

# Egzamin Fortran

## Spis treści

<b>1</b>	<b>Opracowanie zagadnień</b>	<b>3</b>
1.1	Reguły zapisu instrukcji w Fortranie77 . . . . .	3
1.2	Podać przykład zastosowania dyrektywy IMPLICIT . . . . .	3
1.3	Ile wartości może być wyznaczonych w segmencie function i jak są zwracane do modułu nadrzędnego . . . . .	3
1.4	W jaki sposób przekazywane są parametry wejściowe do procedury subroutine	3
1.5	Ile wartości może być wyznaczonych w procedurze subroutine i jak są zwracane do modułu nadrzędnego . . . . .	3
1.6	W jaki sposób wartości zmiennych określone w jednym segmencie mogą być dostępne w innym module . . . . .	4
1.7	Podaj postać deklaracji COMMON . . . . .	4
1.8	W różnych segmentach może wystąpić deklaracja COMMON o tej samej nazwie. Co można powiedzieć o zmiennych wyszczególnionych w tych deklaracjach . . . . .	4
1.9	Jaki jest cel stosowania segmentu BLOCK DATA . . . . .	4
1.10	Podać postać i cel stosowania instrukcji INCLUDE . . . . .	5

# 1 Opracowanie zagadnień

## 1.1 Reguły zapisu instrukcji w Fortranie77

## 1.2 Podać przykład zastosowania dyrektywy IMPLICIT

```

1 program test
2   implicit none
3   integer :: a, b, c
4   ...
5 end program

```

Listing 1: dyrektywa implicit

## 1.3 W jaki sposób przekazywane są parametry wejściowe do segmentu function

-podprogram może być wykonywany z danej jednostki programowej wielokrotnie z różnym zestawem danych.

-funkcja jest wywoływana poprzez podanie jej nazwy wraz z listą parametrów aktualnych ujętych w nawiasy okrągłe.

postać:

zmienna=nazwa(lista parametrów aktualnych)

## 1.4 Ile wartości może być wyznaczonych w segmencie function i jak są zwracane do modułu nadrzędnego

-w segmencie function może być wyznaczony jeden element.

postać:

typ function nazwa (lista parametrów formalnych)

deklaracje

część wykonawcza

nazwa=zwracana wartość

return

end

-instrukcja return powoduje zakończenie wykonywania programu i przekazania sterowania do segmentu z, którego następuje jej wywołanie.

## 1.5 W jaki sposób przekazywane są parametry wejściowe do procedury subroutine

Procedura wywoływana jest w następujący sposób:

call nazwa(lista parametrów aktualnych)

gdzie parametry aktualne to parametry wejściowe.

## 1.6 Ile wartości może być wyznaczonych w procedurze subroutine i jak są zwracane do modułu nadrzędnego

-procedura pozwala na zwracanie większej liczby wartości niż jedna.

-procedura nie ma określonego typu.

ogólna postać:

subroutine nazwa (lista parametrów formalnych)

deklaracje

część wykonawcza

return

end

instrukcja return powoduje zakończenie wykonywania programu i przekazania sterowania do segmentu z, którego następuje jej wywołanie.

## 1.7 W jaki sposób wartości zmiennych określone w jednym segmencie mogą być dostępne w innym module

-takie zmienne można uwspólnić poprzez umieszczenie ich na liście obszarów wspólnych.

-obszar ten musi pojawić się w części deklaracyjnej segmentów, pomiędzy którymi są uwspólnione umieszczane w nim zmienne.

## 1.8 Podaj postać deklaracji COMMON

common /nazwa/ zmienna1, zmienna2

-zmienne mogą być proste lub tablicowe.

jeden obszar może być bez nazwy.

nazwy zmiennych mogą być inne taka sama musi być nazwa obszaru wspólnego i jego długość.

## 1.9 W różnych segmentach może wystąpić deklaracja COMMON o tej samej nazwie. Co można powiedzieć o zmiennych wyszczególnionych w tych deklaracjach

- Jeżeli nazwy zmiennych są inne ale nazwa obszaru wspólnego i jego długość jest taka sama, to zmienne o tej samej liczbie porządkowej są sobie równoważne.

### 1.10 Jaki jest cel stosowania segmentu BLOCK DATA

-Poprzez segment block data mogą być wprowadzane dane do programu.

-segment ten służy do nadawania wartości początkowych zmiennym umieszczonym w obszarach wspólnych.

struktura:

BLOCK DATA nazwa

common /nazwa obszaru/ X,Y,I(10)

data x,y,I /0.0,5.92,4\*3,6\*0/

end

### 1.11 Podać postać i cel stosowania instrukcji INCLUDE

postać: include 'nazwa pliku'

cel: służy do dołączenia do pliku fortranowskiego innego pliku zawierającego procedury, funkcję czy bloki danych.

## Spis rysunków

## Kod źródłowy

1	dyrektywa implicit . . . . .	3
---	------------------------------	---