Technologia informacyjna

(laboratoria)

Rok II / semestr 3

I. Informacje o przedmiocie

1. Forma zaliczenia

- kolokwium zaliczeniowe – teoria, schemat blokowy algorytmu, zapis algorytmu w języku fortran (13 spotkanie);
- ćwiczenie projektowe (oddanie poprawnie wykonanego projektu na 13-15 zajęciach);
- aktywność na zajęciach;
- obecność (dopuszczalne są dwie nieobecności w ciągu semestru);

2. Cel przedmiotu

Celem przedmiotu jest przekazanie studentom podstawowej wiedzy z zakresu tworzenia algorytmów. Przedmiot nie skupia się na nauce programowania, lecz na wykorzystywaniu zalet algorytmów w codziennych czynnościach zawodowych jak i prywatnych, w celu ułatwienia i usystemowienia przeprowadzania kolejnych kroków wykonywanych czynności.

II. *Podstawowe wiadomości z zakresu przedmiotu*

Technologia informacyjna

Całokształt zagadnień, metod, środków i działań związanych z przetwarzaniem informacji. Stanowi połączenie zastosowań informatyki i telekomunikacji, obejmuje również sprzęt komputerowy oraz oprogramowanie, a także narzędzia i inne technologie związane z przetwarzaniem, przesyłaniem, przechowywaniem, zabezpieczaniem i prezentowaniem informacji. Dostarcza ona użytkownikowi narzędzi, za pomocą, których może on pozyskiwać informacje, selekcjonować je, analizować, przetwarzać, gromadzić, zarządzać i przekazywać innym ludziom.

Informatyka

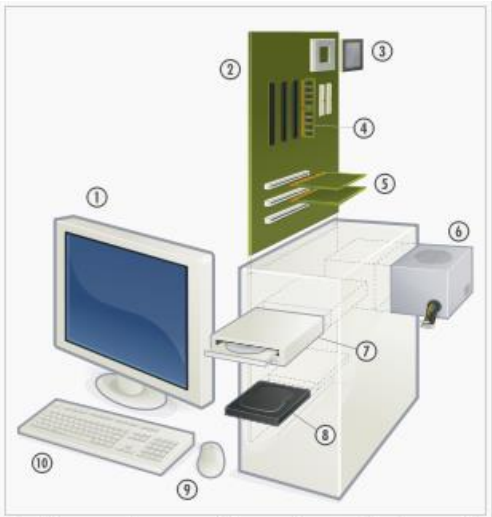
Dyscyplina nauki zaliczana do nauk ścisłych oraz techniki zajmująca się przetwarzaniem informacji, w tym również technologiami przetwarzania informacji oraz technologiami wytwarzania systemów przetwarzających informację. Początkowo stanowiła część matematyki, później rozwinęła się do odrębnej dyscypliny – pozostaje jednak nadal w ścisłej relacji z matematyką, która dostarcza informatyce podstaw teoretycznych.

Sprzęt komputerowy(ang. hardware)

Materialna część komputera. Ogólnie hardware'em nazywa się sprzęt komputerowy odróżniony od software'u – czyli oprogramowania. Podział ten jest nieostry, gdyż współcześnie wiele elementów sprzętu komputerowego posiada "wszyte" weń na stałe oprogramowanie, stanowiące jego integralną część, bez którego elementy te nie mogłyby funkcjonować. Np. większość drukarek komputerowych posiada w swojej pamięci zestaw komend, przy pomocy, których realizuje proces drukowania i których odpowiednik znajduje się w pamięci komputera stanowiąc programowy sterownik tego urządzenia. Wiele urządzeń – typu karty graficzne, płyty główne posiada własne oprogramowanie nazywane BIOS-em. W stosunku do oprogramowania niektórych urządzeń używa się słowa firmware.

Rodzaje komputerów

- ***komputery osobiste*** ("PC", z ang. *personal computer*) – o rozmiarach umożliwiających ich umieszczenie na biurku, używane zazwyczaj przez pojedyncze osoby;



Podstawowe komponenty komputera osobistego:
1) Monitor
2) płyta główna
3) procesor (CPU)
4) pamięć operacyjna (RAM)
5) karta rozszerzenia
6) zasilacz
7) napęd optyczny (CD, DVD itp.)
8) dysk twardy (HDD/SSD)
9) mysz
10) klawiatura

- ***komputery domowe*** – poprzedniki komputerów osobistych, korzystające z telewizora, jako monitora.

- ***konsola*** – następca komputera domowego wyspecjalizowany w programach rozrywkowych. Zazwyczaj korzysta z telewizora jako głównego wyświetlacza. Posiada ograniczone oprogramowanie przygotowane do wydajnego uruchamiania programów i gier. Na niektórych modelach można zainstalować inny system operacyjny i wykorzystywać do specyficznych zastosowań, np. procesory graficzne konsoli PS3 znakomicie nadają się do łamania różnego rodzaju kodów.

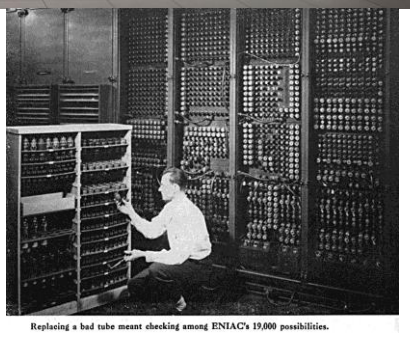
- ***komputery mainframe*** – często o większych rozmiarach, których zastosowaniem jest przetwarzanie dużych ilości danych na potrzeby różnego rodzaju instytucji, pełnienie roli serwerów itp.

- ***komputery gospodarcze*** – używane w gospodarstwach rolnych w celu efektywnego sterowania procesami produkcyjnymi.

- ***Komputery wbudowane*** – (lub osadzone, ang. *embedded*) specjalizowane komputery służące do sterowania urządzeniami z gatunku automatyki przemysłowej, elektroniki użytkowej (np. telefony komórkowe itp.) czy wręcz poszczególnymi komponentami wchodzącymi w skład komputerów.

- ***Laptop*** (ang. *lap* – kolana, *top* – na wierzchu), nazywany również *notebook* – przenośny komputer osobisty. Inne zminiaturyzowane komputery (mniejsze od laptopów) to netbooki, palmtopy (np. Palm lub Pocket PC) lub smartfony.

- ***superkomputery*** – największe komputery o dużej mocy obliczeniowej, używane do czasochłonnych obliczeń naukowych i symulacji skomplikowanych systemów.



Replacing a bad tube meant checking among ENIAC's 19,000 possibilities.

- ***Laptop*** (ang. *lap* – kolana, *top* – na wierzchu), nazywany również *notebook* – przenośny komputer osobisty. Inne zminiaturyzowane komputery (mniejsze od laptopów) to netbooki, palmtopy (np. Palm lub Pocket PC) lub smartfony.

- ***Palmtop*** (także: PDA, Personal Digital Assistant, komputer kieszonkowy) – bardzo mały, przenośny komputer osobisty. Mniejszy od laptopa, czy też netbooka – z powodzeniem mieści się w dłoni lub w kieszeni (ang. palm – dłoń, top – na wierzchu). Palmtopy są komputerami programowalnymi – można w nich instalować oprogramowanie, np. pobrane lub zakupione w Internecie.

- ***Tablet*** przenośny komputer większy niż telefon komórkowy lub palmtop, którego główną właściwością jest posiadanie dużego ekranu z zastosowaną technologią Multi-Touch. W przeciwieństwie do większości klasycznych urządzeń tablet PC tablety nowszego typu nie posiadają fizycznej klawiatury, użytkownik posługuje się klawiaturą wirtualną, dotykając ekran bezpośrednio, bez użycia rysika.

- ***Smartfon*** (ang. *smartphone*) – przenośne urządzenie telefoniczne łączące w sobie funkcje telefonu komórkowego i komputera kieszonkowego (PDA – Personal Digital Assistant). Pierwsze smartfony powstały pod koniec lat 90., a obecnie łączą funkcjetelefonu komórkowego, poczty elektronicznej, przeglądarki sieciowej, pagera, GPS, cyfrowego aparatu fotograficznego i kamery wideo.

Podstawowy skład zestawu komputerowego

jednostka centralna; monitor; klawiatura

Oprogramowanie (ang. software)

Całość informacji w postaci zestawu instrukcji, zaimplementowanych interfejsów i zintegrowanych danych przeznaczonych dla komputera do realizacji wyznaczonych celów. Celem oprogramowania jest przetwarzanie danych w określonym przez twórcę zakresie. Oprogramowanie to dział informatyki. Oprogramowanie jest synonimem terminów *program komputerowy* oraz *aplikacja,* przy czym stosuje się go zazwyczaj do określania większych programów oraz ich zbiorów.

Oprogramowanie tworzą programiści w procesie programowania. Oprogramowanie jako przejaw twórczości jest chronione prawem autorskim, twórcy zezwalają na korzystanie z niego na warunkach określanych w licencji.

Oprogramowanie pisane jest zazwyczaj przy użyciu różnych języków programowania z wykorzystaniem algorytmów. Programy przekształcające oprogramowanie z postaci źródłowej na binarną to kompilatory. Niektóre oprogramowanie, np. napisane w całości w językach interpretowanych, może występować tylko w jednej postaci, spełniającej oba zadania.

Podział oprogramowania

- oprogramowanie systemowe – realizujące funkcje konieczne dla działania systemu komputerowego,
- oprogramowanie do tworzenia oprogramowania,
- biblioteki programistyczne – oprogramowanie do wykorzystania przez inne programy,
- oprogramowanie użytkowe – mające bezpośredni kontakt z użytkownikiem i realizujące usługi dla tego użytkownika za pomocą aplikacji, z technicznego punktu widzenia jest to oprogramowanie korzystające z usług m.in. systemu operacyjnego, a szerzej – oprogramowania systemowego.

Pojęcie oprogramowania systemowego jest znacznie szersze niż systemy operacyjne. W jego skład wchodzi też oprogramowanie serwerowe i każde oprogramowanie, z którym użytkownik nie ma bezpośredniej styczności.

System operacyjny(ang. Operating System, skrót OS)

Oprogramowanie zarządzające systemem komputerowym, tworzące środowisko do uruchamiania i kontroli zadań użytkownika.

W celu uruchamiania i kontroli zadań użytkownika system operacyjny zajmuje się:

- planowaniem oraz przydziałem czasu procesora poszczególnym zadaniom,
- kontrolą i przydziałem pamięci operacyjnej dla uruchomionych zadań,
- dostarcza mechanizmy do synchronizacji zadań i komunikacji pomiędzy zadaniami,
- obsługuje sprzęt oraz zapewnia równolegle wykonywanym zadaniom jednolity, wolny od interferencji dostęp do sprzętu.

Dodatkowe przykładowe zadania, którymi może ale nie musi zajmować się system operacyjny:

- ustalanie połączeń sieciowych
- zarządzanie plikami.

Wiele systemów operacyjnych posiada środowiska graficzne ułatwiające komunikacje maszyny z użytkownikiem.

Rodzaje systemów operacyjnych

Amiga; Apple; Atari ST; Be i pochodne; DEC/Compaq;
Google; IBM; ICL; Novell; SuSE Linux;
NeXT; Unisys, UNIX i jego pochodne i inne.

Microsoft i pochodne (MS-DOS, Microsoft Windows: 1.0, 2.0, 3.x, 95/98/Me, CE i Mobile, NT/2000/XP/2003/FLP/Vista/2008/7/8);

Jednostki pamięci

Bit (w ang. *kawałek*, skrót od *binary digit*, czyli cyfra dwójkowa) – najmniejsza ilość informacji potrzebna do określenia, który z dwóch równie prawdopodobnych stanów przyjął układ. Jednostka logiczna. Jest to również najmniejsza jednostka informacji używana w odniesieniu do sprzętu komputerowego a oznaczana jest za pomocą „b". Przeważnie stosuje się podstawowe przedrostki wielokrotności SI, czyli o mnożniku **1000**. Bit przyjmuje jedną z dwóch wartości, które zwykle określa się jako 0 (zero) i 1 (jeden), choć można przyjąć dowolną inną parę wartości, np. prawda i fałsz, tak lub nie czy -1 i +1. W pierwszym przypadku bit jest tożsamy z cyfrą w systemie dwójkowym. Binarny sposób zapisu informacji związany jest z tym, że komputer jako urządzenie cyfrowe rozpoznać może dwa stany napięciowe:

0 – brak napięcia lub bardzo niskie (mniej niż 10% wartości wysokiego)
1 – wysokie napięcie.

Z tego względu obliczenia wykonywane przez procesor opierają się na binarnym (dwójkowym) systemie liczbowym.

Bajt (dop. *bajtu* lub *bajta*) – najmniejsza adresowalna jednostka informacji pamięci komputerowej, składająca się z bitów. W praktyce przyjmuje się, że jeden bajt to 8 bitów,

Przelicznik					
bit [b]	bajt [B]	kilobajt [kB]	megabajt [kB]	gigabajt [kB]	terabajt [kB]
8	1	0,000976563	9,53674E-07	9,31323E-10	9,09495E-13
8192	1024	1	0,000976563	9,53674E-07	9,31323E-10
8388608	1048576	1024	1	0,000976563	9,53674E-07
8589934592	1073741824	1048576	1024	1	0,000976563
8,79609E+12	1,09951E+12	1073741824	1048576	1024	1
9,0072E+15	1,1259E+15	1,09951E+12	1073741824	1048576	1024

Język w ujęciu logicznym

Logice matematycznej to pewien zbiór symboli, przy użyciu których można tworzyć bardziej złożone wyrażenia (na przykład formuły, zdania) według ściśle określonych reguł syntaktycznych. Przyjmuje się, że w danym języku L mogą występować (w dowolnej ilości) symbole funkcyjne, relacyjne oraz symbole stałych. Zdania napisane przy użyciu języków tego typu wystarczają do opisu większości własności dowolnych struktur matematycznych oraz do wyrażenia twierdzeń mówiących o tych strukturach.

Język formalny

Jest to podzbiór zbioru wszystkich słów nad skończonym alfabetem. Język formalny jest kluczowym pojęciem w informatyce, logice matematycznej i językoznawstwie. Język formalny nie jest uściśleniem pojęcia języka naturalnego i nie powinien być z nim mylony.

Aby zdefiniować język formalny, najpierw definiuje się alfabet, wybierając jakiś niepusty zbiór skończony. Elementy tego zbioru nazywane są symbolami. Ciągi symboli, o skończonej długości, nazywane są słowami. Dowolny zbiór takich ciągów nazywany jest językiem formalnym.

Przykłady języków formalnych:

- zbiór wszystkich słów złożonych z liter polskiego alfabetu i występujących w pewnym słowniku,
- zbiór takich słów złożonych z cyfr od 0 do 9, które przedstawiają liczbę pierwszą,
- zbiór słów złożonych z zer i jedynek, w których zer jest więcej niż jedynek,
- zbiór prawidłowo napisanych równań matematycznych,
- zbiór programów, które po skompilowaniu i uruchomieniu zawieszą dany komputer,
- zbiór pusty.

Język naturalny

Ukształtowany społecznie system budowania wypowiedzi, używany w procesie komunikacji interpersonalnej. Na język składają się dwa elementy:

Otwarty zbiór znaków, zbudowanych z fonemów, morfemów, wyrazów, stałych zwrotów i wyrażeń.

III. Schematy blokowe - teoria

Algorytm

To jednoznaczny przepis obliczenia w skończonym czasie pewnych danych *wejściowych* do pewnych danych *wynikowych*.

Zazwyczaj przy analizowaniu bądź projektowaniu algorytmu zakłada się, że dostarczane dane wejściowe są poprawne, czasem istotną częścią algorytmu jest nie tylko przetworzenie, ale i weryfikacja danych.

Zgodnie z założeniem o jednoznaczności dla identycznego zestawu danych początkowych, algorytm zdefiniowany klasycznie zawsze zwróci identyczny wynik.

Schemat blokowy

Jest narzędziem nakierowanym na prezentację kolejnych czynności w projektowanym algorytmie. Realizowane jako diagram, na którym procedura, system albo program komputerowy są reprezentowane przez opisane figury geometryczne, połączone liniami zgodnie z kolejnością wykonywania czynności wynikających z przyjętego algorytmu rozwiązania zadania.

Cechuje je:

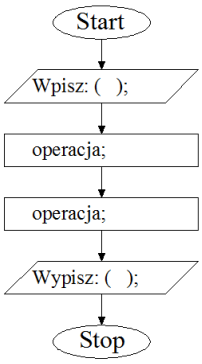
zasada budowy,
elastyczność zapisów,
możliwość zapisu z użyciem składu wybranego języka programowania,
łatwa kontrola poprawności algorytmu.

Schematy blokowe pozwalają na prostą zamianę instrukcji na instrukcje programu komputerowego.

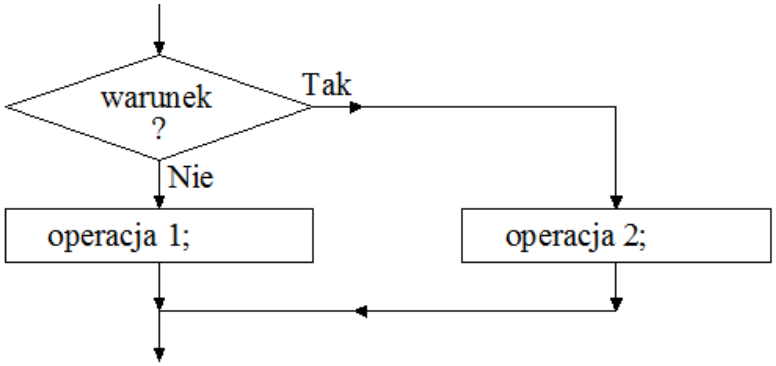
Jest to graficzny sposób przedstawienia algorytmu. Składa się z geometrycznych figur płaskich połączonych liniami tworząc diagram. Połączenia poszczególnych bloków są zgodne z kolejnością wykonywanych zadań w algorytmie

Symbol elementu	Nazwa elementu	Funkcja elementu	Opis
	strzałka	Łączenie poszczególnych bloków	linia zakończona grotem wskazującym kierunek algorytmu
Start Stop	Bloki graniczne	Bloki te rozpoczynają i kończą każdy algorytm	mają kształt owalu. Ze skrzynki START wychodzi tylko jedno połączenie, skrzynka STOP nie ma połączenia wychodzącego.
Wpisz: (w,k); Wypisz: (A);	Bloki wejścia / wyjścia	Wprowadzanie (czytanie) danych lub wyprowadzanie (drukowanie, pisanie) wyników	jest równoległobokiem, wchodzi i wychodzi z niej jedno połączenie. Skrzynka graniczna
i := 1;	Blok operacyjny	Wykonywanie różnych działań, np. sumowania, podstawiania wartości	ma kształt prostokąta.
i <= 1 ? Tak Nie	Blok warunkowy	Sprawdzanie warunku, np. czy i<= l	mają kształt rombu. Ze skrzynki wychodzą tylko dwa połączenia: jedno oznaczone TAK, a drugie NIE.

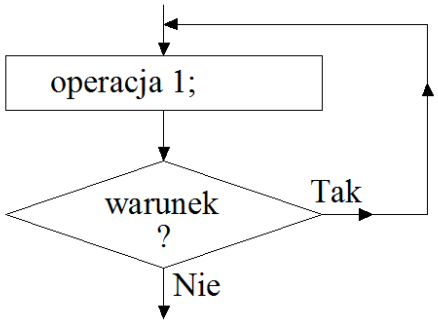
Układy bloków w schemacie



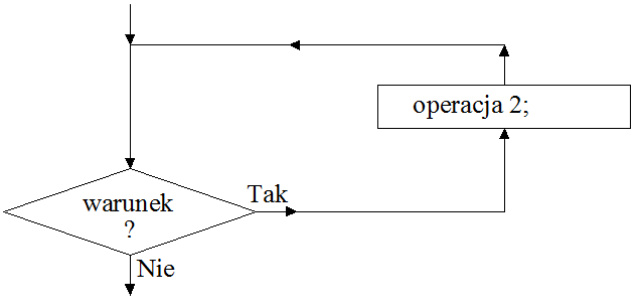
Sekwencja



Rozgałęzienie



Pętla „Do – While”



Pętla „While”

IV. Schematy blokowe – zadania

- Wpisywanie danych w macierz
- Dodawanie, odejmowanie, mnożenie, dzielenie wartości
- Funkcja kwadratowa
- Poszukiwanie min i max w wektorach
- Poszukiwanie min i max w macierzach
- Średnia arytmetyczna, średnia warzona

V. Prototyp języka programowania o cechach strukturalnych, Język programowalny „Fortran”

Operacja	Prototyp języka programowania o cechach strukturalnych	Język programowalny „Fortran”
Deklaracja zmiennych	Nazwa programu POCZĄTEK Całkowite M, N, I, J Rzeczywiste tablice A(10,30)	PROGRAM NAME INTEGER M, N, I, J, REAL A(20,30),
Wprowadzanie danych	Wpisz (M, N)	88 WRITE (*,*) 'Podaj wymiar macierzy:' WRITE (*,*) 'M>=1; M<=20' WRITE (*,*) 'N>=1; N<=30' READ (*,*) M, N
Sprawdzenie poprawności wprowadzonych danych	----------	IF (M.LT.1.OR.M.GT.20.OR.N.LT.1.OR.N.GT.30) THEN PAUSE 'Podano błędy wymiar macierzy [ENTER]' GO TO 88 EDN IF
Uzupełnianie macierzy za pomocą pętli	I:=1; POWTARZAJ POCZĄTEK J:=1; POWTARZAJ POCZĄTEK WPISZ (A(I,J)); J:=J+1; KONIEC AŻ DO (J≤N) PP(I):=I; I:=I+1; KONIEC AŻ DO (I≤M)	DO I=1, M DO J=1, N WRITE (*,*) 'Podaj wartość A(',I,',',J,')=' READ (*,*) A(I,J) END DO END DO
Zakończenie programu	KONIEC	END

operator logiczny	nazwa
.NOT.	negacja
.AND.	koniunkcja
.OR.	alternatywa
.EQV.	równoważność
.NEQV.	zaprzeczenie równoważności

operator relacji	opis
.LT.	ang. less than - inaczej <
.LE.	ang. less equal - inaczej <=
.GT.	ang. greater than - inaczej >
.GE.	ang. greater equal - inaczej >=
.EQ.	ang. equal - inaczej == (lub = w Pascalu)
.NE.	ang. not equal - inaczej != (lub <> w Pascalu)

„+” – dodawanie; „-” – odejmowanie; „*” – mnożenie ; „/” – dzielenie; „**” – potęgowanie;

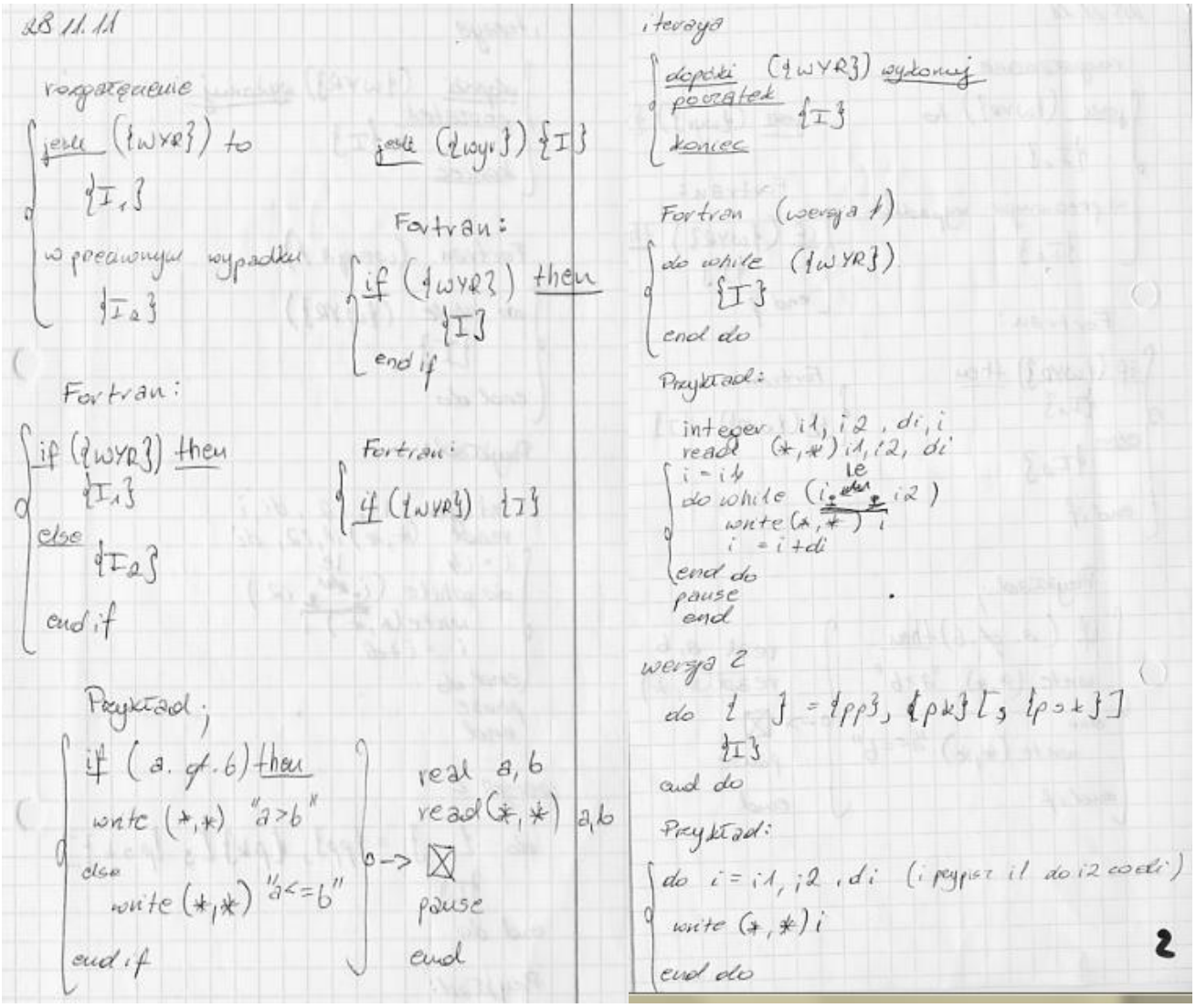
Rozgałęzienie

jeżeli {warunek} to
Operacja 1
w przeciwnym wypadku
Operacja 2

if {warunek} then
Operacja 1
else
Operacja 2
end if

jeżeli {warunek} to
Operacja 1

if {warunek} then
Operacja 1
end if



Zastosowanie do zapisu przekazania tablic

```
integer n, i
real a(100)   ! deklaracja tablicy

read (*,*) n
do i = 1,n
   read (*,*) a(i)
end do
```

```
do i = 1,n
   write (*,*) i, a(i)
end do

pause
end
```

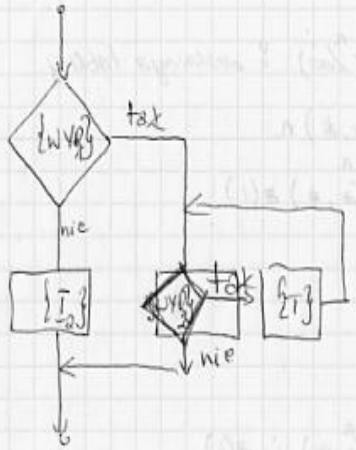
Tablice 2d

```
integer m, n, i, j
real a(10,10)

read (*,*) m, n

do i = 1,m
   do j = 1,n
      read (*,*) a(i,j)
   end do
end do
```

```
do i = 1,m
   do j = 1,n
      write (*,'(e11.4\)') a(i,j)
   end do
   write (*,*) " "
end do

pause
end
```

blokowane naturalne przejście (do następnej linii)

Zad 1 Zapisać w wersji tekstowej (pseudokod)



jeśli ({wVR}) to

dopóki ({wVR₂}) ~~to~~ wykonuj

pauz

{I};

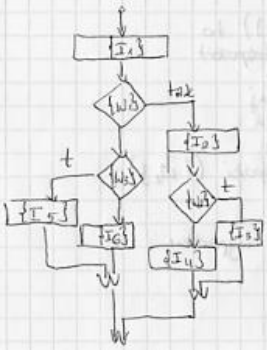
koniec

w przeciwnym wypadku

{I₂};

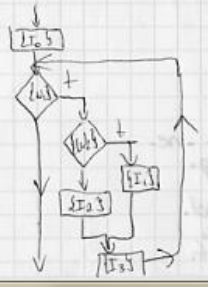
koniec jeśli

Zad 2 Zapisać w wersji tekstowej



Zad 3 Narysować schemat blokowy na podstawie pseudokodu

{I₀};
dopóki ({w₁}) wykonuj
początek
jeśli ({w₂}) to
{I₁};
w przeciwnym wypadku
{I₂};
koniec jeśli
{I₃}
koniec dopóki



Zad 4 Napisać schemat blokowy

{I₁};
jeśli ({w₁}) to
początek ~~(dopóki)~~
{I₂};
wykonuj
początek
{I₃}
koniec dopóki ({w₂});
{I₄};
koniec
w przeciwnym wypadku
{I₅}
koniec jeśli
{I₆}

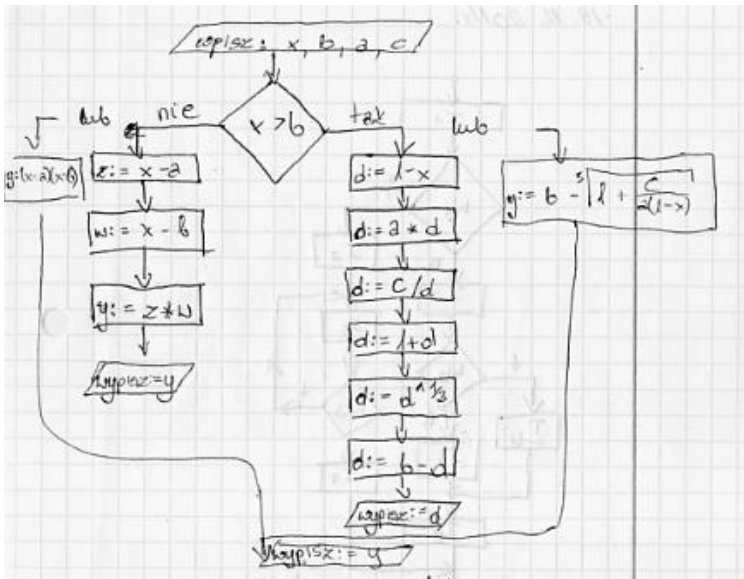
12.12.2011r.

$$y := \begin{cases} \text{jeśli } x \geq b \text{ to} \\ b - \sqrt[3]{1 + \frac{c}{a(1-x)}} \\ \text{w przeciwnym wypadku} \\ (x-a)(x-b) \end{cases}$$

1) sch. blokowy
2) pseudokod
3) pr. w „fortran"

> .gt.
≥ .ge.
< .lt.
≤ .le.
≠ .ne.
== .eq.

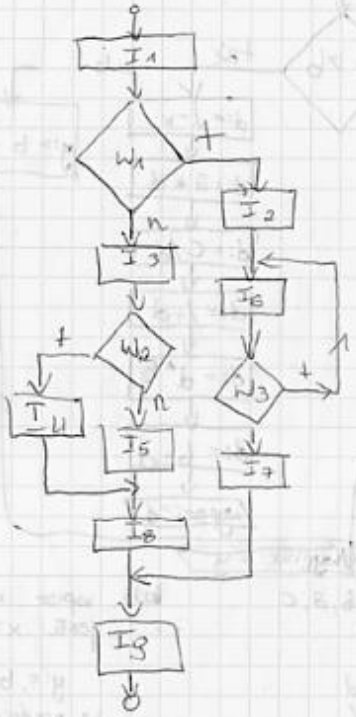
∧ .and.
∨ .or.
¬ .not.



wpisz x, b, a, c
jeśli x > b
d := 1 - x
d := a * d
d := c / d
d := 1 + d
d := d ^ 1/3
d := b - d
wypisz := d
w przeciwnym wypadku
z := x - a
w := x - b
y := z * w
wypisz y
koniec jeśli

lub wpisz x, b, a, c
jeśli x > b
$y := b - \sqrt[3]{1 + \frac{c}{2(1-x)}}$
w przeciwnym wypadku
y := (x - a)(x - b)
koniec jeśli
wypisz y

19. 12. 2011r.



1) pseudo - pr
2) program w j.f

1) I_1
jeżeli W_1 to
I_2
wykonuj
I_6
dopóki W_3
(*) I_7
w przeciwnym wypadku
I_3
jeżeli W_2 to
I_4
w przeciwnym wypadku
I_5
koniec jeżeli
I_8
koniec jeżeli
I_9

(*) - koniec dopóki

2) I_1
if W_1 than
I_2
do while W_3
I_6
end do
I_7
else
I_3
if W_2 than
I_4
else
I_5
end if
I_8
end if
I_9

→ 1000 continue
I_6
if W_3 go to 1000