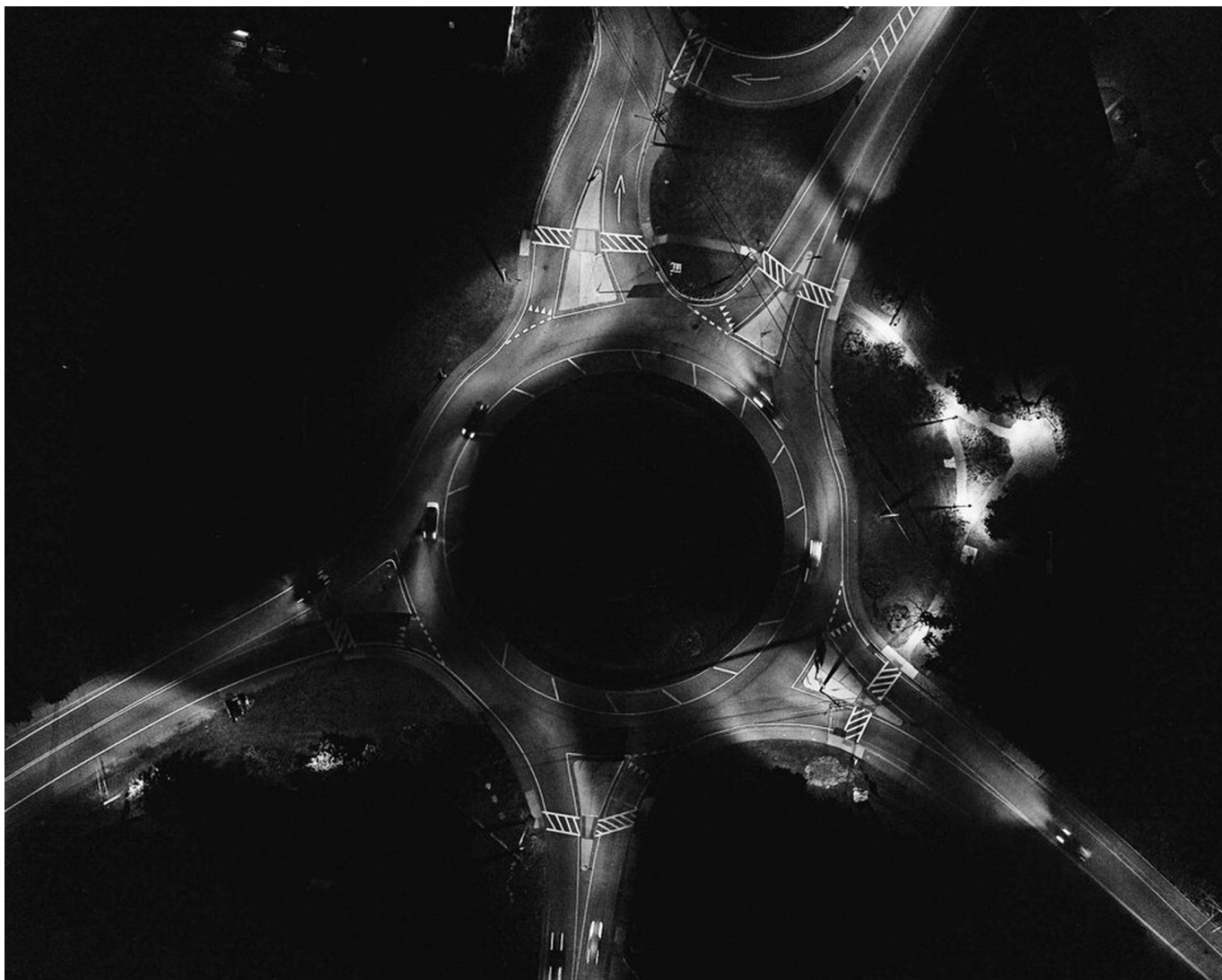




Instrukcja warunkowa – ćwiczenia

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



Instrukcja warunkowa – ćwiczenia

Źródło: Osman Rana, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

Wiesz już, czym charakteryzują się instrukcje warunkowe. Informacje na ten temat znajdziesz w e-materiale [Instrukcja warunkowa](#).

W tym e-materiale wykonasz ćwiczenia utrwalające twoją wiedzę.

Implementacje instrukcji warunkowych w wybranych językach programowania znajdziesz w e-materiałach:

- [Instrukcja warunkowa w języku C++](#),
- [Instrukcja warunkowa w języku Java](#),
- [Instrukcja warunkowa w języku Python](#).

Twoje cele

- Przeanalizujesz przykładowe instrukcje warunkowe.
- Scharakteryzujesz sytuacje, w których trzeba zastosować instrukcje warunkowe.
- Zbudujesz rozbudowane instrukcje warunkowe.
- Zapoznasz się z algorytmem jednoczesnego wyszukiwania minimum i maksimum w zbiorze liczb.

Przeczytaj

Instrukcja warunkowa – powtórzenie

Instrukcja warunkowa spełnia rolę zwrotnicy – pozwala wprowadzić w programie rozgałęzienia, które odpowiadają różnym zdarzeniom w czasie działania aplikacji oraz odpowiednio na nie reagować.

To, jak program zachowa się po dotarciu do rozgałęzienia, zależy od tego, czy spełniony został przekazany warunek logiczny. Przykładowo, system obsługi kart płatniczych odmawia zrealizowania transakcji, gdy przekroczono dzienny limit wartości operacji, bądź jeśli na koncie brakuje pieniędzy. W przeciwnym przypadku karta jest obciążana.

Trudno wyobrazić sobie jakikolwiek złożony algorytm, w którym nie zastosowano instrukcji warunkowej. Oto przykłady:

- przed uruchomieniem aplikacji **system operacyjny** musi sprawdzić, czy na dysku znajdują się wszystkie pliki niezbędne do jej działania; jeżeli ich nie ma, pojawia się komunikat o błędzie, a program nie startuje;
- podczas zakładania skrzynki e-mailowej specjalna **aplikacja internetowa** porównuje identyfikator podany przez użytkownika z nazwami innych kont i odmawia rejestracji w przypadku pojawienia się dublera;
- gdy pracujemy w systemie bankowości online, na bieżąco mierzony jest czas, który upłynął od momentu wykonania ostatniej operacji; jeżeli przekroczy on pewną wartość, następuje automatyczne zamknięcie sesji.

We wszystkich przedstawionych sytuacjach programista musi odwołać się do instrukcji warunkowej.

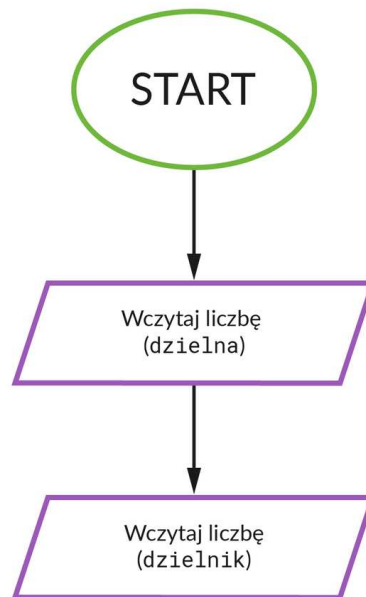
Wykrywanie próby dzielenia przez 0

Przygotujemy algorytm, który posłuży do skontrolowania poprawności liczb biorących udział w operacji dzielenia.

Zastanówmy się jednak, po co to w ogóle sprawdzać? Każdy wie, że dzielnik musi być różny od zera. Niestety, nie zawsze można to zagwarantować. Po pierwsze – i mniej ważne – użytkownik programu może celowo lub niechcący podać liczbę 0 jako dzielnik. O wiele częściej zdarza się coś innego: zerowa wartość dzielnika nie jest wpisywana przez człowieka, ale pojawia się jako efekt działania jakiegoś urządzenia – np. rozkalibrowanego czujnika wchodzącego w skład systemu kontrolnego w zakładzie przemysłowym.

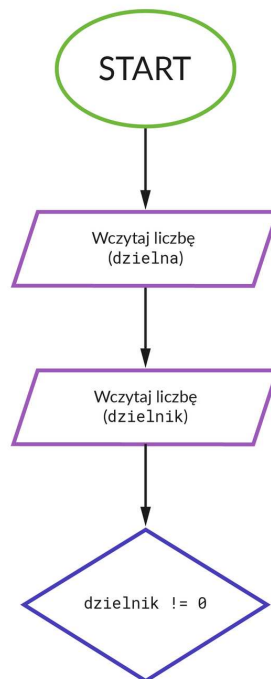
Programista musi takie zdarzenia przewidzieć i przygotować się na nie. Właśnie dlatego trzeba sprawdzać, czy dane spełniają określone kryteria nawet w banalnych przypadkach.

Pisząc algorytm, rozpoczynamy od wczytania dwóch wartości: dzielna oraz dzielnik.



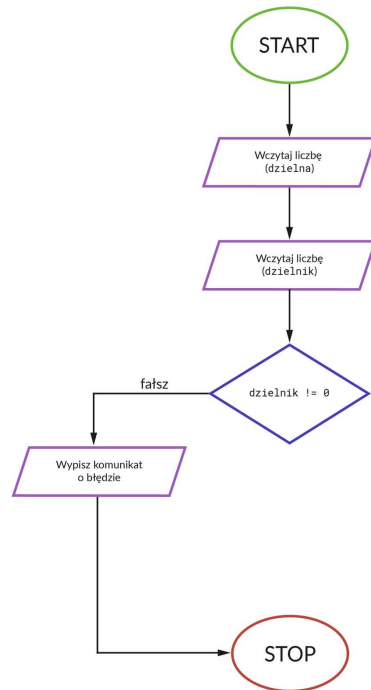
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Znając już dane, przystępujemy do ich sprawdzenia ([sprawdzenie poprawności danych](#)). Zmienna **dzielna** może być dowolną liczbą, zatem ją pomijamy. W przypadku zmiennej **dzielnik** należy sprawdzić, czy jej wartość różni się od zera. Używamy operatora **!=**, który oznacza brak równości (chcemy się upewnić, że wartość zmiennej **dzielnik** **nie jest równa** 0).



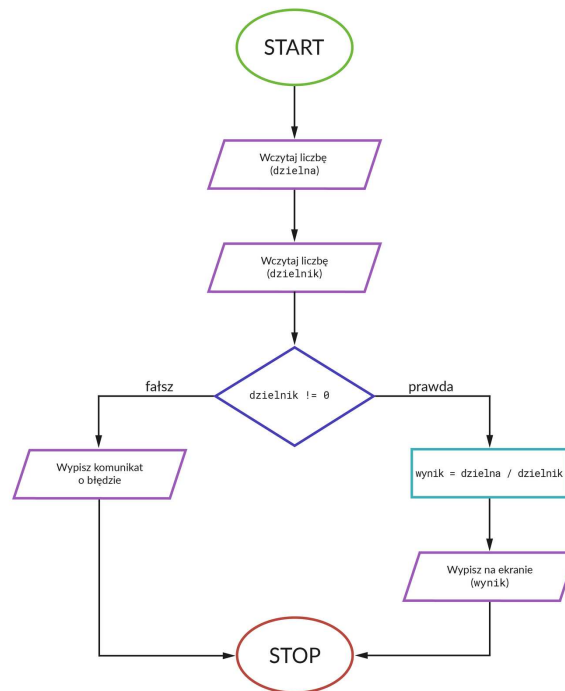
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Jeśli warunek nie jest spełniony (czyli wartość zmiennej **dzielnik** wynosi 0), wypisujemy komunikat o błędzie i kończymy działanie programu.



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

Gdy z kolei warunek jest spełniony (zmienna **dzielnik** różni się od zera), zmiennej **wynik** przypisujemy wartość ilorazu, wyświetlamy jej wartość na ekranie i kończymy działanie programu.



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

Logowanie w serwisie społecznościowym

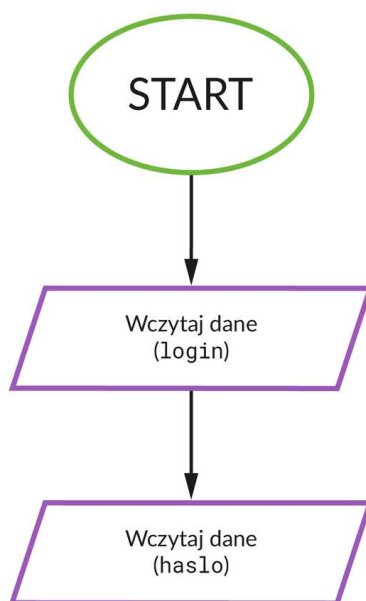
Przeanalizujmy teraz inny proces, w którym nie obędzie się bez instrukcji warunkowej. Mowa o mechanizmie logowania, stosowanym w serwisach internetowych. Specjalna aplikacja odczytuje wówczas informacje przekazane przez użytkownika, a następnie sprawdza, czy jest on właścicielem konta.

Informacje (login i hasło) wprowadzone podczas zakładania konta (rejestracji) przechowywane są w bazie danych. W trakcie logowania zapamiętany identyfikator wraz z hasłem są pobierane z bazy i porównywane z podanymi przez użytkownika informacjami. Jeśli dane są identyczne, użytkownik otrzymuje dostęp do konta. W przeciwnym wypadku pojawia się komunikat o błędzie.

Ciekawostka

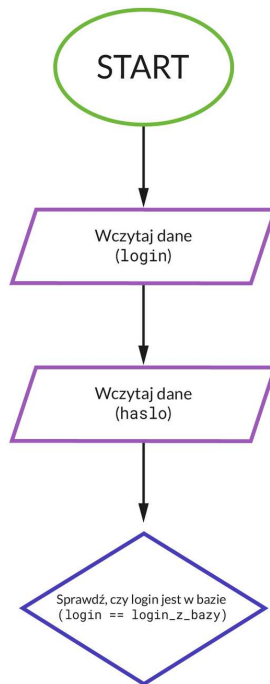
Hasła w bazie danych, ze względów bezpieczeństwa, nie są przechowywane w formie otwartego tekstu. Przed zapisem hasła do bazy zostaje na nim wywołana funkcja haszująca. Funkcja ta, dla tego samego ciągu wejściowego, np. tekstu, zwraca dokładnie tę samą wartość, która następnie zostaje zapisana w bazie. W celu sprawdzenia poprawności hasła wykonujemy na nim tę samą funkcję haszującą, a następnie porównujemy otrzymaną wartość z wartością przechowywaną w bazie danych. System wykonany w taki sposób jest bezpieczny, ponieważ sama znajomość funkcji haszującej oraz zakodowanego hasła nie pozwala na jego odszyfrowanie.

Proces logowania zaczynamy od wczytania informacji przekazanych przez użytkownika do dwóch zmiennych: `login` oraz `hasło`. W zwykłych warunkach należałoby przeprowadzić sprawdzenie poprawności danych. Dla uproszczenia załóżmy, że ją przeprowadziliśmy.



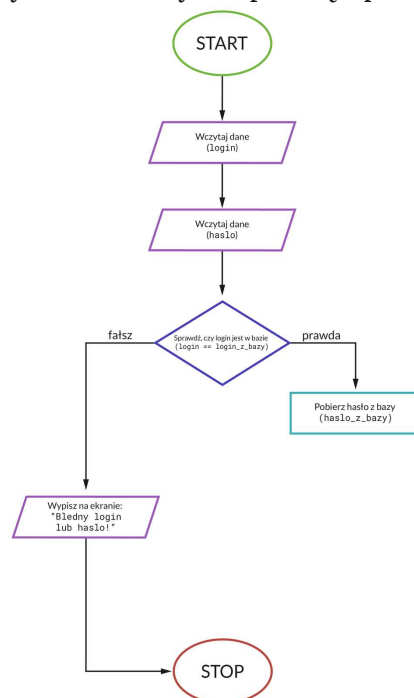
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Najpierw sprawdzimy, czy dany login jest w bazie. Robimy to za pomocą operatora `==`, który oznacza równość wyrażeń znajdujących się po obu stronach (sprawdzamy, czy podane wartości **są identyczne**).



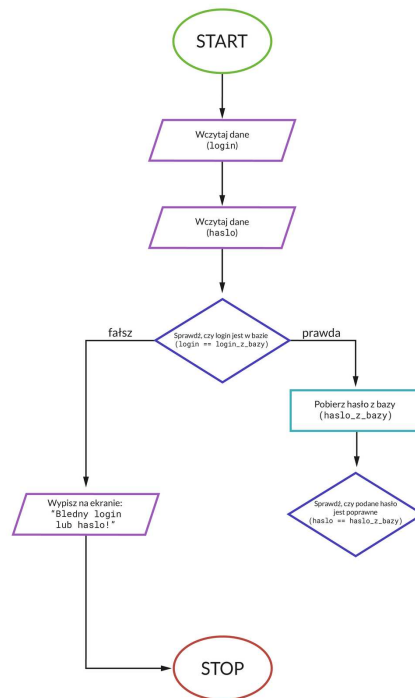
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Jeżeli identyfikator nie jest poprawny (login nie znajduje się w bazie), zostaje wyświetlony komunikat o błędzie, a program kończy działanie. W przeciwnym wypadku, gdy login okaże się zgodny z zapamiętanym, możemy rozpocząć procedurę sprawdzenia hasła.



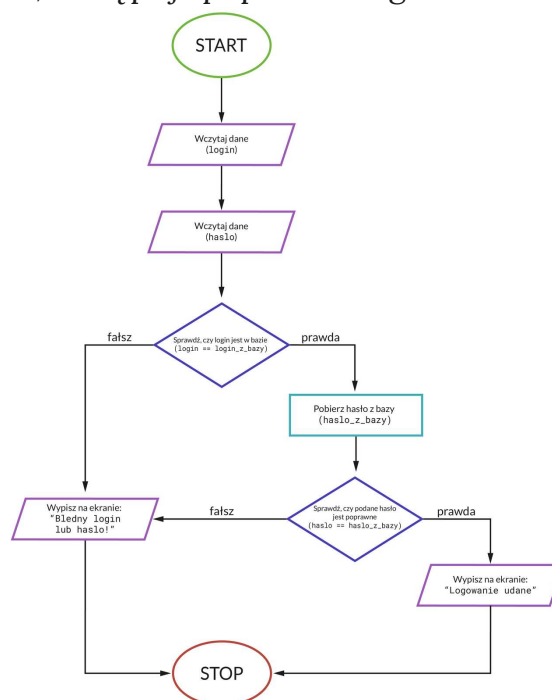
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Odczytujemy powiązane z nim hasło i porównujemy z hasłem podanym przez użytkownika.



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

Jeśli hasło nie jest poprawne, pojawia się komunikat o błędzie, a program kończy pracę. Gdy natomiast zgadza się z hasłem przechowywanym w bazie danych, użytkownik otrzymuje dostęp do serwisu, następuje poprawne logowanie.



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

Słownik

aplikacja internetowa

program uruchamiany z poziomu przeglądarki WWW (interfejs serwisu Facebook, system bankowości online itp.)

sprawdzenie poprawności danych

sprawdzanie, czy dane spełniają kryteria niezbędne do działania programu
(przykładowo, czy mają odpowiedni format lub wartość)

system operacyjny

oprogramowanie zarządzające zasobami komputera (procesorem, pamięcią operacyjną, pamięcią masową itd.) oraz uruchamianymi aplikacjami; odpowiada także za interakcję z użytkownikiem

Schemat interaktywny

Polecenie 1

Napisz algorytm, który w zbiorze n liczb podanych przez użytkownika znajdzie jednocześnie liczbę największą i najmniejszą. Instrukcji warunkowej użyjemy podczas porównywania wartości między sobą.

Do rozwiązania tego problemu użyjemy algorytmu znajdowania jednocześnie minimum i maksimum, używającego najmniejszej liczby porównań.

Ciekawostka

W algorytmie znajdowania obu wartości: minimum oraz maksimum w zbiorze potrzebujemy $2 \cdot (n - 1)$ porównań. Dzięki zastosowaniu algorytmu do jednoczesnego znajdowania wartości minimalnej i maksymalnej możemy znacząco ograniczyć liczbę porównań do: $\frac{3 \cdot n - 4}{2}$ dla parzystej liczby elementów oraz do $\frac{3 \cdot n - 3}{2}$ dla nieparzystej.

Specyfikacja:

Dane:

- n – liczba elementów w tablicy; liczba naturalna dodatnia;
- n liczb wprowadzonych przez użytkownika; zbiór liczb całkowitych.

Wynik:

- $\min$ – najmniejsza liczba w zbiorze liczb wprowadzonych przez użytkownika; liczba całkowita;
- $\max$ – największa liczba w zbiorze liczb wprowadzonych przez użytkownika; liczba całkowita.



Kliknij, aby uruchomić w trybie pełnoekranowym.

Polecenie 2

Zaprojektujmy algorytm do sprawdzania stopnia *K-gładkości* danej liczby `liczba`.

Liczba naturalna jest *K-gładka*, jeżeli wszystkie dzielniki pierwsze są nie większe niż *K*. W celu określenia stopnia *K-gładkości* liczby, musimy rozłożyć ją na czynniki pierwsze, a następnie określić, który z tych czynników jest największy.

Przykłady:

- $26 = 2 \cdot 13$ – jej stopień *K-gładkości* wynosi 13 (największy dzielnik pierwszy);
- $17 = 17$ – jest to liczba pierwsza, dlatego jej stopień *K-gładkości* wynosi tyle ile jej wartość, czyli 17;
- $700 = 2 \cdot 2 \cdot 5 \cdot 5 \cdot 7$ – stopień *K-gładkości* wynosi 7.

Specyfikacja:

Dane:

- `liczba` – liczba naturalna

Wynik:

Program wypisuje `K-liczba`, stopień *K-gładkości* liczby.



Kliknij, aby uruchomić w trybie pełnoekranowym.

Polecenie 3

Przygotuj algorytm, który losuje trzy liczby naturalne dodatnie z zakresu od 1 do 100 włącznie – wiek trzech osób, a następnie wyznacza najstarszą osobę (jeżeli osoby będą w tym samym wieku, wybierz pierwszą z nich).

Specyfikacja:

Dane:

- osoba1 – wylosowany wiek pierwszej osoby; liczba naturalna z zakresu $\langle 1, 100 \rangle$;
- osoba2 – wylosowany wiek drugiej osoby; liczba naturalna z zakresu $\langle 1, 100 \rangle$;
- osoba3 – wylosowany wiek trzeciej osoby; liczba naturalna z zakresu $\langle 1, 100 \rangle$;

Wynik:

Program najpierw wyświetla wiek każdej osoby, a następnie podaje, która z osób jest najstarsza.

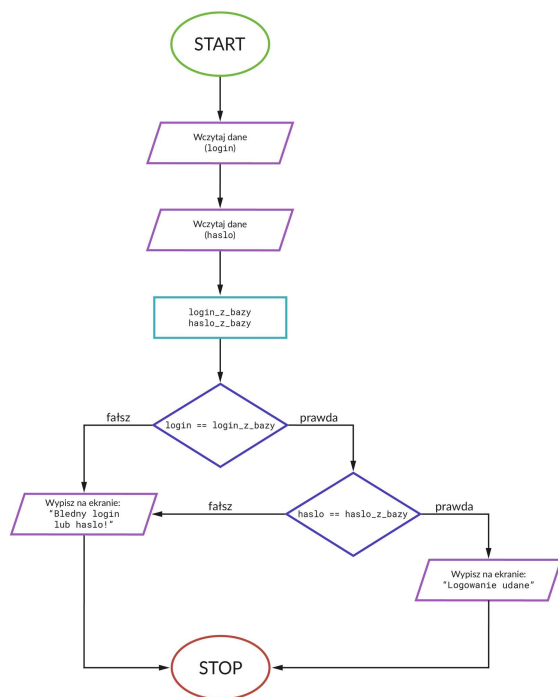
Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, jaki będzie rezultat zastosowania przedstawionego algorytmu logowania, jeśli podany login jest poprawny, a podane hasło niepoprawne.



- ☐ Zostanie wyświetlona prośba o ponowne wprowadzenie danych.
- ☐ Pojawi się komunikat „Błędny login lub hasło!”.
- ☐ Pojawi się komunikat „Logowanie udane”.

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

Ćwiczenie 2



Wskaż, jakie mogą być konsekwencje sytuacji, w której użytkownik poda niepoprawne informacje, a program nie będzie mieć wbudowanej procedury walidacji danych.

- ☐ Nie będzie żadnych negatywnych konsekwencji.
- ☐ System operacyjny przejmie obowiązki walidacji danych.
- ☐ Program może zacząć działać nieprzewidywalnie, co może mieć negatywny wpływ na pozostałą część algorytmu.

Ćwiczenie 3



Wskaż poprawną odpowiedź.

Czy w programie obsługującym płatności elektroniczne może być wskazane użycie instrukcji warunkowych?

- ☐ nie
- ☐ tak

Ćwiczenie 4



Wskaż, w której sytuacji uzasadnione jest użycie instrukcji warunkowej.

- ☐ rozpoczęcie algorytmu
- ☐ wczytanie danych od użytkownika (bez sprawdzania poprawności danych)
- ☐ porównanie wartości dwóch liczb

Materiał źródłowy do ćwiczeń nr 5-6.

```
1 jeżeli a > b:
2     t ← b
3     b ← a
4     a ← t
5 jeżeli b > a:
6     t ← b
7     b ← a
```



```
8      a ← t
9      jeżeli a > b oraz c < b:
10         wypisz a - c
11      w przeciwnym razie:
12         jeżeli c > b oraz a > b:
13             wypisz a + b
14         w przeciwnym razie:
15             wypisz 2 * b
16      w przeciwnym razie:
17         wypisz a + b + c
```

Ćwiczenie 5



Wskaż, co wypisze algorytm, jeżeli początkowe wartości zmiennych będą wynosić kolejno: a = 10, b = 5, c = 7.

☐ 15

☐ 10

☐ nic

☐ 22

Ćwiczenie 6



Uzupełnij tekst.

Algorytm wypisze , jeżeli początkowe wartości zmiennych będą wynosić kolejno: a = 3, b = 2, c = 1.

Ćwiczenie 7



Napisz program, który wyznaczy miejsce zerowe funkcji liniowej $f(x) = ax + b$. Działanie swojego programu przetestuj dla funkcji $f(x) = 5x + 8$.

Specyfikacja:

Dane:

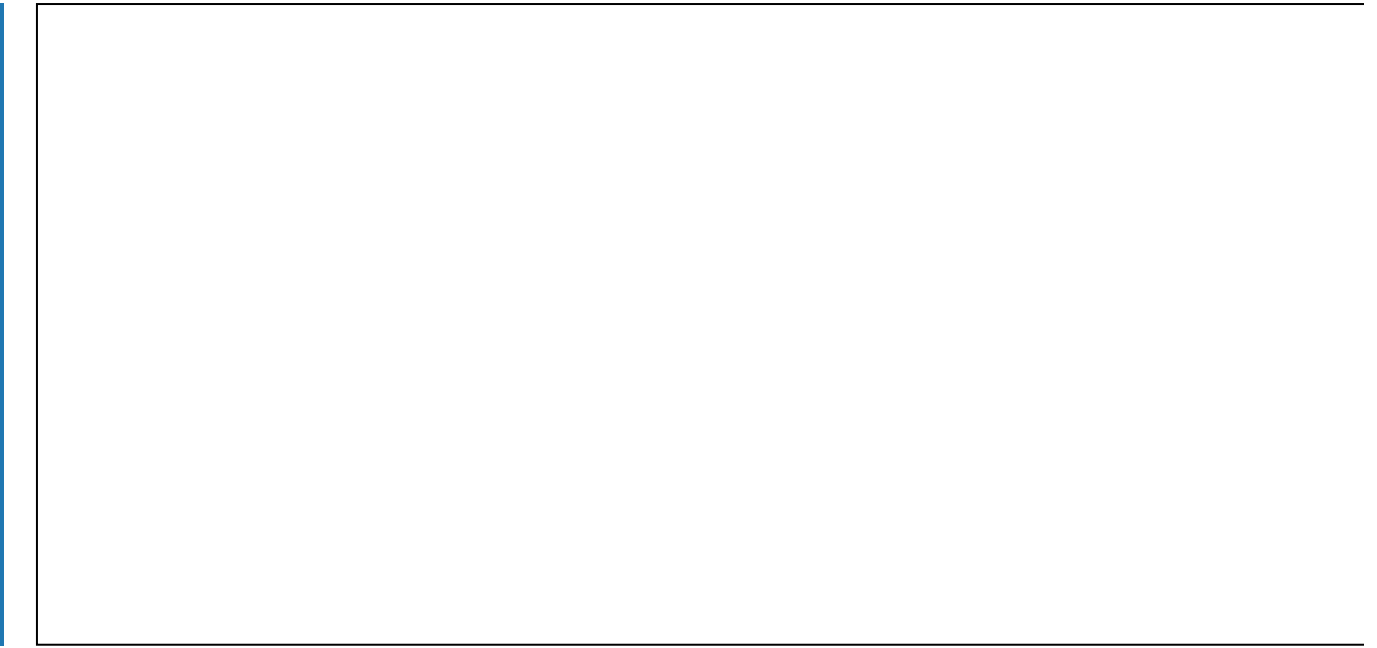
- a – współczynnik kierunkowy prostej; liczba rzeczywista
- b – wyraz wolny; liczba rzeczywista

Wynik:

Na standardowym wyjściu program wypisuje komunikat informujący czy występuje miejsce zerowe funkcji oraz wskazuje jej wartość (jeżeli istnieje).

1

1





Pan Jan jest podróżnikiem. Szczególnie ukochał: Londyn, Nowy Jork, Tokio. Pan Jan chciałby wiedzieć, jaka godzina jest w wybranym mieście w zależności od czasu w Warszawie. Napisz program, który dla podanej godziny w Warszawie pokaże czas w wybranym ulubionym mieście pana Jana.

Na potrzeby zadania założmy, że kraje, w których znajdują się poszczególne miasta, nie zmieniają czasu na letni. Przyjmujemy dla każdego miejsca czas standardowy.

Działanie swojego programu przetestuj, sprawdzając, która godzina jest w Nowym Jorku, jeśli w Warszawie jest 14:24.

Specyfikacja:

Dane:

- *godzina* – godzina w Warszawie; liczba naturalna w zakresie $\langle 0, 23 \rangle$
- *minuta* – minuta w Warszawie; liczba naturalna w zakresie $\langle 0, 59 \rangle$
- *miasto* – miasto, dla którego program ma policzyć czas (dozwolone wartości: Londyn, Nowy Jork, Tokio); ciąg znaków

Wynik:

Na standardowym wyjściu program prezentuje czas dla wskazanego miasta.

Wskazówka:

- Warszawa znajduje się w strefie czasowej GMT+1;
- Londyn znajduje się w strefie czasowej GMT+0;
- Nowy Jork znajduje się w strefie czasowej GMT-5;

- Tokio znajduje się w strefie czasowej GMT+9.

Przykładowe wyjście:

1 Czas w mieście Tokio: 4:37

1

1

Dla nauczyciela

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

Przedmiot: Informatyka

Temat: Instrukcja warunkowa – ćwiczenia

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:

- 3) objaśnia dobrany algorytm, uzasadnia poprawność rozwiązania na wybranych przykładach danych i ocenia jego efektywność;

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

Zakres podstawowy. Uczeń:

1) projektuje i programuje rozwiązania problemów z różnych dziedzin, stosuje przy tym: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje z parametrami i bez parametrów, testuje poprawność programów dla różnych danych; w szczególności programuje algorytmy z punktu I.2);

2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

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

3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

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 przykładowe instrukcje warunkowe.
- Scharakteryzujesz sytuacje, w których trzeba zastosować instrukcje warunkowe.
- Zbudujesz rozbudowane instrukcje warunkowe.
- Zapoznasz się z algorytmem jednoczesnego wyszukiwania minimum i maksimum w zbiorze liczb.

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.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Instrukcja warunkowa – ćwiczenia”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Nauczyciel inicjuje rozmowę wprowadzającą w temat lekcji. Przedstawia cele zajęć oraz kryteria sukcesu.

Faza realizacyjna:

1. Uczniowie analizują przykład z sekcji „Przeczytaj” i powtarzają zaprezentowane rozwiązanie na swoim komputerze.
2. **Praca z multimedium.** Nauczyciel wyświetla zawartość sekcji „Schemat interaktywny”, wybrany uczeń czyta treść polecenia nr 1: „Spróbujmy znaleźć w zbiorze liczb dwa elementy: największy i najmniejszy. Instrukcji warunkowej użyjemy podczas sprawdzania, czy kontrolowany obiekt jest większy (bądź mniejszy) od wartości, którą uważaliśmy dotychczas za największą (albo najmniejszą). Musimy nieco wybiec naprzód i opowiedzieć o tablicach oraz pętlach. Nie omawialiśmy jeszcze tych zagadnień, ale obecnie wystarczą nam informacje podstawowe. Tablica jest zbiorem zmiennych tego samego typu (np. liczb całkowitych). Pętla służy z kolei do wielokrotnego wykonywania tych samych czynności – przykładowo, do sprawdzania kolejnych elementów umieszczonych w tablicy.” i omawia przykładowe rozwiązanie postawionego problemu.
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. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują ćwiczenie nr 2 „W poniższym schemacie przygotuj algorytm, który losuje wiek 3 osób, a następnie wyznacza jedną najstarszą.” z sekcji „Schemat interaktywny”.

2. Uczniowie zapisują schemat blokowy w wybranym przez siebie języku programowania.

Wskazówki metodyczne:

- Nauczyciel może wykorzystać multimedium w sekcji „Schemat interaktywny” do pracy przed lekcją. Uczniowie zapoznają się z jego treścią i przygotowują do pracy na zajęciach w ten sposób, żeby móc samodzielnie rozwiązać zadania dołączone do e-materiału „Instrukcja warunkowa – ćwiczenia”.