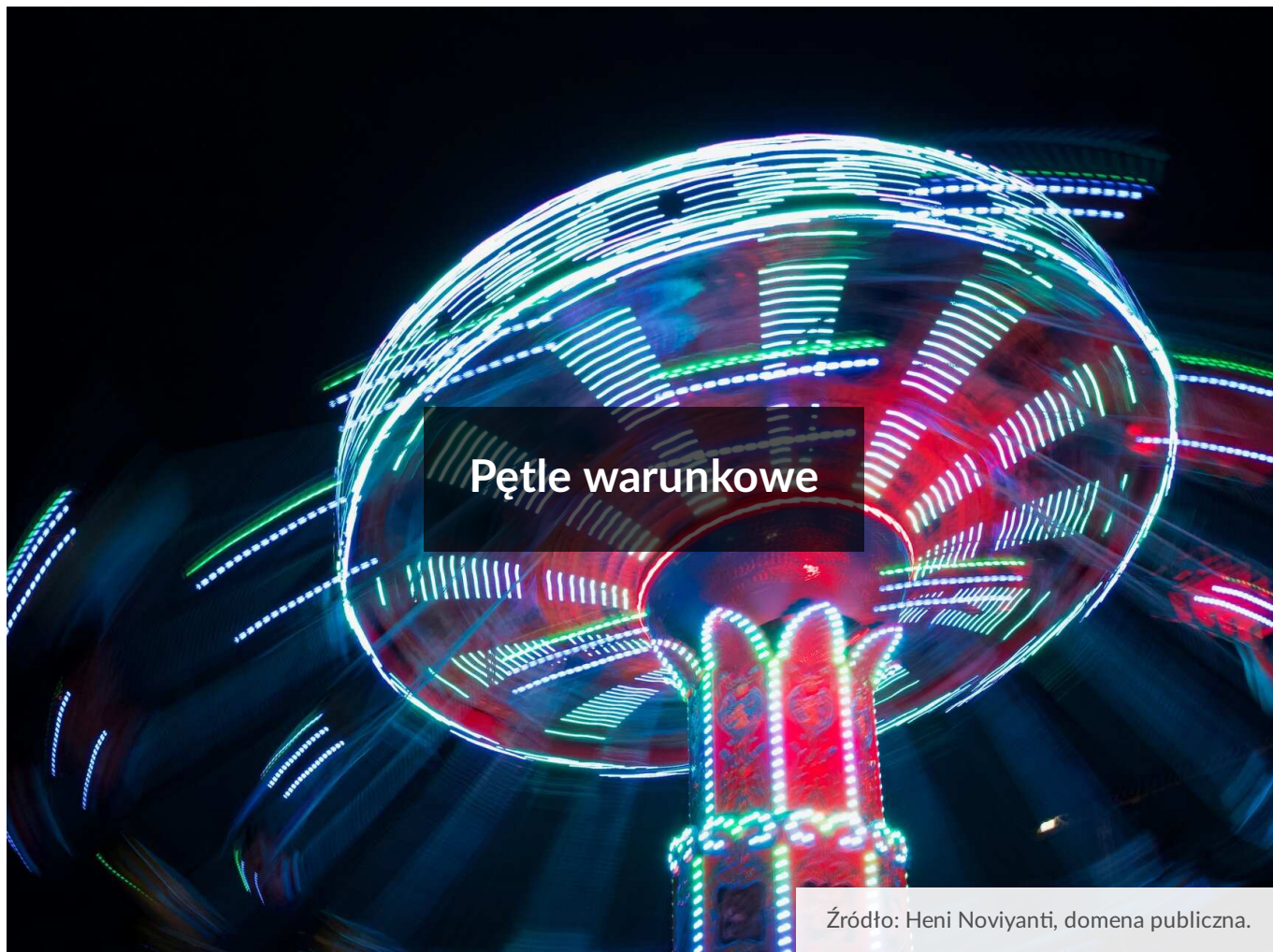




Pętle warunkowe

- Wprowadzenie
- Przeczytaj
- Schemat interaktywny
- Sprawdź się
- Dla nauczyciela



W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Iteracja to metoda wykorzystywana w [pętli for](#), która umożliwia powtarzanie pewnych operacji określoną z góry liczbę razy. W tym e-materiale omówimy działanie pętli [while](#), która powtarza operacje, aż do spełnienia określonego warunku. Przedstawimy także implementację tej pętli w schemacie blokowym.

Implementacje w poszczególnych językach programowania znajdziesz w e-materiałach:

- [Pętle warunkowe w języku C++](#),
- [Pętle warunkowe w języku Java](#),
- [Pętle warunkowe w języku Python](#).

Więcej zadań? Sięgnij do [Pętle warunkowe – zadania maturalne](#).

Twoje cele

- Przeanalizujesz budowę pętli warunkowej.
- Scharakteryzujesz problemy, do których rozwiązania można użyć pętli [for](#) i [while](#).
- Zapiszesz algorytm wykorzystujący pętlę [while](#).

Przeczytaj

Przypomnienie informacji o pętli for

Konstrukcja pętli `for` składa się z trzech głównych elementów, które są umieszczone wewnątrz definicji tej pętli:

- zmienna sterująca z przypisaną początkową wartością;
- warunek określający, do jakiego momentu pętla ma się wykonywać;
- zmiana wartości następująca po każdej iteracji.

Budowa i zastosowanie pętli while

Pętla `while` jest zbudowana inaczej. Zmienna sterująca jest tworzona przed zdefiniowaniem samej pętli. Warunek jest umieszczony, podobnie jak w pętli `for`, wewnątrz definicji, a zmiana wartości zmiennej sterującej następuje w ciele tej pętli, czyli tam, gdzie zdefiniowane są powtarzane operacje. Ta różnica może nie być widoczna na schemacie blokowym, jednak będzie ją można zaobserwować, gdy przejdziemy do implementacji pętli `while` w językach programowania.

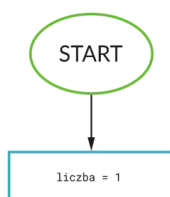
Teoretycznie pętla `while` może być używana w zasadzie w każdej sytuacji, w której mogłaby być użyta pętla `for`. Wszystko zależy od tego, co i w jaki sposób chcemy osiągnąć.

Najczęściej jednak pętli `for` używamy wtedy, gdy wiemy, ile dokładnie razy chcemy powtórzyć daną czynność. Natomiast pętlę `while` wykorzystujemy, gdy nie jest ważna liczba powtórzeń, a zależy nam tylko na spełnieniu określonego warunku.

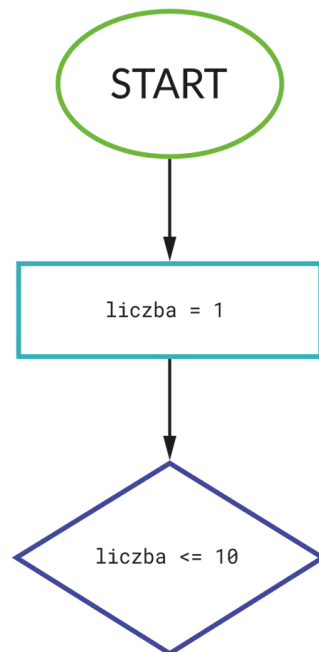
Przykładowe algorytmy wykorzystujące pętlę while

Wypisanie ciągu liczb całkowitych z zakresu 1 — 10

Tworząc nasz schemat blokowy, zaczniemy od zdefiniowania zmiennej `liczba`, która będzie przechowywać kolejno wyświetlane wartości liczbowe i przy okazji będzie zmienną sterującą dla naszej pętli `while`. Niech wartością startową tej zmiennej będzie 1.

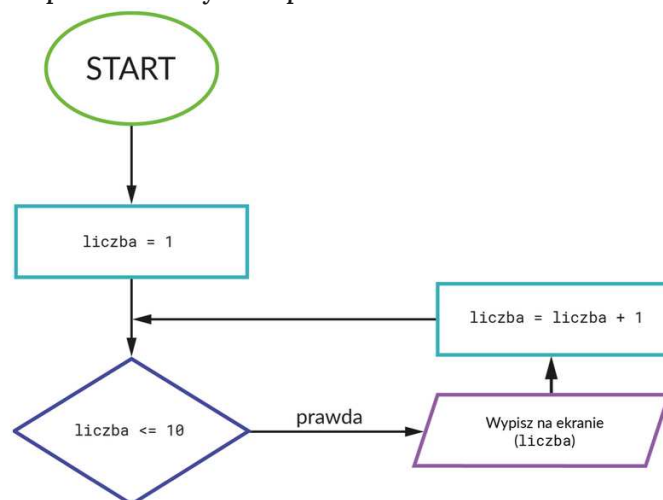


Zastanówmy się teraz, do jakiego momentu chcemy, aby nasza pętla działała. Skoro zaczynamy ciąg liczbowy od wartości 1, to pętla powinna się skończyć, gdy wartość zmiennej `liczba` przekroczy wartość 10. Dodajmy do schematu ten warunek:



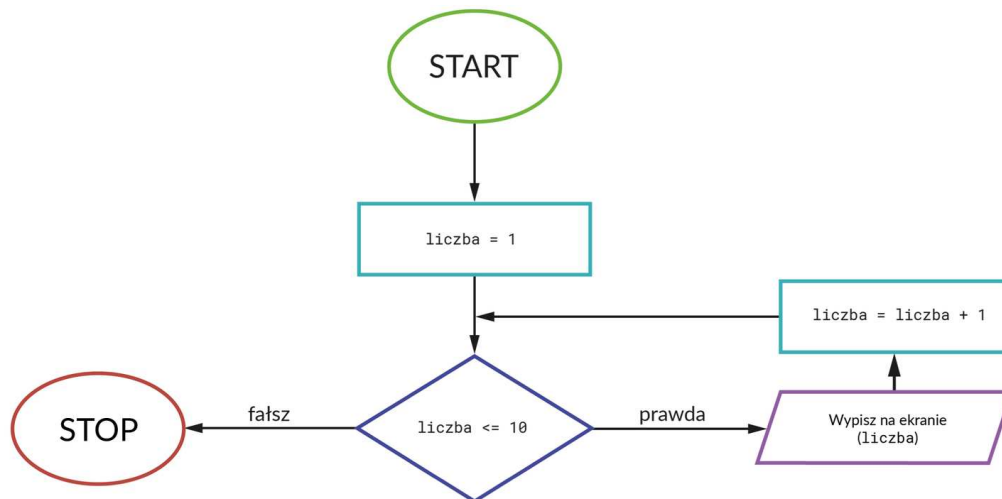
Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Jeśli wartość zmiennej `liczba` mieści się w określonym przedziale, wypisujemy jej wartość na ekranie, zwiększamy jej wartość o 1 i powracamy do sprawdzania warunku.



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

Jeśli wartość zmiennej `liczba` przekroczy wartość 10, oznacza to, że cały zakres liczb, którego oczekiwaliśmy, został wypisany na ekranie. W takim momencie możemy zakończyć nasz algorytm.



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

Ważne!

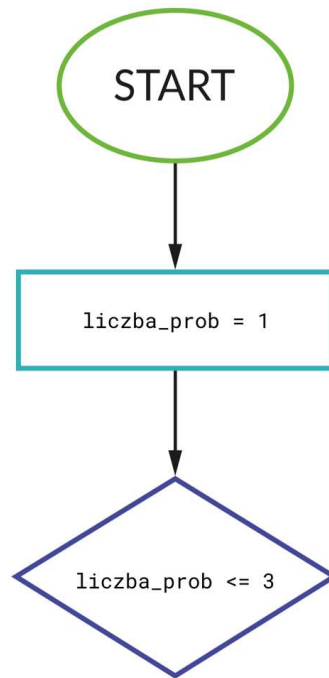
Zazwyczaj warunek w pętli `while` można zapisać na więcej sposobów, np. warunek `liczba <= 10` można zapisać też w postaci `liczba != 11`, a działanie pętli i rezultat będą takie same.

Wpisywanie kodu PIN w bankomacie

Wyobraźmy sobie, że piszemy program obsługujący bankomaty. Nasz schemat działania programu jest następujący:

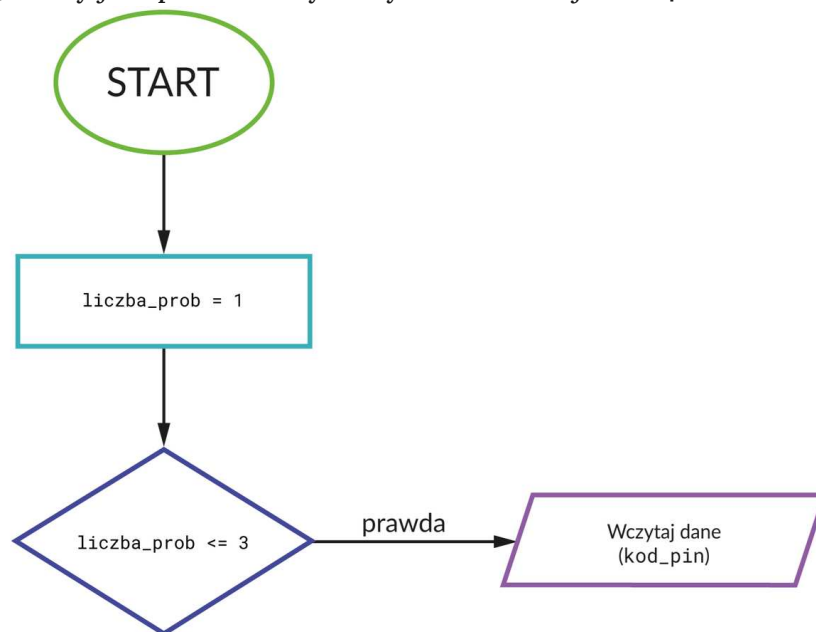
1. Użytkownik podaje kod **PIN** swojej karty, aby móc potwierdzić, że jest właścicielem karty.
2. Jeśli kod PIN jest błędny, program prosi ponownie o wpisanie kodu (użytkownik ma łącznie trzy próby).
3. Jeśli trzy próby zakończą się błędem, wypisujemy komunikat o zablokowaniu dostępu do tego konta i kończymy algorytm.
4. Jeśli natomiast którakolwiek z prób się uda, wypisujemy komunikat o udanej identyfikacji i kończymy algorytm.

Zacznijmy standardowo od startu algorytmu i utworzenia zmiennej `liczba_prob` przechowującej liczbę prób identyfikacji. Dodajemy też warunek sprawdzający, czy użytkownik próbował już 3-krotnie uzyskać dostęp do konta.



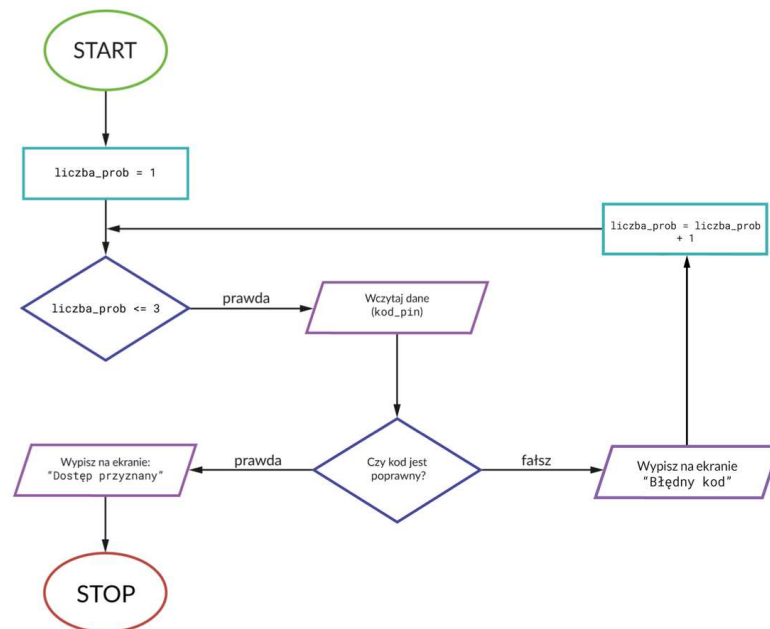
Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Dopóki użytkownik nie przekroczył dostępnej liczby prób, program wczytuje od użytkownika kod PIN, który jest przechowywany w zmiennej `kod_pin`.



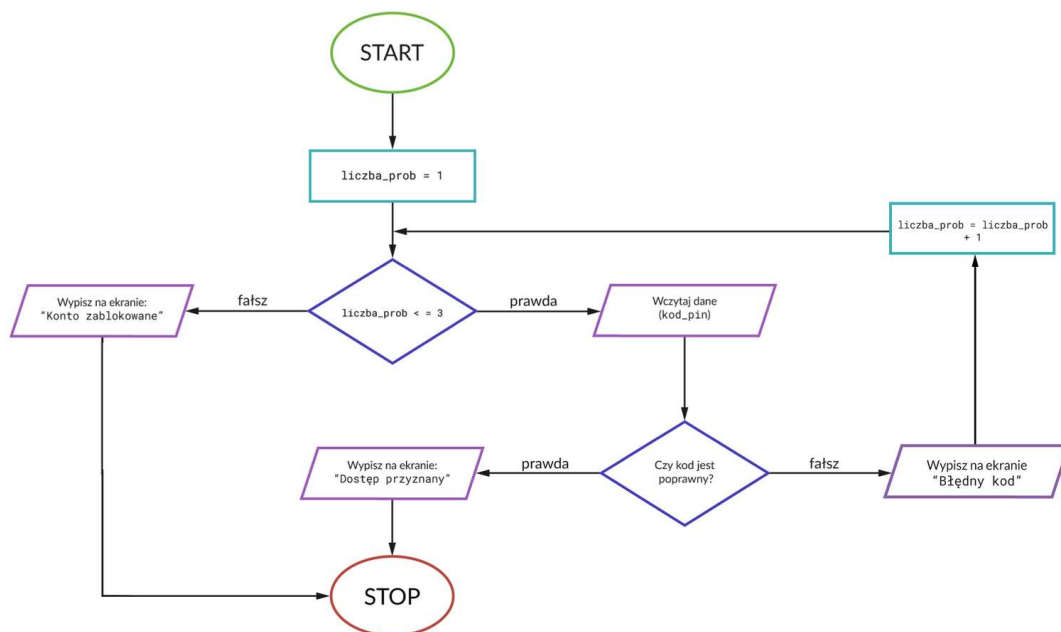
Źródło: Contentplus.pl sp. z o.o., licencja: CC BY-SA 3.0.

Jeśli kod PIN jest poprawny, to uzyskujemy dostęp do konta i kończymy algorytm, a jeśli nie, wypisujemy informację o błędzie, zwiększamy wartość zmiennej `liczba_prob` i ponawiamy operację.



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

W przypadku, gdy wszystkie próby okazały się błędne, blokujemy dostęp do konta i kończymy program.



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

Słownik

PIN

jest to akronim angielskiego wyrażenia *Personal Identification Number* (pol. osobisty numer identyfikacji); oznacza kod numeryczny służący do uwierzytelnienia, np. w telefonii komórkowej, bankowości itp.

Schemat interaktywny

Polecenie 1

Zapisz algorytm wypisywania ciągu liczb w taki sposób, że dla danej liczby całkowitej x wypisywana jest jej wartość, następnie jej wartości dziesiętne zwiększające się o 0.1, a na koniec kolejna liczba całkowita. Wykorzystaj do tego schemat interaktywny lub język programowania.

Specyfikacja:

Dane:

- x – liczba całkowita dodatnia

Wynik:

Program wypisuje najpierw liczbę całkowitą x , następnie jej wartości dziesiętne zwiększające się o 0.1. Na końcu wypisuje kolejną liczbę całkowitą.

Przykładowe wyjście dla $x = 1$:

1 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2

Polecenie 2

Porównaj swoje rozwiązanie z prezentacją.



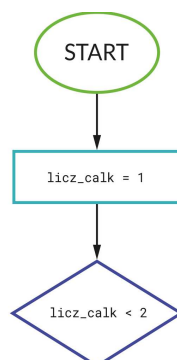
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Zacznijmy od zdefiniowania zmiennej `licz_calk`, która ma przechowywać naszą liczbę całkowitą. Zaczynamy od wartości 1.

2



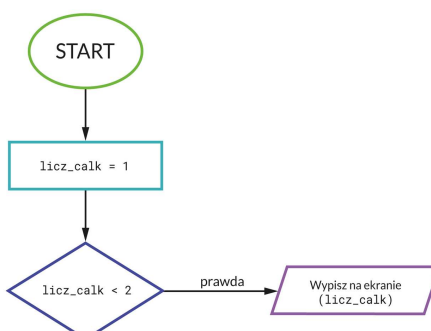
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

W tym miejscu tworzymy naszą pierwszą pętlę, która w tym przypadku zostanie wykonana tylko raz (chcemy, aby została wykonana tylko dla liczby 1).

3



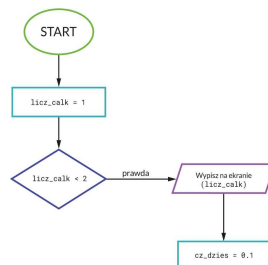
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Tak, jak wspomnieliśmy w treści polecenia, najpierw wypisujemy wartość liczby całkowitej, czyli wartość zmiennej `licz_calk`.

4



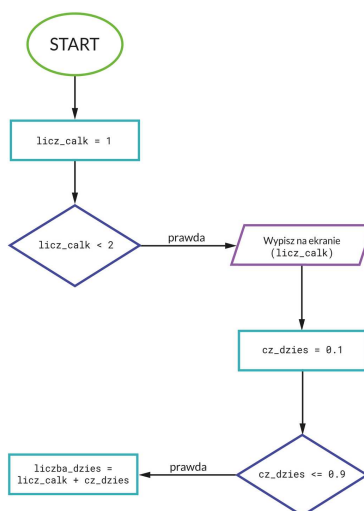
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Oprócz samych liczb całkowitych, chcemy też wyświetlać liczby z częścią dziesiętną. Do tworzenia takich liczb będziemy używać części dziesiętnych przechowywanych w zmiennej `cz_dzies`.

5



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

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

W tym miejscu stworzymy kolejną pętlę, która posłuży nam do tworzenia liczb dziesiętnych.

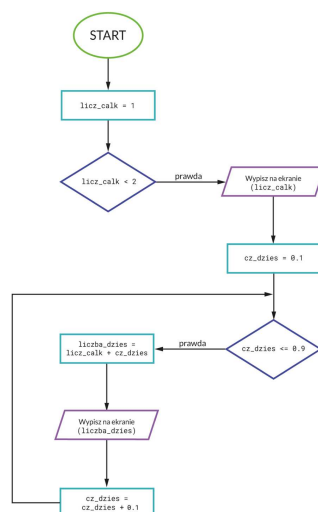
Wartości zmiennej `cz_dzies` będziemy dodawać do wartości zmiennej `licz_calk`. Korzystając z funkcji rzutowania typów (zamiany typów danych), wynikiem takiego dodawania będzie liczba zmiennoprzecinkowa (np. $1 + 0.1 = 1.1$). Do przechowywania tych wyników użyjemy zmiennej `liczba_dzies`.

6

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Rzutowanie typów danych w tym przypadku polega na zamianie typu całkowitoliczbowego na typ zmiennoprzecinkowy. Rzutowanie może odbywać się w różny sposób, zależy to od tego, w jakiej technologii tego mechanizmu używamy.



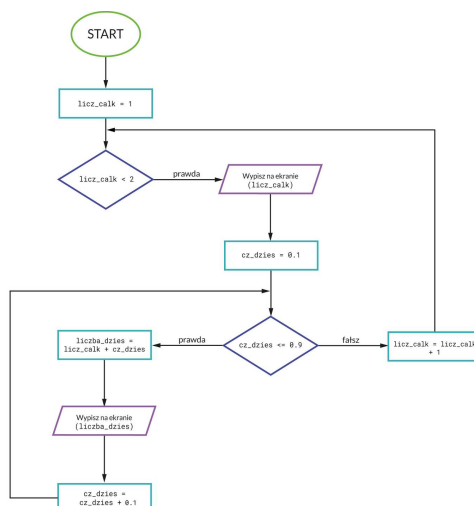
7

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

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Gdy stworzymy daną liczbę dziesiętną, należy ją wyświetlić na ekranie. Potem zwiększamy jej wartość o 0.1 i powracamy do sprawdzania warunku drugiej pętli.

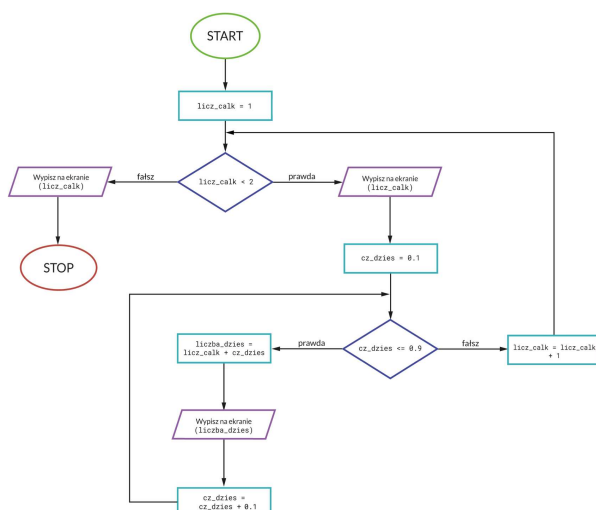


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Jeśli wypiszemy dla danej liczby całkowitej wszystkie jej wartości dziesiętne, przechodzimy do powtórzenia tych operacji dla kolejnej liczby całkowitej.



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

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Gdy wypiszemy już wszystkie wartości liczbowe, wypisujemy liczbę kończącą dany ciąg

liczb i przerywamy wykonywanie algorytmu.

10

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/P1HfRALDU>

Spróbuj teraz spisać na kartce pięć sytuacji w różnych aplikacjach, gdzie wskazane byłoby użycie pętli `while`.

Przykłady: wyświetlanie odpowiedniego stanu baterii w zależności od pozostałej ilości energii, odliczanie czasu pozostałego do restartu systemu itp.

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

Polecenie 3

Przygotuj algorytm, który wypisuje wszystkie liczby pierwsze z zadanego zakresu liczb naturalnych. Przetestuj swój program dla zakresu $\langle 17, 89 \rangle$.

Specyfikacja:

Dane:

m, n – liczby naturalne podane przez użytkownika, wyznaczają zakres, w którym szukane mają być liczby pierwsze

Wynik:

Program wypisuje wszystkie liczby pierwsze, jakie znajdują się w zakresie podanym przez użytkownika.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, czy w pewnych sytuacjach pętle `for` i pętle `while` można stosować zamiennie.

☐ nie

☐ tak

Ćwiczenie 2



Wskaż, której pętli lepiej użyć, jeśli chcesz na ekranie wypisać ten sam tekst pięć razy.

pętla `while`

pętla `for`

Ćwiczenie 3



Zaznacz, w jakiej sytuacji użyjesz pętli `while`.

☐ kilkukrotne nadanie zmiennej tej samej wartości

☐ wczytywanie odpowiedzi od użytkownika, dopóki nie poda poprawnej odpowiedzi

☐ wielokrotne zakończenie algorytmu

Ćwiczenie 4



Wskaż, czy zmienną sterującą w pętli `while` może być liczba zmiennoprzecinkowa.

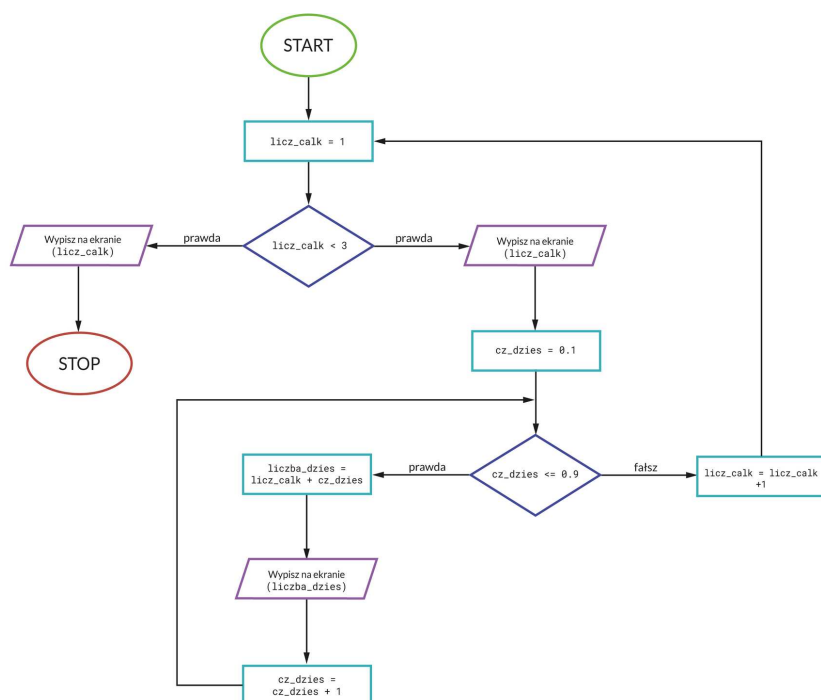
☐ nie

☐ tak

Ćwiczenie 5



Wskaż, jaki rezultat zobaczymy na ekranie po wykonaniu poniższego algorytmu.



☐ 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2

☐ 1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2 2.1 2.2 2.3 2.4 2.5
2.6 2.7 2.8 2.9 3

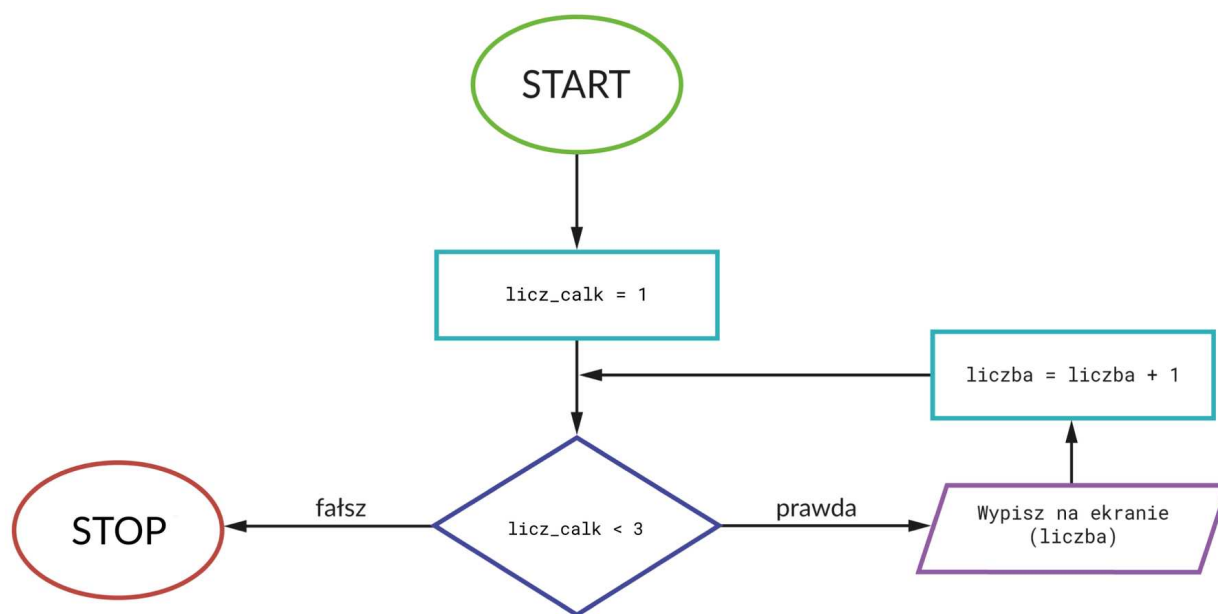
☐ 1 2 3

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

Ćwiczenie 6



Wskaż, ile razy wykona się pętla w algorytmie.



☐ nieskończenie wiele razy

☐ 2

☐ 0

☐ 1

Ćwiczenie 7



Wpisz odpowiedź na pytanie.

Czy implementacja pętli `for` i pętli `while` w schemacie blokowym może wydawać się podobna?

Ćwiczenie 8



Wskaż, która pętla może być użyta do wypisania ciągu liczb.

obie pętle mogą być użyte ☐

pętla `for` ☐

pętla `while` ☐

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Pętle warunkowe

Grupa docelowa:

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

Podstawa programowa:

Cele kształcenia – wymagania ogólne

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

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

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

2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;

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):

- Przeanalizujesz budowę pętli warunkowej.
- Scharakteryzujesz problemy, do których rozwiązania można użyć pętli `for` i `while`.
- Zapiszesz algorytm wykorzystujący pętlę `while`.

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.

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: „Pętle warunkowe”. Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel wyświetla temat i cele zajęć. Prosi uczniów, by na podstawie wiadomości zdobytych przed lekcją zaproponowali 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.** Jeżeli przygotowanie uczniów do lekcji jest niewystarczające, nauczyciel prosi o indywidualne zapoznanie się z treścią zawartą w sekcji „Przeczytaj”. Każdy uczestnik zajęć podczas cichego czytania wynotowuje najważniejsze kwestie poruszane w tekście.
2. **Praca z multimediu.** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”, czyta treść polecenia nr 1: „Zapisz algorytm wypisywania ciągu liczb w taki sposób, że dla danej liczby całkowitej x wypisywana jest jej wartość, następnie jej wartości dziesiętne zwiększające się o 0.1, a na koniec kolejna liczba całkowita. Wykorzystaj do tego schemat interaktywny lub język programowania” i omawia kolejne kroki rozwiązania.

3. Ćwiczenie umiejętności. Liga zadaniowa – uczniowie realizują indywidualnie ćwiczenia nr 1–8 z sekcji „Sprawdź się”. Po ich wykonaniu nauczyciel omawia najlepsze rozwiązania problemów postawionych w zadaniach.

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”.

Praca domowa:

1. Uczniowie wykonują polecenie nr 3 z sekcji „Schemat interaktywny”.

Wskazówki metodyczne:

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