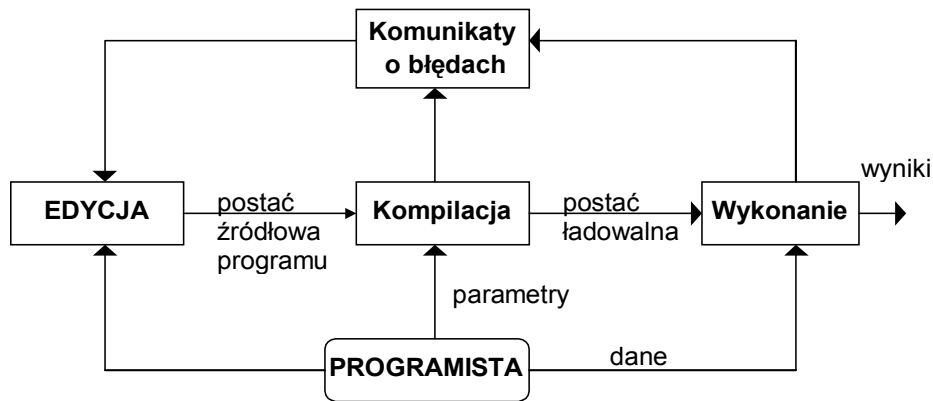
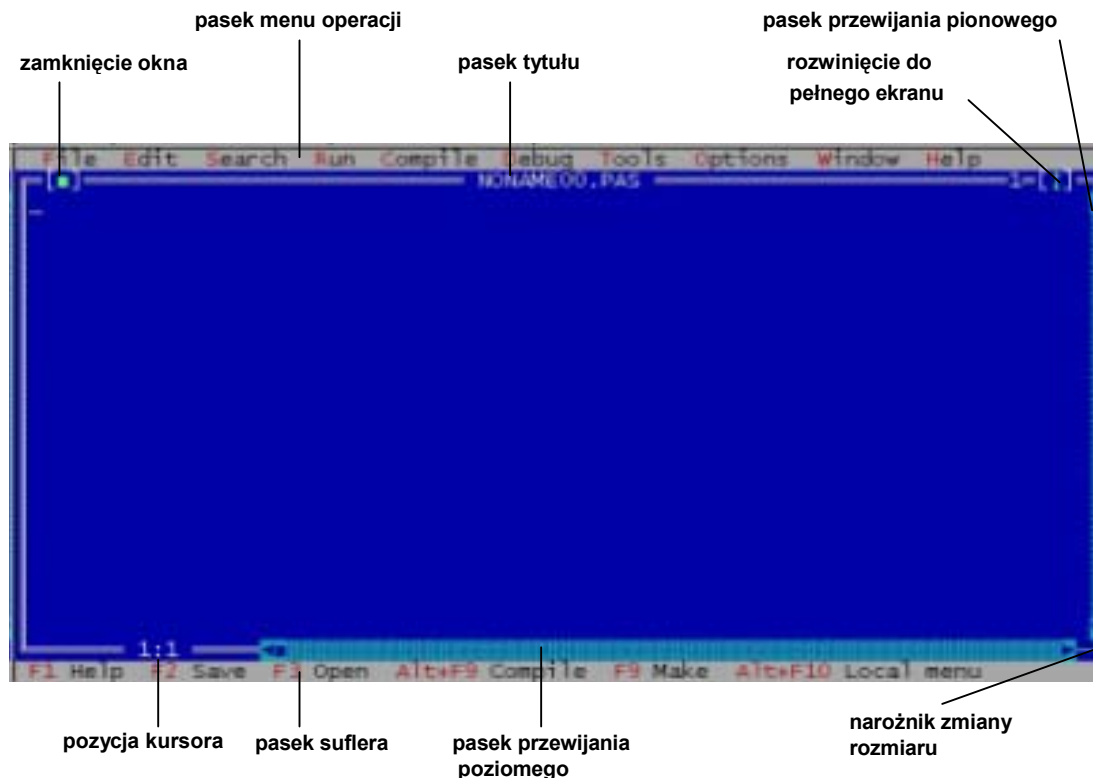


1. Wstęp

Turbo Pascal jest jednym z najpopularniejszych języków programowania wyższego poziomu. Program napisany w tym języku jest ciągiem zdań opisującym określony algorytm. Nazywamy go *postacią źródłową* programu. Aby operacje zawarte w programie mogły być wykonane przez procesor komputera, program musi być przetłumaczony na wykonalną *postać binarną*. Tłumaczenie to, zwane *kompilacją*, jest wykonywane przez kompilator języka, co ilustruje poniższy schemat.



Wszystkie trzy fazy wyróżnione w schemacie: *edycja*, *kompilacja* i *wykonanie* realizowane są przez zintegrowane środowisko Turbo Pascala. Główny program zarządzający tym środowiskiem nazywa się **turbo.exe** (lub **tpc.exe** w trybie chronionym) i przy standardowym zainstalowaniu pakietu Turbo Pascal w wersji 7.0 znajduje się w katalogu **bin**. Główne składowe tego środowiska to **edytor tekstu** przeznaczony do wprowadzania i modyfikacji programu źródłowego i **kompilator** służący do tłumaczenia programu źródłowego na postać binarną. Program źródłowy może być przechowywany w pliku z rozszerzeniem **pas**, a program w postaci binarnej w pliku z rozszerzeniem **exe**. Program **turbo.exe** uruchamia pełnoekranowy edytor w trybie do wprowadzania nowego programu źródłowego z nazwą *noname00.pas* i numerem okienka *1*. Ekran przyjmuje postać jak na rysunku poniżej.



Korzystając z menu można otwierać do edycji inne programy, które pojawiają się w następnych oknach ponumerowanych od 1 do 9.

Operacje z głównego menu dostępne są przez:

- klawisz **F10** i nawigację strzałkami;
- klawisz **Alt** + wyróżniona litera nazwy grupy menu;
- kliknięcie myszką w nazwę grupy menu.

Niektóre operacje dotyczące okienek dialogowych mogą być wykonywane za pomocą myszki przy wykorzystaniu elementów okna pokazanych na rysunku.

Ważniejsze operacje dostępne są poprzez skróty klawiszowe podane w pasku suflera i w poniższej tabeli:

Znaczenie	Skrót klawiszowy
Menu operacji	Alt + wyróżniona litera nagłówka menu
Wybór okna edycyjnego	Alt + numer okna
Przełączenie na pierwszy plan ekranu użytkownika	Alt + F5
Otwarcie pliku	F3
Zapis pliku	F2
Kompilacja	Alt + F9 (F9)
Wykonanie	Ctrl + F9
Step over (krokowo)	F8
Trace into	F7
Watch	Śledzenie wartości wyszczególnionych zmiennych
Breakpoints	„pułapki” – miejsca chwilowego zatrzymania programu
Pomoc	F1
Indeks elementów języka	Shift + F1
Informacja o bieżącym elemencie języka	Ctrl + F1
Poprzednia odpowiedź suflera	Alt + F1
Zamknięcie okna	Alt + F3
Przełączenie okna o numerze podanym cyfrą na pierwszy plan	Alt + cyfra
Wyświetlenie okienek aktywnych	Alt + 0

2. Edytor tekstu

Edycja może dotyczyć programu zapisanego w pliku (**File / Open**) lub nowego (**File / New**). Aktualny tekst programu można zapisać do pliku dyskowego poprzez **File / Save (F2)** lub **File / Save as**. Zakończenie programu edytora (**File / Exit**) jest równoznaczne z zakończeniem działania środowiska Turbo Pascala.

W edytorze dostępne są operacje wycinkowe: kopiowanie, wycinanie i wklejanie. Wycinek można zaznaczyć:

- myszką;
- klawiszami **Ctrl + k + b** – początek, **Ctrl + k + b** – koniec;
- wciśnięty klawisz **Shift** i klawisze ze strzałkami.

Zaznaczony wycinek można:

- skopiować do schowka – **Ctrl + Ins** lub **Edit / Copy**;
- przenieść do schowka – **Shift + Del** lub **Edit / Cut**;
- wstawić ze schowka – **Shift + Ins** lub **Edit / Paste**;

3. Tłumaczenie i wykonanie programu

Program, który znajduje się w aktywnym okienku edytora może zostać skompilowany i wykonany (uruchomiony). Kompilację można wykonać na dwa sposoby:

- z tworzeniem pliku z programem w postaci binarnej o tej samej nazwie co postać źródłowa i rozszerzeniu **exe** (taki plik można wielokrotnie wykonywać z poziomu systemu operacyjnego);
- bez tworzenia pliku **exe** (kompilacja do pamięci operacyjnej).

Podczas procesu kompilacji wykrywane są błędy formalne programu. Błędy te należy poprawić i ponownie skompilować program. Program, który znajduje się w aktywnym oknie edytora i został poprawnie skompilowany można uruchomić poleceniem **Run / Run** lub skrótem klawiszowym **Ctrl + F9**. Polecenie **Run / Run** wykonuje jednocześnie kompilację i uruchomienie programu.

4. Struktura programu

Na program w języku Pascal składają się następujące elementy:

- nagłówek programu;
- opis obiektów (definicje i deklaracje);
- blok instrukcji opisujących operacje, rozpoczynający się słowem kluczowym **begin** a kończącym słowem **end**, po którym występuje kropka kończąca program.

W dowolnym miejscu programu może wystąpić komentarz, jest to tekst ujęty w nawiasy { }.

5. Stałe i zmienne

Stałe i zmienne to podstawowe rodzaje danych (obiektów) jakie mogą wystąpić w programie.

Stałe są to dane wprowadzane wprost w programie (po słowie **const**), których wartość nie zmienia się podczas przebiegu programu, np.:

```
const a = 5;                               {stała numeryczna}
      zn: char = 'P';                       {stała typu znakowego}
      nazwisko: string[10] = 'Nowicki';     {stała typu tekstowego}
```

Zmienne są to miejsca w pamięci określone nazwami zmiennych, w którym można przechowywać różne wartości. Wartości zmiennych ulegają zmianie podczas przeprowadzanych w programie obliczeń.

Stałe i zmienne muszą mieć określony typ. Typ stałej wynika ze sposobu zapisu, typ zmiennej musi być zadeklarowany.

Typ danych określa zbiór wartości jakie mogą być przechowywane w stałej lub zmiennej. Są dwa rodzaje typów: *proste* i *złożone*. Typ prosty oznacza wartości niepodzielne, typ złożony jest strukturą składającą się z innych typów.

5.1. Stałe liczbowe

- stała liczbowa **całkowita (Integer)** jest to ciąg cyfr, ewentualnie poprzedzonych znakiem + lub -, określający liczbę całkowitą z przedziału [-32768, 32768], np. 126, -345, +11, 0;

Typy pokrewne dla typu Integer:

Nazwa	Zakres	Długość	Liczba całkowita
integer	od -32 768 do 32 768	2 bajty	ze znakiem
shortint	od -128 do 127	1 bajt	krótka ze znakiem
longint	od -2 147 483 648 do 2 147 483 647	4 bajty	długa ze znakiem
byte	od 0 do 255	1 bajt	krótka bez znaku (bajt)
word	od 0 do 65 535	2 bajty	bez znaku (słowo)

- stała liczbowa **rzeczywista (Real)** jest to liczba rzeczywista przedstawiona w zapisie zwykłym: $\pm m . n$ lub w zapisie wykładniczym: $\pm m . n E \pm c$, gdzie **m**, **n** i **c** oznaczają ciąg cyfr; wartości typu rzeczywistego mogą należeć do przedziału [-2.9E-39, 1.7E+38], np. 13.76, -123.56, 0.0, 1.34E-2, 2.3E+12.

Typy pokrewne dla typu Real:

Nazwa	Zakres	Długość	Dokładność (cyfr)
real	od 2.9e-39 do 1.7e38	6 bajtów	11 ÷ 12
single	od 1.5e-45 do 4.0e38	4 bajty	7 ÷ 8
double	od 5.0e-324 do 1.7e308	8 bajtów	15 ÷ 16
extended	od 3.4e-4932 do 1.1e4932	10 bajtów	19 ÷ 20
comp	od -9.2e18 do 9.2e18	8 bajtów	19 ÷ 20



5.2. Stała tekstowa

Stała tekstowa to ciąg znaków podanych w apostrofach, np. 'Jan Kowalski', '123', 'suma liczb wynosi'.

5.3. Zmienne

Deklaracje zmiennych mają postać:

```
var
  lista_nazw_zmiennych: typ;
  lista_nazw_zmiennych: typ;
```

Typ zmiennych numerycznych określony jest słowem kluczowym:

Integer – numeryczny całkowity,
Real – numeryczny rzeczywisty

Nazwy zmiennych na liście oddzielone są przecinkami, deklaracje typów oddzielone są średnikiem (separatorem), np.:

```
var a, b, suma: Integer;      {zmienne numeryczne całkowite}
    iloraz: Real;           {zmienna numeryczna rzeczywista}
    zn: char;               {zmienna typu znakowego}
    nazwisko: string[20];   {zmienna typu tekstowego}
```

6. Podstawowe instrukcje proste

Przy opisie instrukcji przyjęta została konwencja: nawiasy klamrowe { } oznaczają elementy do wyboru, nawiasy kwadratowe [] – elementy opcjonalne.

6.1. Instrukcja przypisania

Instrukcja przypisania ma postać: Zmienna := Wyrażenie;
gdzie symbol := oznacza operację nadania (przypisania) wartości zmiennej, np.:

```
a := 10;
b := 20;
suma := a + b;
```

6.2. Instrukcja wprowadzania danych z klawiatury

Do **wprowadzania** (standardowo z klawiatury) wartości zmiennych w trakcie wykonania programu służy procedura:

```
read  } [(lista zmiennych)];
readln }
```

Zmiennym wymienionym jako argumenty przyporządkowane zostają wartości wynikające z danych podawanych tekstowo z klawiatury w postaci zgodnej z typem zmiennych, np.

```
read(a, b);
readln(x, y);
```

Liczby podajemy polami oddzielonymi co najmniej jedną spacją lub w osobnych wierszach.

Jeżeli wpisujemy za dużo liczb to nadwyżka jest pamiętana dla następnej instrukcji **read**, natomiast w instrukcji **readln** czytane jest tyle ile potrzeba, a reszta zostaje zignorowana. Należy uważać na zgodność typów zmiennych i danych (stała całkowita może być wczytana do zmiennej rzeczywistej lecz nie odwrotnie).

6.3. Instrukcja wyprowadzania danych na ekran

Procedurą **wyprowadzania** danych (standardowo na ekran) jest:

```
write  } [(lista elementów)];
writeln }
```

Po wykonaniu instrukcji **write** kursor pozostaje na końcu wiersza, po **writeln** następuje zmiana wiersza. Element listy może przyjąć jedną z trzech postaci:

- W – wyrażenie, oznaczające wartość, która ma być wyprowadzona na ekran;
 W: m – m oznacza szerokość pola wyświetlania (liczba znaków), dla liczb typu **real** $m \geq 8$;
 W: m: n – m – jak wyżej (w tym .), n – liczba cyfr po kropce dziesiętnej (tylko dla typu **real**).

Przykład:

```
writeln(suma: 10);
writeln(a: 4, b: 4, suma: 5);
writeln('suma liczb wynosi: ', a + b: 5);
```

Jeśli liczba jest za duża, aby mogła zmieścić się w polu, to szerokość pola jest ignorowana i Turbo Pascal wyprowadza liczbę ze wszystkimi cyframi znaczącymi.

7. Przykład programu

Wprowadź kod następującego programu:

```
program p1; {W programie ustawiane są wartości dwóch zmiennych typu
             całkowitego i wyprowadzana jest na ekran wartość ich sumy,
             różnicy i iloczynu}
var
  a, b: integer;
begin
  a := 100;
  b := -150;
  writeln('a = ', a:6, 'b = ':7, b:6);
  writeln('a + b = ', a + b, ' a - b = ', a - b, ' a * b = ', a * b);
end.
```

Sprawdź działanie programu, a następnie, jako ostatnią instrukcję podaj: **readln**;

Powoduje ona pokazanie (zatrzymanie) końcowego stanu ekranu użytkownika, do czasu naciśnięcia przycisku **Enter**. Sprawdź działanie programu.

Na koniec, uzupełnij program o deklarację modułu: **uses crt**; (przed słowem **var**) oraz pierwszą instrukcją: **clrscr**; oznaczającą początkowe wyzerowanie ekranu. Sprawdź działanie programu.

Zadania

- Zamień w programie `Przyklad_1` drugą instrukcję `writeln` na trzy instrukcje, dające kolejne wyniki w trzech wierszach.
- Dodaj do programu z zadania 1 odpowiednią szerokość pól numerycznych, aby uzyskać wynik, w którym liczby będą pod sobą wyrównane do prawej kolumny, jak poniżej:


```
a + b =      -50
a - b =      250
a * b = -15000
```
- Zmodyfikuj program z zadania 2, aby wartości zmiennych były wczytywane z klawiatury i sprawdź jego działanie dla następujących par danych:
 - 3 7
 - 3
7
 - 3 7.5
- Napisz program, który, prowadząc dialog z użytkownikiem, prosi o wprowadzenie z klawiatury liczb rzeczywistych **a** i **b**, a następnie obliczy i wyświetli sumę, różnicę oraz iloczyn tych liczb w postaci:


```
Suma liczb      35.5 i 5.0   wynosi      40.5
Różnica liczb   35.5 i 5.0   wynosi      30.5
Iloczyn liczb   35.5 i 5.0   wynosi     177.5
```

Sprawdź działanie programu dla danych wprowadzanych w zapisie zwykłym i wykładniczym (naukowym), np.: 7 -7.5 3.25E5 5.75E-4