

Algorytmy i struktury danych

wykład 1

Plan całego wykładu:

- Podstawowe typy danych.
- Pojęcie algorytmu, projektowanie wstępujące i zstępujące, rekurencja.
- Klasy algorytmów.
- Poprawność algorytmu, złożoność obliczeniowa.
- Wskaźniki, dynamiczne struktury danych: listy, drzewa, drzewa wyższych rzędów, grafy, zaawansowane algorytmy grafowe.
- Sortowanie.
- Kodowanie, wyszukiwanie wzorca.
- Mieszanie, zapobieganie kolizji.
- Algorytmy numeryczne.

Literatura:

1. N. Wirth – „Algorytmy + struktury danych = programy”,
2. T.H. Cormen, C.E. Leiserson, R.L. Rivest – „Wprowadzenie do algorytmów”.

Plan wykładu:

- Podstawowe typy danych.
 - typ całkowity,
 - typ zmiennoprzecinkowy,
 - typ znakowy i tekstowy,
 - typ wyliczeniowy i logiczny,
 - typ tablicowy,
 - typ wskaźnikowy,
 - rekordy i obiekty.

Podstawowe typy danych

Typ – w języku programowania jest to definicja określająca:

- rodzaj,
- strukturę,
- zakres wartości

jakie może przyjąć stała, zmienna, identyfikator, argument lub wynik funkcji.

Typ – w języku programowania jest to definicja określająca:

- rodzaj,
 - strukturę,
 - zakres wartości
- jakie może przyjąć stała, zmienna, identyfikator, argument lub wynik funkcji.

całkowity	zmiennoprzecinkowy
znakowy	tekstowy
wyliczeniowy	logiczny
wskaźnikowy	

Typ – w języku programowania jest to definicja określająca:

- rodzaj, ←
 - strukturę, ←
 - zakres wartości
- jakie może przyjąć stała, zmienna, identyfikator, argument lub wynik funkcji.

całkowity	zmiennoprzecinkowy
znakowy	tekstowy
wyliczeniowy	logiczny
wskaźnikowy	

tablica	rekord
obiekt	

Typ – w języku programowania jest to definicja określająca:

- rodzaj,
 - strukturę,
 - zakres wartości
- jakie może przyjąć stała, zmienna, identyfikator, argument lub wynik funkcji.

całkowity	zmiennoprzecinkowy
znakowy	tekstowy
wyliczeniowy	logiczny
wskaźnikowy	

8 bitów
16 bitów
32 bity
64 bity
80 bitów
128 bitów
n bitów

tablica	rekord
obiekt	

Typ – w języku programowania jest to definicja określająca:

- rodzaj,
 - strukturę,
 - zakres wartości
- jakie może przyjąć stała, zmienna, identyfikator, argument lub wynik funkcji.

całkowity	zmiennoprzecinkowy
znakowy	tekstowy
wyliczeniowy	logiczny
wskaźnikowy	

tablica	rekord
obiekt	

8 bitów
16 bitów
32 bity
64 bity
80 bitów
128 bitów
n bitów

Zakres danych zależy od długości słowa systemu obliczeniowego

Typ całkowity

Typ całkowity – reprezentuje liczbę całkowitą z pewnego zakresu liczb całkowitych dodatnich i ujemnych.

Standard języka C określa następujące zależności pomiędzy typami całkowitymi:

- typ całkowity prosty ma długość 16 bitów,
- typ całkowity jest co najmniej takiego rozmiaru, jak typ prosty,
- typ całkowity rozszerzony o podwójnej precyzji ma minimum 64 bity.

W praktyce współczesne kompilatory przeznaczone dla maszyn 32 bitowych stosują:

- typ całkowity prosty ma długość 16 bitów,
- typ całkowity ma 32 bity,
- typ całkowity rozszerzony ma 64 bity.

Typy całkowite w językach programowania:

Język C		Język Pascal	
<ul style="list-style-type: none"> ▪ [unsigned] char <i>reprezentuje: liczbę lub znak</i> 		<ul style="list-style-type: none"> ▪ Char <i>reprezentuje: znak</i> 	
rozmiar	: 8 bitów	rozmiar	: 8 bitów
zakres	: -128 do +127 (ze znakiem) 0 do +255 (bez znaku - unsigned)	zakres	: #0..0-9..A-Z..a-Z..#255 – tablica ASCII
<ul style="list-style-type: none"> ▪ [unsigned] short int <i>reprezentuje: liczbę</i> 		<ul style="list-style-type: none"> ▪ Integer <i>reprezentuje: liczbę</i> 	
rozmiar	: 16 bitów	rozmiar	: 16 bitów
zakres	: -32 768 do +32 767 (ze znakiem) 0 do +65 535 (bez znaku)	zakres	: -32 768 do +32 767
<ul style="list-style-type: none"> ▪ [unsigned] int, [unsigned] long int <i>reprezentuje: liczbę</i> 		<ul style="list-style-type: none"> ▪ LongInt <i>reprezentuje: liczbę</i> 	
rozmiar	: 32 bity	rozmiar	: 32 bity
zakres	: -2147483648 do +2147483647 (ze znak.) 0 do +4294967295 (bez znaku)	zakres	: -2147483648 do +2147483647
<ul style="list-style-type: none"> ▪ [unsigned] long long int <i>reprezentuje: liczbę</i> 		<ul style="list-style-type: none"> ▪ Int64 (ObjectPascal) <i>reprezentuje: liczbę</i> 	
rozmiar	: 64 bity	rozmiar	: 64 bity
zakres	: -9223372036854775808 do +9223372036854775807 (ze znakiem) 0 do +18446744073709551615 (bez znaku).	zakres	: -9223372036854775808 do +9223372036854775807

Przykłady użycia typów całkowitych w językach programowania:

Język C	Język Pascal
<pre>int liczba; unsigned long odleglosc; ... liczba = -100; odleglosc = 30000000; ... ---</pre>	<pre>var liczba: Integer; odleglosc: LongInt; ... liczba := 100; odleglosc := 30000000; ... ---</pre>
<pre>char wartosc; ... wartosc = 10; wartosc = '\n'; ... ---</pre>	<pre>var wartosc: Byte; ... wartosc := 10; ... ---</pre>
<pre>unsigned int liczba_unsigned_int; long liczba_long; ... int liczba_unsigned_int = 1234; liczba_long = liczba_unsigned_int; ... ---</pre>	<pre>var liczba_int: Integer; liczba_long: LongInt; ... liczba_int := 1234; liczba_long := liczba_int; ... ---</pre>
<pre>short int liczba_short_int; ... liczba_short_int = -128; liczba_short_int = 127; ... ---</pre>	<pre>var liczba_short_int: ShortInt; ... liczba_short_int = -128; liczba_short_int = 127; ... ---</pre>

Typ zmiennoprzecinkowy

Typ zmiennoprzecinkowy – reprezentuje liczbę dodatnią i ujemną z pewnego zakresu liczb rzeczywistych.

Sposób zapisu typów zmiennoprzecinkowych reguluje standard IEEE-754, jednakże istnieją praktyczne implementacje tych typów odbiegające od standardu, np. 6-bitowy typ Real z języka Pascal. Wg IEEE-754 typ float jest 32 bitowy, typ double 64 bitowy, a long double jest 80 bitowy, co odpowiada pierwszej implementacji koprocesora arytmetycznego (8087).

Forma liczby zmiennoprzecinkowej jest następująca: $Z = (-1)^S \cdot M \cdot 2^{E - bias}$, gdzie: S – znak (1 bit), M – mantysa (23(single), 52(double), 64(long double) bitów), E-bias – wykładnik (8, 11, 15 bitów). bias – przesunięcie wykładnika (127(7Fh), 1023(3FFh) i 16383(3FFFh)).

Wartości specjalne typu zmiennoprzecinkowego:

Wartość	Znaczenie
NaN	Not-a-Number – wynik operacji jest niedozwolony, np. po pierwiastkowaniu wartości ujemnej.
sNaN	signalling NaN – sygnalizowane NaN, powoduje zgłoszenie wyjątku.
qNaN	quiet NaN – ciche NaN, przekazanie tej wartości jako argumentu nie powoduje wyjątku, stosowane np. w operacjach SSE, gdzie można ustawić, że qNaN jest traktowane jako wartość dążąca do zera.
Zero	rozdziela się zero+ i zero-
Inf	Infinite – nieskończoność, rozdziela się + i -

Typy zmiennoprzecinkowe w językach programowania:

Język C		Język Pascal	
<ul style="list-style-type: none"> float 		<ul style="list-style-type: none"> Single 	
rozmiar	: co najmniej 32 bity	rozmiar	: 32 bity
zakres	: 7-8 cyfr, $1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$ dla 32 bitów	zakres	: 7-8 cyfr, $1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{38}$
<ul style="list-style-type: none"> double 		<ul style="list-style-type: none"> Real 	
rozmiar	: 64 bity	rozmiar	: 48 bitów
zakres	: 15-16 cyfr, $5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{308}$	zakres	: 11-12 cyfr, $2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{38}$
<ul style="list-style-type: none"> long double 		<ul style="list-style-type: none"> Extended 	
rozmiar	: 64-128 bitów (80 typowe implementacje)	rozmiar	: 80 bitów
zakres	: 19-20 cyfr, $3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$	zakres	: 19-20 cyfr, $3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{4932}$

Przykłady użycia typów zmiennoprzecinkowych w językach programowania:

Język C

```
float liczba_f_a;
float liczba_f_b;
...
liczba_f_a = -100.1234;
liczba_f_b = 100.1234;
...
---
```

```
double liczba_d_a;
double liczba_d_b;
float liczba_f;
...
liczba_d_a = -123456.0987654321;
liczba_f = 100.00;
liczba_d_b = liczba_f;
...
---
```

```
float liczba_f = 200.00;
double liczba_d;
long double liczba_ld_a;
long double liczba_ld_b;
...
liczba_d = liczba_f;
liczba_ld_a = liczba_f * liczba_d;
liczba_ld_b = liczba_ld_a;
...
```

Język Pascal

```
var liczba_s_a: Single;
    liczba_s_b: Single;
...
liczba_s_a = -100.1234;
liczba_s_b = 100.1234;
...
---
```

```
var liczba_r_a: Real;
    liczba_r_b: Real;
    liczba_s: Single;
...
liczba_r_a = -123456.098765;
liczba_s = 100.00;
liczba_r_b = liczba_s;
...
---
```

```
var liczba_s: Single;
    liczba_r: Real;
    liczba_e: Extended;
...
liczba_s = 200;
liczba_r = liczba_s * 123.334455;
liczba_e = liczba_r;
...
```


Typ znakowy i tekstowy

Typ znakowy – służy do przechowywania znaków.

Typ znakowy w językach programowania:

Język C		Język Pascal	
▪ char		▪ Char	
rozmiar	: 8,16 bitów	rozmiar	: 8 bitów
zakres	: tablica kodów ASCII	zakres	: tablica kodów ASCII

W języku C typ znakowy reprezentuje jednocześnie liczbę.

W zależności od implementacji typ znakowy może przechowywać znak zapisany w kodzie ASCII, Unicode, itp.

Typ tekstowy – służy do przechowywania ciągów znakowych.

Typ tekstowy w językach programowania:

Język C		Język Pascal	
▪ <code>*char, char[], char[n]</code>		▪ <code>String, String[n]</code>	
rozmiar	: ~	rozmiar	: 256 bajtów
zakres	: tablica kodów ASCII	zakres	: tablica kodów ASCII

W języku C:

- ciąg znaków rozpoczyna się od adresu 0 i kończy znakiem #0.

W języku Pascal:

- ciąg znaków rozpoczyna się od adresu 1, pierwszy bajt zawiera informacje o liczbie znaków w ciągu.

W języku ObjectPascal:

- wprowadzono typ `ShortString` – odpowiednik `String` z Pascala, oraz nowe typy `String`, `AnsiString` i `WideString` (Unicode), a także `PChar` odpowiednik `char*` z języka C.

Przykłady użycia typów znakowego i tekstowego w językach programowania:

Język C	Język Pascal
<pre>char znak, ... znak = 'a'; znak = 100; ... ---</pre>	<pre>var znak: Char; ... znak = 'a'; znak := #32; ... ---</pre>
<pre>char ciag_znakow[20] = "ciąg znakowy"; ... ---</pre>	<pre>var ciag_znakowy: String[20]; ... ciag_znakowy := "ciąg znakowy"; ... ---</pre>
<pre>char ciag_znakow[128]; ... strcpy(ciag_znakowy, "ciąg znakowy"); ...</pre>	<pre>var ciag_znakow_a: String; ciag_znakow_b: String; ciag_znakow_c: String; ... ciag_znakow_a := "ciąg znakow a"; ciag_znakow_b := "ciąg znakow b"; ciag_znakow_c := ciag_znakow_a + ciag_znakow_b; ... ---</pre>
<pre>char *ciag_znakow; ... ciag_znakow = malloc(128); strcpy(ciag_znakow, "ciąg znakow"); ... free(ciag_znakow); ... ---</pre>	

Typ wyliczeniowy i logiczny

Typ wyliczeniowy – określa jakie wartości może przyjmować zmienna.

Typ wyliczeniowy w językach programowania:

Język C		Język Pascal	
▪ enum nazwa_typu {wartość, wartość, ... wartość[= liczba], ...};		▪ nazwa_wartości = wartość;	
rozmiar	: ~	rozmiar	: ~
zakres	: zbiór zawężony do zdefiniowanych wartości	zakres	: zbiór zawężony do zdefiniowanych wartości

W językach programowania typ wyliczeniowy jest wykorzystywany do poprawy czytelności kodu. Ponadto pozwala na tworzenie stałych w czasie kompilacji.

Typ logiczny – typ określający wartości typu prawda lub fałsz.

Typ logiczny w językach programowania:

Język C		Język Pascal	
▪ bool		▪ Boolean	
rozmiar	: 1 bajt	rozmiar	: 1 bajt
zakres	: 0 lub dowolna wartość niezerowa	zakres	: true, false

W języku C typ logiczny może nie być jawnie zdefiniowany. Ponadto odpowiednikiem wartości nieprawdziwej jest wartość liczbowa 0, a każda wartość inna niż zero traktowana jest jako prawda.

Przykłady użycia typów wyliczeniowego i logicznego w językach programowania:

Język C

```
bool znacznik,  
...  
znacznik = 0;  
znacznik = FALSE;  
znacznik = TRUE;  
znacznik = 1;  
...  
---  
  
enum Boolean {  
    true, false  
};  
Boolean znacznik;  
...  
znacznik = true;  
znacznik = false;  
...  
---  
  
enum Tydzien {  
    nie, pon, wto, sro, czw, pia, sob  
};  
Tydzien dzien;  
...  
dzien = pon;  
...  
---
```

Język Pascal

```
var znacznik: Boolean;  
...  
znacznik := true;  
znacznik := false;;  
...  
---  
  
type Bool = { true, false };  
var znacznik: Bool;  
...  
znacznik := true;  
znacznik := false;  
...  
---  
  
type Tydzien = { nie, pon, wto, sro, czw, pia, sob };  
var dzien: Tydzien;  
...  
dzien := nie;  
dzien := czw;  
...  
---
```

Typ tablicowy



Typ tablicowy – służy do reprezentacji tablic danych.

Typ tablicowy w językach programowania:

Język C		Język Pascal	
▪ <code>typ nazwa_tablicy[n]</code>		▪ <code>nazwa_tablicy: array[n..m] of typ;</code>	
rozmiar	: n słów o rozmiarze typu danych tablicy	rozmiar	: n słów o rozmiarze typu danych tablicy
zakres	: zakres zmiennych typu danych tablicy	zakres	: zakres zmiennych typu danych tablicy

W języku C zmienna tablicowa jest jednocześnie adresem pierwszego elementu tablicy. Ponadto tablica jest indeksowana od 0 do n-1.

W języku Pascal indeksowanie tablicy przebiega od n do m.

W obu przypadkach jako indeksy można stosować dowolne typy całkowite lub wyliczeniowe.

Przykłady deklaracji typu tablicowego w językach programowania:

Język C	Język Pascal
<pre>typedef char tab[11]; tab T; ... T[0] = 'a'; T[9] = 'b'; ...</pre>	<pre>type tab = array[0..10] of Char; var T: tab; ... T[0] := 'a'; T[9] := 'b'; ...</pre>
<pre>typedef int tab[128][2]; tab L; ... L[0][0] = 1; L[0][1] = 2; L[1][0] = 2; L[1][1] = 2; L[2][0] = 3; L[2][1] = 2; ...</pre>	<pre>type tab = array[-10..20,1..2] of Integer; var L: tab; ... L[-10,1] := 1; L[-10,2] := 2; L[1,1] := 2; L[1,2] := 2; L[20,1] := 3; L[20,2] := 2; ...</pre>
<pre>typedef char String[256]; String ciag; ... ciag = "abc"; ...</pre>	<pre>type String = array[0..255] of Char; var ciag: String; ... ciag := 'abc'; ...</pre>

Typ wskaźnikowy

Typ wskaźnikowy – służy do przechowywania adresu pamięci.

Typ wskaźnikowy w językach programowania:

Język C		Język Pascal	
▪ *		▪ Pointer	
rozmiar	: ~	rozmiar	: ~
zakres	: ~	zakres	: ~

Rozmiar i zakres wartości typów wskaźnikowych zależą od:

- architektury systemu komputerowego,
- systemu operacyjnego,
- kompilatora programów.

Przykłady użycia typu wskaźnikowego w językach programowania:

Język C

```

int *wsk_int;
int liczba = 100;
...
wsk_int = &liczba;
*wsk_int = 200;           // → liczba jest równa 200
...
---

char *wsk_tab;
char tab[100] = "ciąg znakow";
...
wsk_tab = tab;
// lub
*wsk_tab = &(tab[0]);
...
wsk_tab[10] = 'i';
wsk_tab[11] = ' ';       // → tab zawiera "ciąg znaki "
...
*(wsk_tab+1) = 'C';     // → tab zawiera "Ciąg znaki "
...

```

Język Pascal

```

var wsk_int: Pointer;
const liczba: Integer := 100;
...
wsk_int := @liczba;
wsk_int^ := 200;         // → liczba jest równa 200
...
---

var PChar: Pointer;
    tab: array[0..99] of Char;
...
PChar := @tab;
...
PChar^[10] := 'i';
PChar^[11] := ' ';     // → tab zawiera "ciąg znaki "
...

```

Rekordy i obiekty

Struktury/rekordy – służą do reprezentowania typów składających się ze zmiennych różnych typów.

Typ rekordowy w językach programowania:

Język C		Język Pascal	
▪ <code>struct nazwa {typ pola, typ pole ... };</code>		▪ <code>nazwa = record pole:typ; pole:typ; ... end;</code>	
rozmiar	: suma rozmiarów zmiennych	rozmiar	: suma rozmiarów zmiennych
zakres	: ~	zakres	: ~

Przykłady deklaracji typu rekordowego w językach programowania:

Język C	Język Pascal
<pre>struct osoba { char imie[20]; char nazwisko[20]; int wiek; }; osoba os; ... strcpy(&os.imie, "Jan"); strcpy(&os.nazwisko, "Kowalski"); os.wiek = 33; ...</pre>	<pre>type osoba = record imie: String; nazwisko: String; wiek: Integer; end; var os: osoba; ... os.imie := "Jan"; os.nazwisko := "Kowalski"; os.wiek := 33; ...</pre>

Obiekty – są to złożone struktury danych zawierające pola (zmienne) oraz metody (funkcje) przetwarzające dane pobierane z pól.

Typ obiektowy w językach programowania:

Język C		Język Pascal	
<pre> ▪ class nazwa_typu { [private:][protected:][public:] typ pole; typ pole; ... [definicja constructor'a;] [definicja destructor'a;] [definicja funkcji;] [definicja funkcji;] ... }; </pre>		<pre> ▪ nazwa = object [private:][protected:][public:] pole: typ; pole: typ; ... [definicja constructor'a;] [definicja destructor'a;] [definicja funkcji;] [definicja funkcji;] ... end; </pre>	
rozmiar	: suma rozmiarów zmiennych	rozmiar	: suma rozmiarów zmiennych
zakres	: ~	zakres	: ~

Pod względem funkcjonalnym obiekt przypomina rekord/strukturę, jednak w obiektach występują mechanizmy dziedziczenia, hermetyzacji, ochrony itp..

Koniec wykładu