Instrukcje warunkowe

Conditional statements

Daria Wawryk[1], Elżbieta Turkiewicz[1], Jakub Lachowicz[1]

STRESZCZENIE: Tematem przewodnim artykułu są instrukcje warunkowe. Poniżej zostaną omówione poszczególne typy instrukcji warunkowych, ich zastosowań oraz składni. Zostaną również przedstawione schematy ideowe przedstawiające zasadę działania owych instrukcji.

Słowa kluczowe: instrukcja warunkowe, instrukcja if, instrukcja if-then, instrukcja else-if, instrukcja switch, dopasowanie do wzorca, pętla while, pętla until, instrukcja wyboru, pętla, wyrażenia symboliczne, instrukcja wyboru, warunek.

ABSTRACT:
This article focuses on conditional statements. Their different types, syntax and applications will be discussed. There will also be presented diagrams ilustrating their principle of operation.

Keywords:
conditional statements, if statments, if-then statements, else-if statement, pattern matching,while loop, until loop, ternary operator, symbolic statements, condition.

1.WSTĘP

Instrukcja warunkowa jest składową języka programowania, za pomocą której umożliwione jest wykonywanie różnych obliczeń zależnie od tego, czy definiowane przez programistę wyrażenie jest prawdziwe czy fałszywe. Współczesne komputery cechują się możliwością warunkowego decydowania o tym, jaki krok zostanie wykonany w dalszej kolejności. Każdy model obliczeniowy zdolny do wykonywania algorytmów musi zawierać instrukcję warunkową [1]. W imperatywnych językach programowania używa się terminu instrukcja warunkowa, podczas gdy w programowaniu funkcyjnym preferowane są nazwy wyrażenie warunkowe lub konstrukcja warunkowa, gdyż posiadają one inną zasadę działania. Instrukcja warunkowa sprawia, że pewne instrukcje wykonają się przy spełnionym bądź nie spełnionym warunku. Dzięki temu rozwiązaniu program nie wykonuje zawsze ciągu instrukcji w linearny sposób lecz w zależności od napotkanej sytuacji może wybrać właściwą drogę. Dzięki niej komputery są w stanie rozwiązywać różne problemy na wielu poziomach skomplikowania. Przykładowa instrukcja warunkowa napisana w języku C++ wygląda następująco:

if(warunek) instrukcja$_X$; else instrukcja$_Y$;
warunek jest wyrażeniem logicznym, które może przyjmować wartość false albo true. W zależności od jego wartości wykonywana jest jedna z instrukcji:
warunek = true ─ wykonywana jest instrukcja$_X$, a instrukcja$_Y$ zostaje pominięta
warunek = false ─ pominięta zostanie instrukcja$_X$, a wykonana instrukcja$_Y$. [11]

2.INSTRUKCJE WARUNKOWE, INSTRUKCJE WYBORU, PĘTLE - ZASADY DZIAŁANIA

Spośród instrukcji warunkowych wyróżniamy:

- Instrukcję warunkową if-then
- Instrukcję warunkową else-if
- Operator warunkowy
- Instrukcję wyboru(instrukcję switch-case)
- Dopasowanie do wzorca

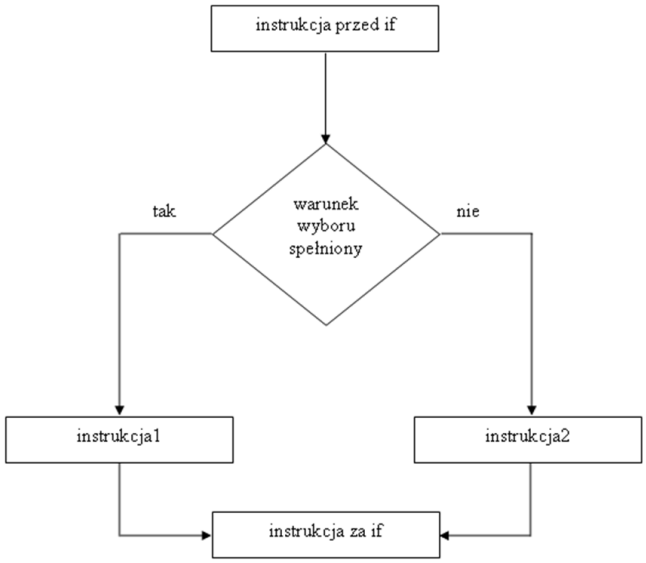
1.Wrocławska Wyższa Szkoła informatyki Stosowanej “Horyzont” Wejherowska 28, 54-239 Wrocław elzbietaturkiewicz0@gmail.com,

W dalszej części publikacji zostaną dokładnie omówione poszczególne typy instrukcji warunkowych. Natomiast w ostatniej części została omówiona jedynie pętla warunkowa.

If-Else, If-Then oraz Else- If

Podstawowym rodzajem instrukcji warunkowej jest If-Then. Wykorzystywana jest w większości języków programowania w celu warunkowego wykonania określonego bloku kodu, a w przypadku gdy warunek jest niespełniony, odpowiada za wykonanie bloku alternatywnego. W różnych językach programowania instrukcje różnią się nieznacznie składnią, natomiast schemat ogólny zawsze wygląda następująco:

Pierwszym krokiem jest analiza warunku podanego w postaci wyrażenia logicznego. Następnie, w przypadku gdy wynikiem jest true, wykonywany jest właściwy blok kodu, a gdy wynikiem jest false – alternatywny blok kodu.



Ryc. 1Schemat działania instrukcji warunkowej If-Else
Fig. 1 If- Else statement principle of operation

W wielu językach programowania istnieje możliwość zdefiniowania więcej niż jednego warunku sprawdzającego przy użyciu opcjonalnego bloku else-if. Dzięki temu możemy rozszerzyć podstawową instrukcję warunkową if, przy użyciu słowa else. Słowo kluczowe else jest jednoznaczne z ”w przeciwnym wypadku”, co oznacza, że jeśli warunek nie zostanie spełniony wykonany zostanie inny kod. Generalny zapis instrukcji warunkowej, który zawiera instrukcje else wyglądać będzie następująco: if (…) … else… , gdzie trzy ostatnie kropki oznaczają instrukcje które mają zostać wykonane w przypadku, gdy warunek if nie został spełniony.
Przykład działania instrukcji warunkowej if-then:

if warunek x **then**
  pierwszy blok kodu
else-if warunek y **then**
  drugi blok kodu
else-if warunek z **then**
  trzeci blok kodu
else
  alternatywny blok kodu
end if

W takim wypadku warunki analizowane są kolejno aż do momentu, gdy któryś z nich nie da wartości true – wtedy zostaje wykonany przypisany mu blok kodu. Gdy wystąpi sytuacja, w której żaden z warunków nie okaże się prawdziwy, wykonany zostanie blok alternatywny. Ilość prawdziwych warunków nie ma znaczenia, ponieważ zawsze wykona się tylko pierwszy z nich, a pozostałe zostaną pominięte.

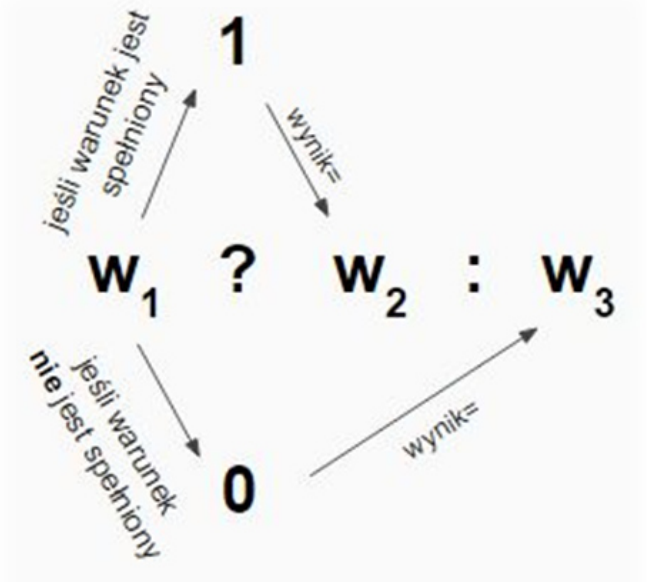
Istnieją języki programowania w których występuje konstrukcja alternatywna do opisanej powyżej instrukcji warunkowej, natomiast jej działanie jest odwrotne do instrukcji warunkowej if-else. Polega ona na tym, że pierwszy blok kodu jest wykonywany gdy pierwszy warunek jest niespełniony, a w przeciwnym razie wykonywane są inne warunki fraz else. Jest to więc równoważnik konstrukcji if not (warunek) then (instrukcja). Aby odróżnić tę konstrukcję od pierwotnej strosuje się inne słowo kluczowe, np. Unless. [3]

Operator warunkowy

Wyrażenie warunkowe różni się od instrukcji warunkowej If-Then tym, że wykonany blok kodu musi zwrócić jakąś wartość, która jednocześnie jest wynikiem całego wyrażenia. Wyrażenia warunkowe często występują w funkcyjnych językach programowania. W programowaniu jest konstrukcją języka programowania, w których odróżnia się wyrażenia od instrukcji, będącą formą instrukcji warunkowej wyrażoną za pomocą operatora trójargumentowego; bywa ona instrukcją wyrażeniową. Umożliwia ona sprawdzenie warunku na poziomie wyrażenia, co w pewnym stopniu zaciera rozróżnienie między wyrażeniami a instrukcjami, dzięki czemu przy jej rozsądnym używaniu kod źródłowy może zyskać na zwięzłości i prostocie.

W języku C/ C++ operator ten ma postać:
warunek ? wyrażenie1 : wyrażenie2

Celem działania operatora jest sprawdzenie wartości logicznej wyrażenia i zwrócenie jej na podstawie jednej z dywiz: wyrażenie 1 gdy warunek jest prawdziwy oraz wyrazenie 2 gdy nie jest, gdy wyrażenie nie zostało zwrócone, zwykle wartość tego wyrażenia jest niepoliczalna.



Ryc. 2 Schemat sprawdzania warunku [12]
Fig. 2 Ternary operator working principle

Instrukcja wyboru

Jest to instrukcja decyzyjna, czyli instrukcja w danym języku programowania, dająca możliwość wyboru spośród wielu opcji. Składnia instrukcji wyboru zależna jest od języka programowania, lecz istnieją charakterystyczne elementy takie jak: nagłówek, ciało, koniec.

nagłówek instrukcji wyboru:
- słowo kluczowe rozpoczynające instrukcję (np. case, select, switch)
- opcjonalnie wyrażenie, na podstawie którego następuje wybór
- słowo łączące (np. of, do)

ciało instrukcji wyboru:
- kolejne instrukcje podlegające selekcji
- opcjonalnie poprzedzone frazami zawierającymi wartości porównywane z wyrażeniem z nagłówka instrukcji
- opcjonalnie poprzedzone słowem kluczowym (np. case, when)
- słowo lub symbol łącznika (np. ”do”)
- opcjonalnie fraza domyślna wykonywana, gdy żadna z fraz nie spełni warunku (np. else, otherwise, other, default, when else)

koniec instrukcji wyboru – słowo zamykające blok (np. end, end case, end select itp.)

Poniżej zaprezentowany został podstawowy kod zawierający instrukcję wyboru switch-case. Oraz jego omówienie

```
switch( <wyrażenie> )
{
case <wartośćA>:
instrukcjaX;
instrukcjaY;
 //...
break;
case <wartośćB>:
instrukcjaX;
instrukcjaY;
//...
break;
//...
default:
instrukcjaX;
instrukcjaY;
//...

}
```

Konstrukcja switch-case umożliwia wybór więcej niż jednej opcji. Instrukcja switch oblicza wartość <wyrażenia> i przypisuje go do jednej z podanych opcji. Wszystkie opcje muszą być zawarte w bloku. Po słowie kluczowym case należy podać <wartość>, a następnie dwukropek. Po dwukropku występuje instrukcja, która nie musi być zawarta w nowym bloku.
Gdy wartość wyrażenia będzie odpowiadała jednej z wartości, to wykonywane są wszystkie instrukcje występujące pomiędzy nią, a kolejnym słowem kluczowym-break.
W wypadku gdy żadna z wartości nie będzie pasowała do wartości wyrażenia, wdrożone zostaną instrukcje znajdujące się po słowie kluczowym default.
Główną różnicą między instrukcją warunkowa if a switch jest to, że w regularnej instrukcji if możemy określić dokładnie co ma się wydarzyć w zależności od stanu jednej lub kilku zmiennych. Instrukcja if daje nam pełną kontrolę nad przebiegiem programu. W języku C++ jest jednak dostępna również instrukcja wielokrotnego wyboru switch. W przypadku niej możemy wykonywać decyzje tylko na podstawie wartości jednej zmiennej. Możliwości instrukcji switch są nieporównywalnie mniejsze, jednak używanie jej w niektórych przypadkach jest znacznie korzystniejsze dla szybkości działania programu i estetyki kodu niż użycie instrukcji if. Aby poprawnie używać instrukcji switch-case należy zapoznać się z pewnym warunkiem, który określa, jakie dozwolone typy danych można używać. Tak więc dozwolonymi typami danych w instrukcji switch są liczby całkowite. Oznacza to, że możemy użyć tylko zmiennych, które są typów takich jak: char, short, int, long, long long. Do tego dochodzi również typ wyliczeniowy enum [12].

Dopasowanie do wzorca

Jest to operacja, w której pewne wyrażenie sprawdza się ze wzorcem, w którym może znajdować się jedno lub więcej "wolnych miejsc". W wyniku, o ile nastąpiło dopasowanie, otrzymuje się listę wyrażeń, które dopasowały się do wolnych miejsc wzorca.
Dopasowywanie do wzorca jest bardzo ekspresywną techniką programistyczną. Dwa najpopularniejsze systemy to:

- wyrażenia regularne
- wzorce symboliczne

Wyrażenie regularne

W większości nowych języków wyrażeń regularnych można używać jako wzorców, np (Perl):

```
{
   # pasuje do wzorca, kolejne liczby w $1 $2 $3 $4
} else {
   # nie pasuje do wzorca
}
```

Alternatywa wyżej przedstawionego kodu w języku C musiałaby wykorzystywać bibliotekę, która umożliwiłaby dopasowanie do wzorców (np. PCRE), w przeciwnym wypadku istniały by duże szanse na wystąpienie błędów.

Wyrażenia symboliczne

Wzorce symboliczne wykorzystywane są w językach funkcyjnych, są to termy ze zmiennymi, które dopasowują się do danego wyrażenia przez unifikację. Dostępny jest również specjalny symbol uniwersalny pasujący do każdego obiektu; w wielu językach to znak podkreślenia _.
Na przykład (w Ocamlu) zamiana wartości liczbowej na ciąg znaków z użyciem dopasowania można zrealizować następująco:

```
match liczba with
    0 -> "zero"
|   1 -> "jeden"
|   2 -> "dwa"
|   _ -> "inna liczba"
```

To samo można uzyskać instrukcją warunkową:

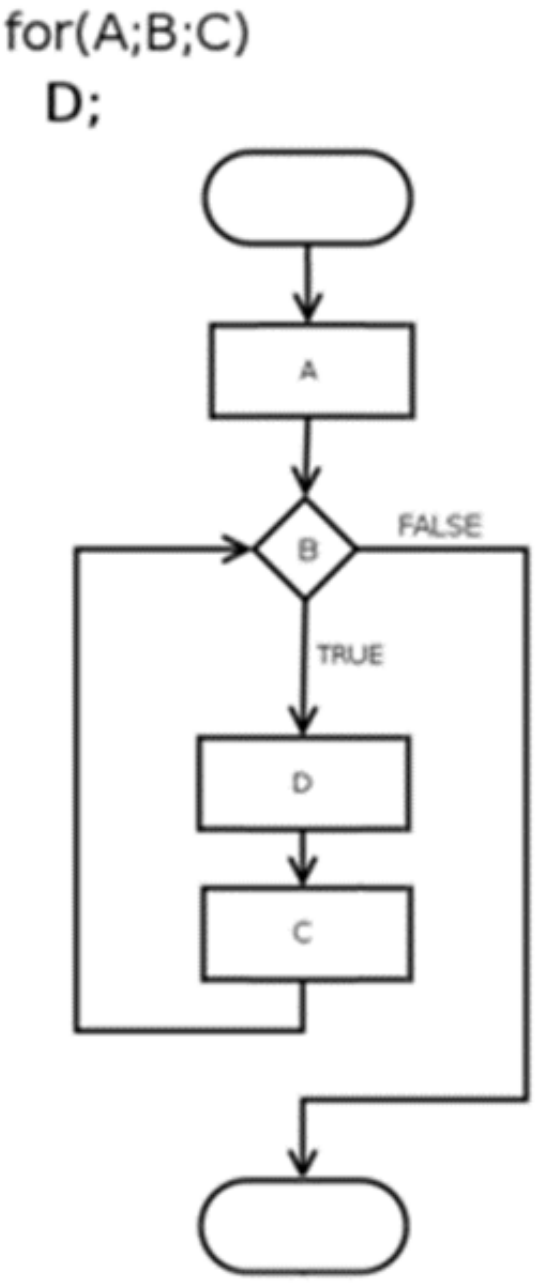
```
if liczba = 0 then "zero" else
if liczba = 1 then "jeden" else
if liczba = 2 then "dwa" else "inna liczba"
```

Jednak ten zapis jest bardziej rozwlekły i mniej czytelny.
Jeszcze inny przykład wykorzystujący konstruktory list; zapis x::lista oznacza doklejenie na początek listy elementu x, wynikiem jest nowa lista.

```
match zmienna with
        [] -> "Lista jest pusta"
|     x::[] -> "Lista ma jeden element: " ^ x
|   x::y::[] -> "Lista ma dwa elementy: " ^ x ^ " i " ^ y
| x::y::z::[] -> "Lista ma trzy elementy: " ^ x ^ ", " ^ y ^" i " ^ z
| x::y::z::_  -> "Lista ma więcej niż trzy elementy: " ^ x ^ ", " ^ y ^", " ^ z ^ "..."
```

Pętla

Pętla jest to jedna z trzech podstawowych konstrukcji używanych w programowaniu strukturalnym. Daje ona możliwość wykonywania ciągu instrukcji w sposób cykliczny określoną liczbę razy aż do momentu, kiedy zajdą pewne warunki dla każdego elementu, lub w nieskończoność.



Ryc. 3 schemat działania pętli for [16]
Fig. 3 For loop principle of operation diagram

Pętla warunkowa

Pętla warunkowa jest to rodzaj pętli, w której programista definiuje warunek, od którego zależne jest wykonanie kolejnej iteracji. Warunek zawarty w definiowanej pętli jest pewnym wyrażeniem, które zwraca wartość typu logicznego, (np. Pascal, Visual Basic, VBA) natomiast nie w każdym jezyku programowania składnia przewiduje taki typ danych. W niektórych jezyka definiowane są wyrażenia które zwracają wartość innego typu. Następnie zostaje ona poddana interpretacji, np. wartość 0 może być jednoznaczna z wartością false typu logicznego, a pozostałe z wartością true. Zależnie od tego, czy wartość logiczna uzyskana w wyniku ewaluacji wyrażenia przedstawiajace-

go warunek jest równa wartości logicznej true (prawda), czy false (fałsz), wykonywanie pętli jest kontynuowane bądź przerywane [8] [9] [11] [14].
Podsumowując, pętla na początku definiowana jest przy użyciu określonego wyrażenia, za pomocą którego określone zostaje, czy nastąpi przejście do kolejnej iteracji, w przeciwnym wypadku nastąpi zakończenie wykonywania pętli, czego konsekwencją będzie przejście do kolejnej instrukcji znajdującej się już poza pętlą. W zależności od składni konkretnego jezyka programowania. Powszechnym jest zapis wyrażeń kontrolnych analogicznie do zapisu tych wyrażeń dla instrukcji warunkowej. Ogromną rolę w tym wypadku odgrywają operatory porównawcze, mimo że warunek może zostać wyrażony również w inny sposób, np. w postaci wywołania funkcji zwracającej wartość logiczną bądź jako identyfikator zmiennej logicznej, której wcześniej przypisano rezultat ewaluacji wyrażenia warunkowego. W programowaniu bardzo użyteczna jest również dostępność operatorów logicznych dająca możliwość budowania warunków złożonych z kilku warunków łącznych lub alternatywnych (zdaża się, że w konkretnym języku mogą być dostępne inne operatory logiczne, np. alternatywy wykluczającej czy implikacji).
Kluczową rolę odgrywa możliwść decydowania o kontynuacji lub zaprzestaniu wykonywania pętli. Możliwość sprawdzenia warunku może występować na różnych etapach działania pętli. Sprawdzanie może odbywać się na początku pętli, tzn. przed wykonaniem pierwszej instrukcji zawartej w bloku danej pętli. Inną możliwością jest weryfikacja warunku wewnątrz pętli, tzn. w jej bloku, pomiędzy wykonaniem części instrukcji, lecz przed zakończeniem wykonywania pozostałych). Ostatnią możliwością jest sprawdzenie warunku na końcu pętli – po wykonaniu wszystkich instrukcji, które są zawarte w definiowanej pętli).
W sytuacji, kiedy warunek sprawdzany jest na początku pętli, istnieje duże prawdopodobieństwo, że instrukcja zawarta w pętli nigdy się nie wykona. Taka możliwość istnieje gdy przy pierwszym wykonaniu warunek nie będzie spełniony. Zupełnie inaczej jest w przypadku, kiedy warunek sprawdzany jest na końcu pętli, wtedy instrukcje muszą wykonać się chociaż jeden raz. Sytuacją łączącą obie te możliwości jest warunek sprawdzany wewnątrz pętli. Oznacza to, że instrukcje zapisane przed sprawdzeniem warunku zostaną wykonane minimum raz, a instrukcje występujące po warunku sprawdzającym mogą się w ogóle nie wykonać.

Podstawowymi konstrukcjami składowymi są :
- while
- until
[12] [13] [14] [15]

Zastosowania i budowa przedstawionych warunków

Warunek **while** wykorzystuje się w sytuacjach, kiedy niezbędne jest wykonywanie danej operacji, dopóki nie zostanie spełniony warunek. Warunek while ma następującą składnię:

while (warunekKoncowy)
  lista_instrukcji

Pętla while podobnie jak pętla **for** oraz jak pętla **do while** umożliwia powtarzanie instrukcji tak długo jak warunek końcowy jest spełniony, a warunek jest zapisywany na samym początku pętli dlatego, że warunek jest sprawdzany zanim cokolwiek innego zostanie wykonane i w ten sposób jeśli warunek nie jest spełniony, nic nie zostanie wykonane.
Brak jest nawiasów klamrowych, bowiem nie są one wymagane, kiedy chcemy wykonać tylko jedną instrukcję. Należy również zauważyć to, że po nawiasach okrągłych znajdujących się za słowem while nie ma średnika.
Innym przypadkiem jest warunek **Until**, gdzie słowo kluczowe w różnych językach ma różne implikacje:
- warunek który określa, iż pętla będzie powtarzana, dopóki warunek nie stanie się spełniony,
- warunek, który jest sprawdzany na końcu pętli (choć zapis warunku znajduje się na jej początku w nagłówku definiującym), a pętla jest powtarzana jeżeli warunek jest spełniony.

Instrukcja repeat ... until (*nazywana pętlą "powtarzaj"*)
Instrukcja ta wykonuje cyklicznie inne instrukcje zawarte pomiędzy słowami repeat i until do momentu gdy *wyrażenie* znajdujące się za słowem until nie przyjmie wartości *prawda* (czyli *true*).
Efekt zastosowania pętli repeat jest bardzo podobny do działania pętli while. Pętla ta także może być wykonywana ogromną liczbę razy. Jedyna różnica polega na tym, że w pętli repeat warunek zakończenia sprawdzany jest dopiero po wykonaniu instrukcji. Oznacza to, że pętla repeat zawsze będzie wykonana co najmniej raz. Dopiero po tej iteracji program sprawdzi, czy można zakończyć działanie pętli. W przypadku pętli while warunek jest sprawdzany bezpośrednio przed jej wykonaniem, co w rezultacie może spowodować, że taka pętla nigdy nie zostanie wykonana.

Budowa pętli repeat jest następująca:
repeat
  { instrukcje do wykonania }
until { warunek zakończenia }
[16]

Warunki w maszynach Turinga

Jednym z wybitnych przykładów zastosowania warunków w praktyce jest użycie ich w maszynie Turinga. Maszyna ta jest abstrakcyjnym modelem komputera służącego do wykonywania algorytmów. Jej działanie polega na tym, że na nieskończenie długiej taśmie podzielonej na pola zapisuje się dane. Taśma może być nieskończona jednostronnie lub obustronnie. Każde pole może znajdować się w jednym z N stanów. Maszyna zawsze jest ustawiona nad jednym z pól i znajduje się w jednym z M stanów. Zależnie od

kombinacji stanu maszyny i pola maszyna zapisuje nową wartość w polu, zmienia stan, a następnie może przesunąć się o jedno pole w prawo lub w lewo. Taka operacja nazywana jest rozkazem. Maszyna Turinga jest sterowana listą zawierającą dowolną liczbę takich rozkazów. Liczby N i M mogą być dowolne, byle skończone. Czasem dopuszcza się też stan M+1, który oznacza zakończenie pracy maszyny. Lista rozkazów dla maszyny Turinga może być traktowana jako jej program. W teorii złożoności obliczeniowej maszyna Turing jest wzorcowym, matematycznym modelem obliczeń komputerowych zdolnym do wykonywania algorytmów. [10]

3. ZAKOŃCZENIE

Na podstawie analizy wszystkich instrukcji warunkowych oraz warunków jakie są stawiane, możemy stwierdzić, że są one niezbędne w życiu programistów,sterują przebiegiem naszego programu. Można stwierdzić, że są nieodłącznym elementem naszego życia. Każdy program, każdy sterownik i każda maszyna wykorzystują mnóstwo warunków, aby były bezpieczne, spełniały określone standardy i wymogi, gdyż to właśnie dzięki instrukcjom warunkowym program może „zachowywać" się zależnie od spełnienia pewnych warunków, tak jak od niego oczekujemy. Instrukcje te wykonują jedynie wybrany kod w zależności od tego czy wartość danego wyrażenia jest prawdą (true) czy fałszem (false). A jak powszechnie wiadomo, każdy program oparty jest na wszelkiego rodzaju warunkach.

4. PODSUMOWANIE

- **Zapis instrukcji warunkowej if-then wygląda następująco: if(...). Wartość, którą umieszczamy w nawiasach zaokrąglonych, to** wyrażenie logiczne. Początkowo wykonywana jest analiza warunku podanego w postaci wyrażenia logicznego. Jeśli wynikiem jest true, wykonywany jest właściwy blok kodu, a jeśli false – alternatywny.

- Słowo kluczowe **else** oznacza „w przeciwnym wypadku", czyli jeśli warunek nie zostanie spełniony, program wykona inny kod. Ogólny zapis instrukcji warunkowej **else** będzie więc wyglądał nastepująco; if(...)...else...

- **Wyrażenie warunkowe** jest odmianą instrukcji warunkowej if-then z tą różnicą, że wykonany blok kodu musi zwrócić jakąś wartość, która staje się jednocześnie wynikiem całego wyrażenia.Umożliwia ona sprawdzenie warunku na poziomie wyrażenia

- Jedynymi dozwolonymi typami danych w instrukcji **switch** są liczby całkowite.

- Instrukcja wielokrotnego wyboru **switch** są nieporównywalnie mniejsze, jednak używanie jej w niektórych przypadkach jest znacznie korzystniejsze

- W zależności od tego, czy wartość logiczna, uzyskana w wyniku ewaluacji wyrażenia reprezentującego warunek jest równa wartości logicznej true (prawda), czy false (fałsz), wykonywanie pętli jest kontynuowane bądź przerywane.

- Bardzo ważną rolę odgrwają warunki pętli decydujące o kontynuacji lub zaprzestaniu wykonywania pętli , które mogą być sprawdzane na początku, wewnątrz i na końcu pętli.

- Pętlę while wykorzystuje się w sytuacjach, kiedy niezbędne jest wykonywanie danej operacji dopóki nie zostanie spełniony warunek.

- W pętli until warunek zakończenia sprawdzany jest dopiero po wykonaniu instrukcji. Oznacza to, że pętla repeat zawsze będzie wykonana co najmniej raz.

- Dopasowanie do wzorca to operacja, gdzie pewne wyrażenie sprawdza się ze wzorcem, w którym może znajdować się jedno lub więcej wolnych miejsc.

Literatura:

1. Arindama Singh: *Logics for Computer Science*. PHI Learning Pvt. Ltd., 2004, s. 283. ISBN 81-203-2284-3

2. Martin Richards: T*he BCPL Cintsys and Cintpos User Guide.* Cambridge: Computer Laboratory University of Cambridge, January 28, 2011. [dostęp 2011-01-31]. (ang.)

3. Michał Iglewski, Jan Madey, Stanisław Matwin: Pascal. *Język wzorcowy – Pascal 360*. Wyd. wydanie trzecie – zmienione. Warszawa: Wydawnictwa Naukowo-Techniczne, 1984, seria: Biblioteka Inżynierii Oprogramowania. ISSN 0867-6011. ISBN 83-85060-53-7. (pol.)

4. Andrzej Marciniak: Borland *Pascal 7.0*. Poznań: Nakom, 1994, seria: Biblioteka Użytkownika Mikrokomputerów. ISSN 0867-6011. ISBN 83-85060-53-7. (pol.)

5. John Walkenbach: *Excel 2003 PL. Programowanie w VBA*.. HELION, 2004. ISBN 837361-504-0. (pol.)

6. Brian W. Kernighan, Dennis M. Ritche: *Język C*. Warszawa: Wydawnictwa Naukowo-Techniczne, 1988, seria: Biblioteka Inżynierii Oprogramowania. ISBN 83-204-1067-3. (pol.)

7. Jan Bielecki: *Turbo C z grafiką na IBM PC*. Warszawa: Wydawnictwa Naukowo-Techniczne, 1990, seria: Mikrokomputery. ISBN 83-204-1101-7. (pol.)

8. Jan Bielecki: *Od C do C++, programowanie obiektowe w języku C*. Warszawa: Wydawnictwa Naukowo-Techniczne, 1990. ISBN 83-204-1332-X. (pol.)

9. Maszyna Turinga w serwisie edukacyjnym I LO w Tarnowie [Autor artykułu: mgr Jerzy Wałaszek
Wersja 2.1 - zakończono 14 IV 2004]

10. Strona internetowa www.eduinf.waw.pl [strona odwiedzona ostatnio 30.08.2018]

11. Operator_warunkowy na stronie inyternetowej https://pl.wikipedia.org/wiki [Treść udostępniana na licencji CC BY-SA 3.0,strona ostatni raz edytowana 14 stycznia 2018]

12. https://pl.wikipedia.org/wiki/Instrukcja_wyboru [Tę stronę ostatnio edytowano 25 lip 2018, 19:57.]

13. Podręcznik Visual Basic na stronie internetowej www.Wikibooks.pl [Tę stronę ostatnio edytowano 1 sierpnia 2014, 19:06.]

14. Podręcznik języka C na stronie internetowej www.Wikibooks.pl [Tę stronę ostatnio edytowano 8 lip 2018, 21:01.]

15. Strona internetowa : https://4programmers.net/Delphi/Repeat [strona odwiedzona 30.08.2018]

16. Dopasowanie do wzorca, portali wikipedia [Tę stronę ostatnio edytowano 18 sty 2016, 09:31.]