

Wykład II

Tablice (wstęp)

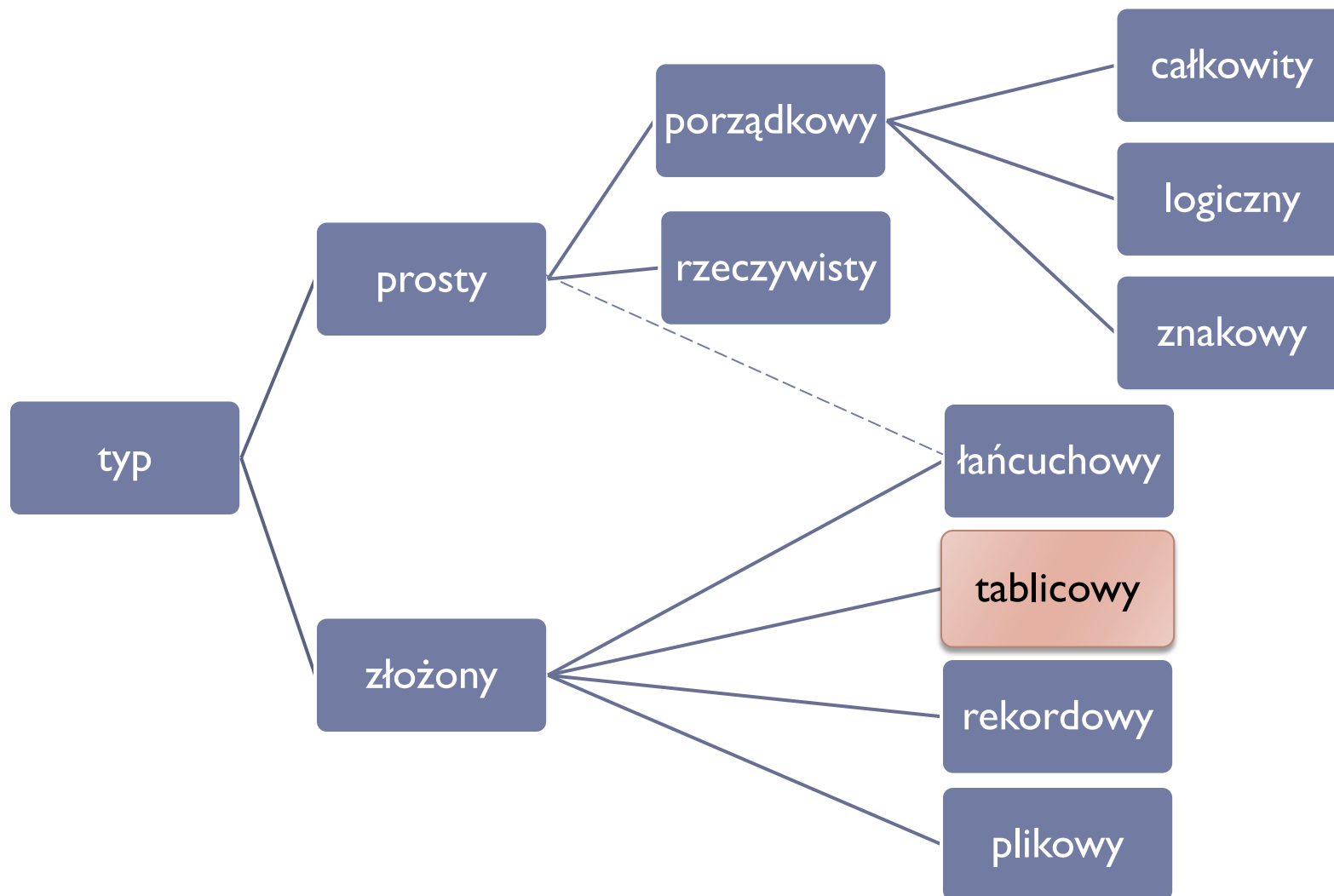
Przykłady algorytmów

Wstęp do języka C/C++



Wstęp do struktur danych: Tablice

Klasyfikacja zmiennych statycznych



Rodzaje tablic

Tablice jednowymiarowe (wektory)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które ponumerowano liczbami naturalnymi – każda z nich ma przypisany na stałe tzw. indeks,
- mogą przechowywać nie większą od ich długości liczbę elementów zbioru danych jednakowego typu.

Tab								- nazwa tablicy
34	56	32	-8	45	2	0	13	- wartości
Tab(0)	Tab(1)	Tab(2)	Tab(3)	Tab(4)	Tab(5)	Tab(6)	Tab(7)	- pole – nazwa(indeks)

W zapisie symbolicznym $T(6)$ oznacza 6 zmienną w tablicy T

Indeks może być określony przez bezpośrednie podanie wartości w odwołaniu do elementu tablicy, np. $T(6)$, lub użycie nazwy zmiennej o typie zgodnym z indeksem, np. $T(X)$. Zmienną X nazywamy wtedy zmienną indeksową i wskazanie elementu tablicy wymaga odczytania jej aktualnej wartości.

Rodzaje tablic

Tablice dwu – i więcej wymiarowe (macierze)

- są zespołem określonej liczby zmiennych o wspólnej nazwie, które oznaczono dwoma lub więcej indeksami,
- mogą przechowywać nie większą od ich rozmiaru liczbę elementów zbioru danych jednakowego typu.

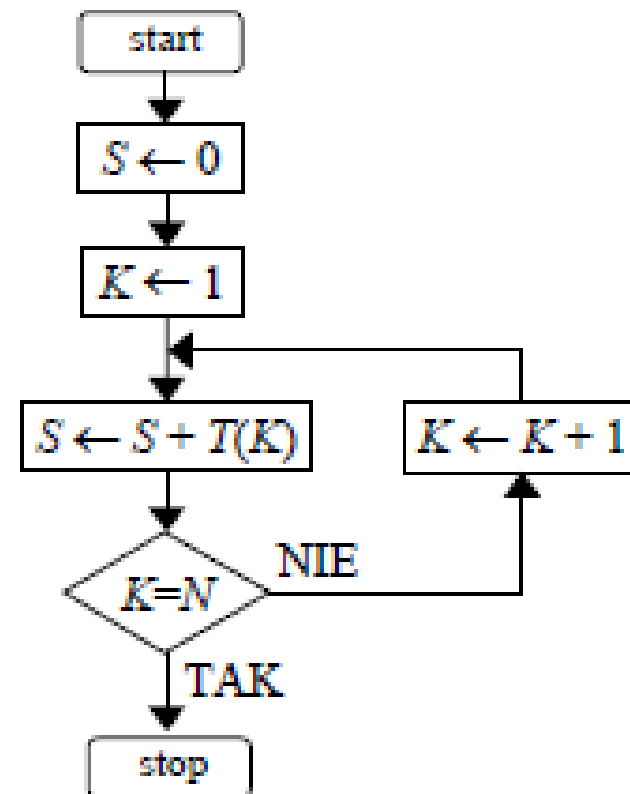
	0	1	2	3	4
0					
1					
2					
3					
4					

W zapisie symbolicznym $W(3, 5)$ oznacza zmienną w tablicy W położoną umownie na przecięciu 3. wiersza i 5. kolumny.

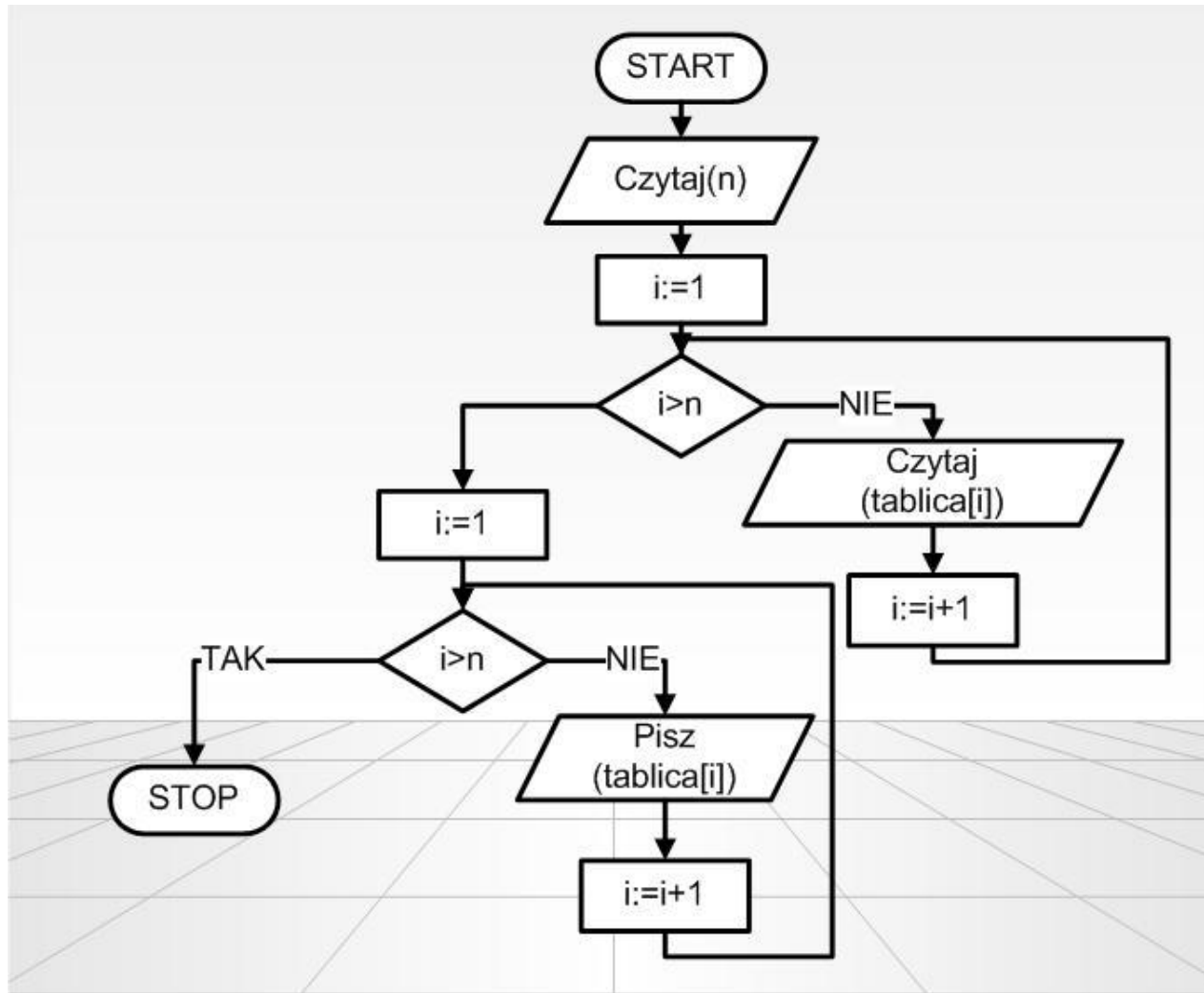
Obsługa tablicy jednowymiarowej

Algorytm sumowania N liczb zapamiętanych w tablicy T

1. $S \leftarrow 0$ (ustalenie początkowej wartości sumy);
2. $K \leftarrow 1$ (ustalenie początkowej wartości zmiennej indeksowej);
3. wykonaj co następuje N razy:
 - 3.1. $S \leftarrow S + T(K)$;
 - 3.2. $K \leftarrow K + 1$.



Obsługa tablicy jednowymiarowej



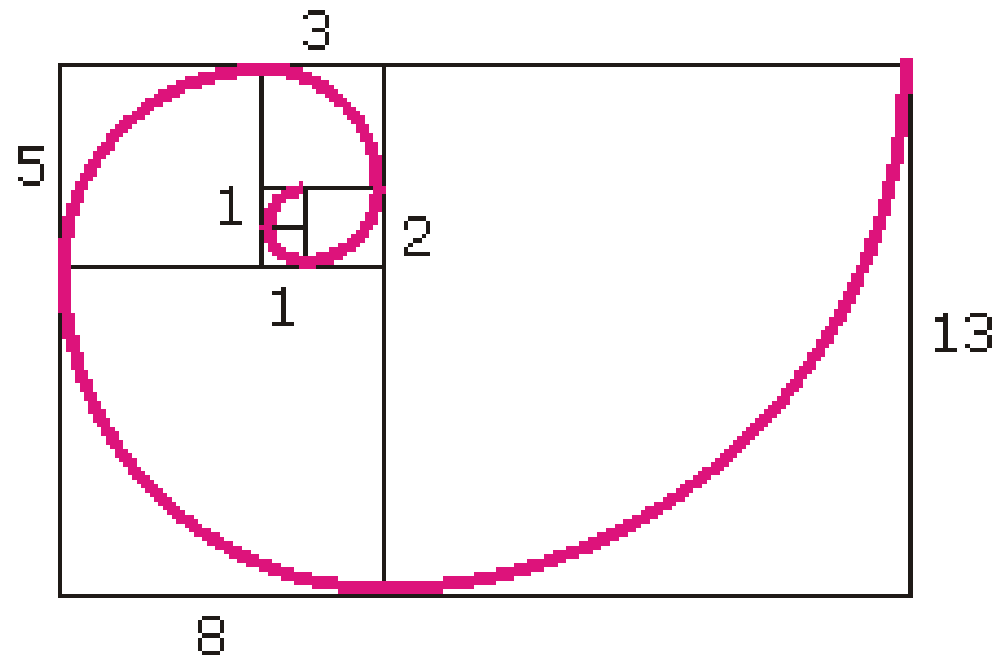


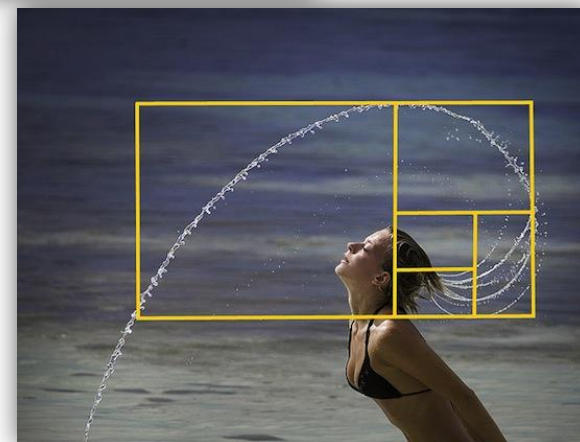
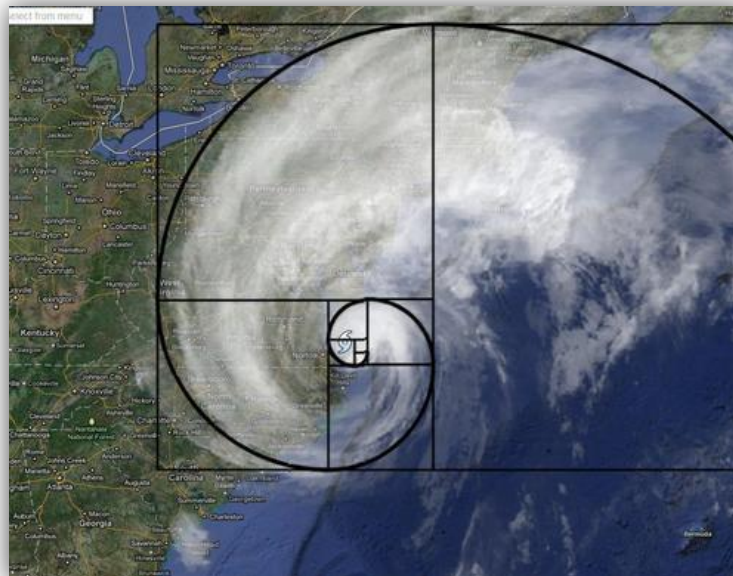
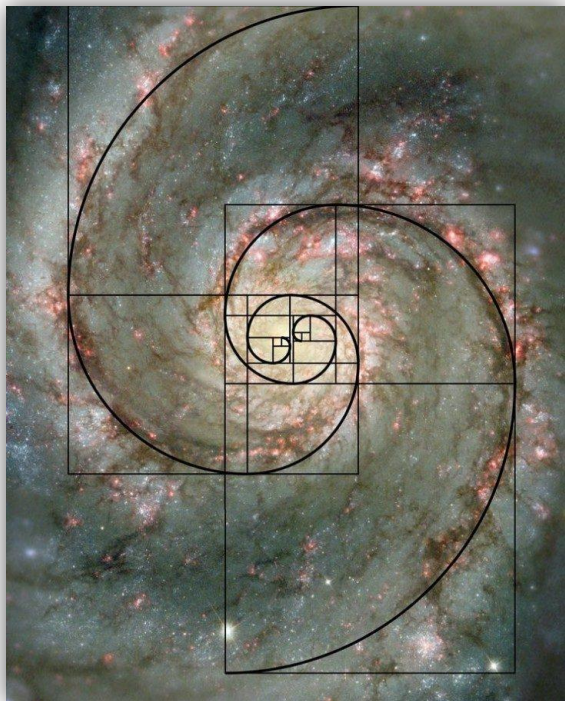
Przykłady algorytmów

Ciąg Fibonacciego

Ciąg Fibonacciego to ciąg liczb, w którym:

- pierwszy wyraz jest równy 1,
- drugi jest równy 1,
- każdy następny jest **sumą dwóch poprzednich**.





Ciąg Fibonacciego



goldenratio

$\frac{a}{b} = \frac{a'}{b'} = 1.618$
goldenratio

the concentric circles are in goldenratio

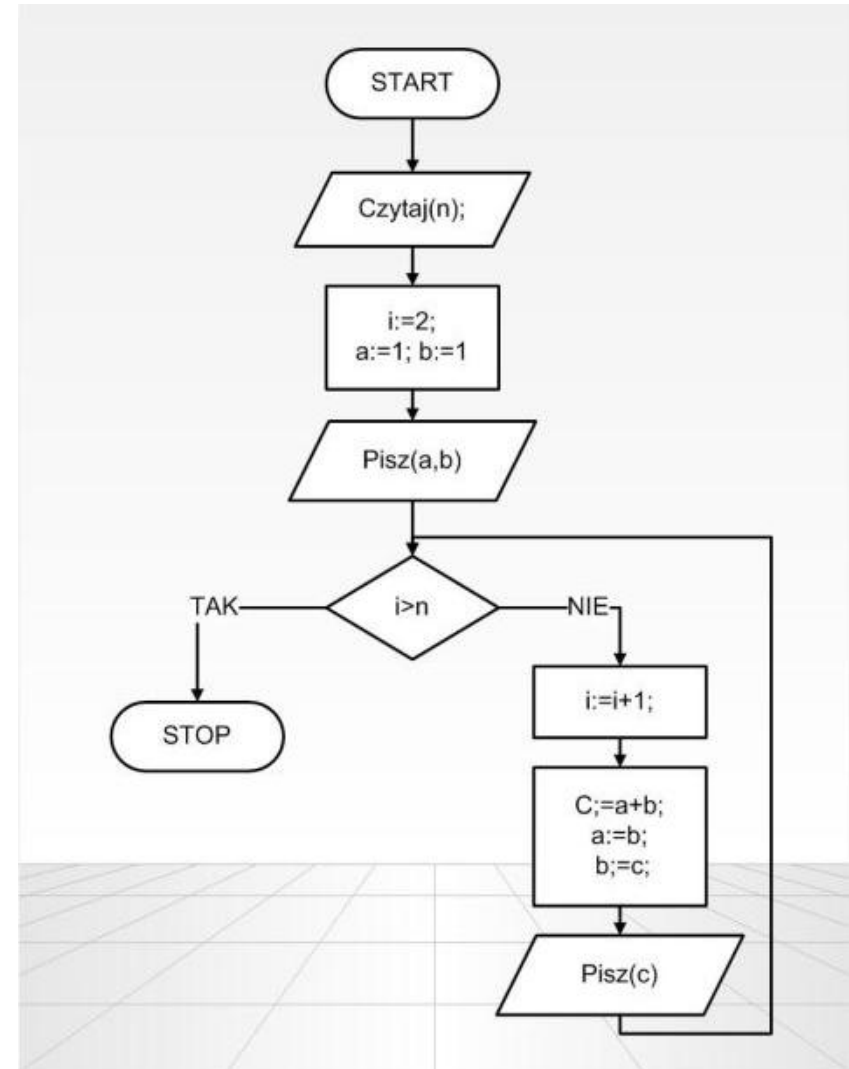
$\frac{a}{b} = 1.618$ goldenratio

$\frac{a}{b} = 1.61$ goldenratio

Ciąg Fibonacciego



- pierwszy wyraz jest równy 1,
- drugi jest równy 1
- każdy następny jest **sumą dwóch poprzednich**.



Trochę historii

Algorytm Euklidesa – obliczanie największego wspólnego dzielnika dwóch liczb naturalnych.

$$NWD(a, b) = \begin{cases} a & \text{dla } b = 0 \\ NWD(b, a \bmod b) & \text{dla } b \geq 1 \end{cases}$$

1. Dane są dwie liczby naturalne a i b.
2. Oblicz c jako resztę z dzielenia a przez b
3. Zastąp a przez b, zaś b przez c.
4. Jeżeli b = 0, to szukane NWD wynosi a, w przeciwnym wypadku wróć do punktu drugiego i kontynuuj.

NWD(liczba całkowita a, liczba całkowita b)

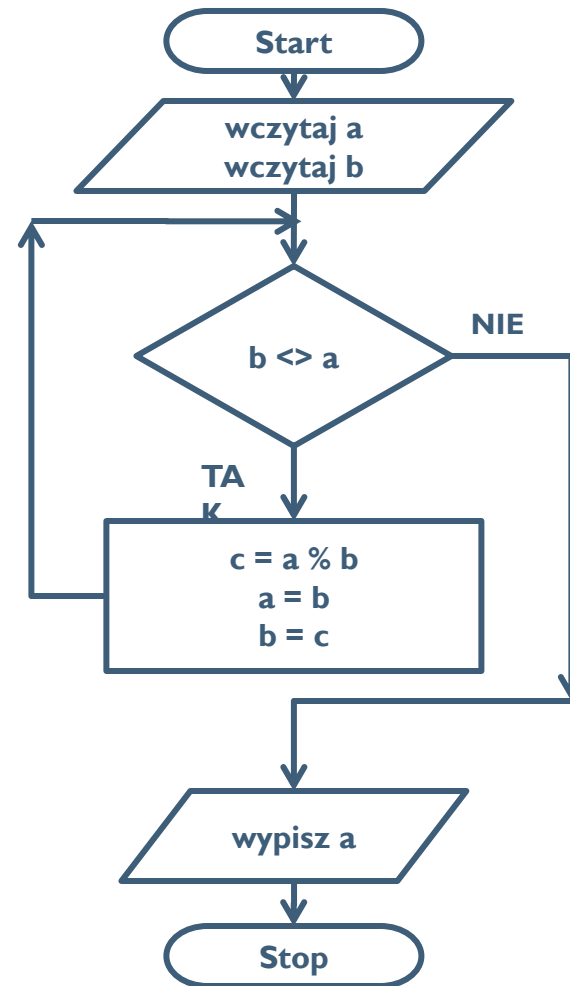
dopóki b różne od 0

c = **reszta z dzielenia a przez b**

a = b

b = c

zwróć a

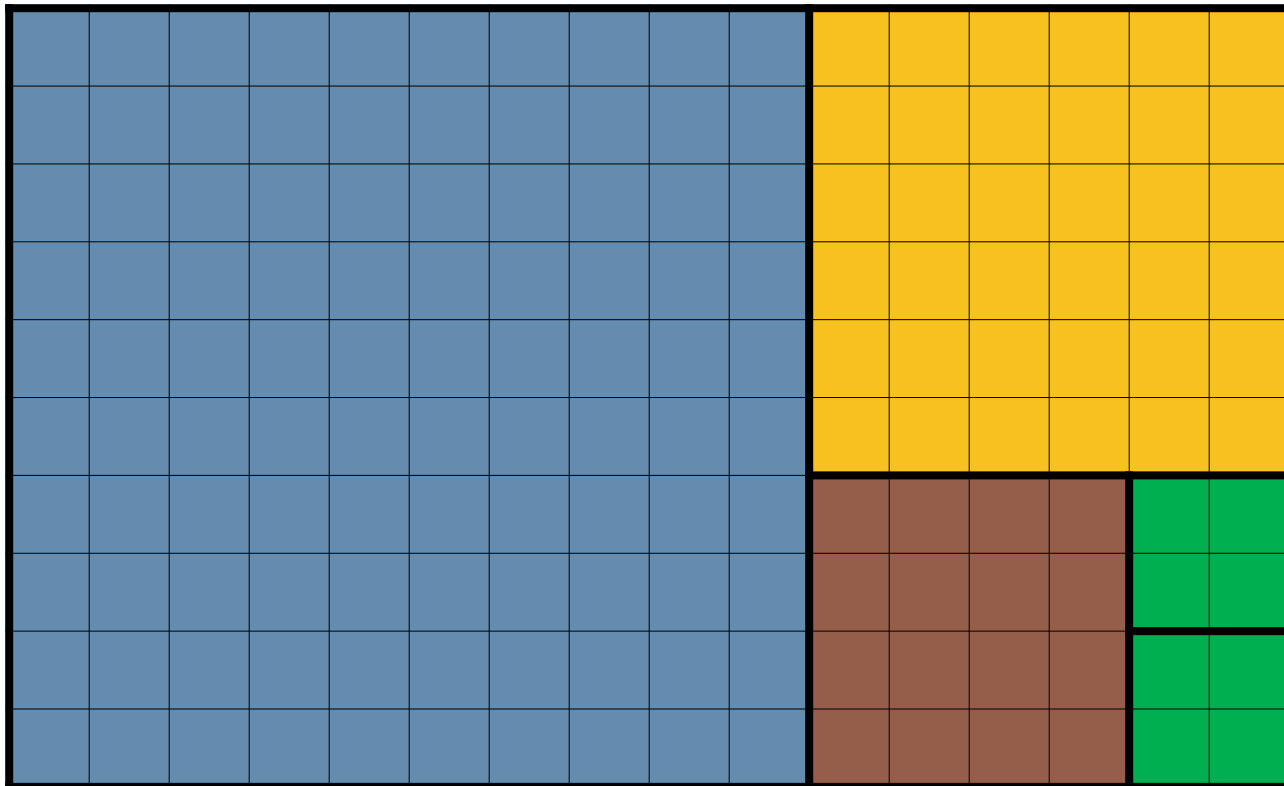


Trochę historii

Algorytm Euklidesa – obliczanie
największego wspólnego dzielnika
dwóch liczb naturalnych.

16

10





Budowa programu



Kod źródłowy - program napisany w języku takim jak Pascal lub C++, czyli w języku algorytmicznym - czytelny dla programisty

Kod wynikowy - program zapisany jako ciąg rozkazów i danych w kodzie maszynowym procesora (w postaci czytelnej dla komputera), najczęściej w postaci liczb kodu dwójkowego

Proces tworzenia programu:

- | | | | |
|--------------|---|----------------|---------------------------------------|
| • edytor | - | (*.cpp) | kod źródłowy |
| • kompilator | - | (obj) | kod wynikowy |
| • Linker | - | (*.exe) | kod wynikowy połączony z bibliotekami |
| • debugger | - | (step/watch) | śledzenie działania, usuwanie błędów |

Wybrane środowiska programistyczne dla C++

- Dev C++



- Code::Blocks



- MS Visual Studio



Pierwszy program (w języku C)

```
1  #include <conio.h>
2  #include <cstdio>
3
4  using namespace std;
5
6  int main()
7  {
8      printf("To nasz pierwszy program w C\n");
9      getch();
10     return 0;
11 }
```

Pierwszy program (w języku C++)

```
1  #include <iostream>
2  #include <conio.h>
3
4  using namespace std;
5
6  int main()
7  {
8      cout << "Hello world!" << endl;
9      getch(); //zatrzymuje działanie programu
10     return 0;
11 }
```

Budowa programu

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      return 0;
8  }
```

Dołączanie plików nagłówkowych bibliotek

Przestrzeń nazw

return...; gdzie za kropki wstawiamy dowolną liczbę - kod wyjścia programu.

Funkcja main() - główna funkcja programu



Int main() – w języku C i C++ nie ma „programu głównego” jest za to funkcja o nazwie main() która wykonywana jest zawsze jako pierwsza.

Każdy program musi posiadać funkcję main()



W językach C i C++ mamy do dyspozycji trzy rodzaje komentarzy:

- komentarz
jednowierszowy;

```
5  int main()  
6  {  
7      // Komentarz jednowierszowy  
8      return 0;  
9  }
```

- komentarz
wielowierszowy;

```
5  int main()  
6  {  
7      /* początek komentarza  
8         komentujemy dalej....  
9         i kończymy komentarz */  
10     return 0;  
11 }
```

- komentarz wykonany za pomocą dyrektyw preprocesora.



Zmienne (wstęp)



Nazwa typu	Zawartość	Przedział wartości	Zajęt. pamięć
char	znak	$-128 \div 127$	1 bajt
int	liczba całkowita	$-32768 \div 32767$	2 bajty
long	liczba całkowita	$-2147\text{mln} \div 2147\text{mln}$	4 bajty
float	liczba rzeczyw.	$10^{-38} \div 10^{38}$ (7cyfr)	4 bajty
double	liczba rzeczyw.	$10^{-308} \div 10^{308}$ (15 cyfr)	8 bajtów

**Modyfikatory typu:**

signed	→	ze znakiem (\pm),	<u>int</u>	<u>char</u>	—
unsigned	→	bez znaku,	int	char	—
short	→	krótka (mniejsza),	int	—	—
long	→	długa (większa)	<u>int</u>	—	double

np. **unsigned long int** *dluga_liczba_bez_znaku* ;

Wartości domyślne:

long	=	long int
int	=	signed int
char	=	signed char



Deklaracja zmiennej - informuje kompilator, że dana nazwa jest znana. Jednak pamięć dla obiektu nie zostaje przydzielona. Do obiektu nie możemy się odwoływać, nie możemy mu przypisywać wartości – obiekt jeszcze nie istnieje.

extern nazwaTypu nazwaZmiennej;

Np.: `extern int liczba;`



Definicja zmiennej - rezerwuje miejsce w pamięci dla danej zmiennej. Po zdefiniowaniu ze zmiennej możemy korzystać.

nazwaTypu nazwaZmiennej;

Np.: int liczba;

Każda definicja jest jednocześnie deklaracją (ale nie odwrotnie).



Inicjalizacja (inicjowanie) zmiennej - polega na przypisaniu wartości do danej zmiennej w momencie jej deklaracji

nazwaTypu nazwaZmiennej = wartość;

Np.: `int liczba = 10;`



Operacje we/wy

Funkcje printf i scanf (proceduralnie, w C)

Wysłanie informacji na zewnątrz (stand. ekran)

```
printf („lancuch formatujacy”, zmienna_1, zmienna_2);
```

Pobranie informacji z zewnątrz (stand. klawitura)

```
scanf („prototypy zmiennych”, &zmienna_1, &zmienna_2);
```

Funkcje printf i scanf (proceduralnie, w C)

Prototypy zmiennych dla funkcji printf i scanf

%c	-	pojedynczy znak
%s	-	łańcuch znaków
%d	-	liczba dziesiętna ze znakiem
%f	-	liczba zmiennoprzecinkowa (notacja dziesiętna)
%e	-	liczba zmiennoprzecinkowa (notacja wykładnicza)
%g	-	liczba zmiennoprzecinkowa (krótszy z formatów %f %e)
%u	-	liczba dziesiętna bez znaku
%x	-	liczba w kodzie szesnastkowym (bez znaku)
%o	-	liczba w kodzie ósemkowym (bez znaku)
l	-	przedrostek l (long) stosowany przed: d u x o

Funkcje printf i scanf (proceduralnie, w C)

Znaki sterujące wypisywaniem tekstu (nie tylko dla printf)

`\b` - cofanie o 1 znak
`\f` - nowa strona
`\n` - nowa linia
`\t` - tabulator
`\a` - sygnał dźwiękowy

Jeśli jednak chcemy po prostu wypisać znak...

`\\` - bekslesz
`\'` - apostrof
`\0` - znak o kosie zero
`\?` - znak zapytania

Funkcje printf i scanf (proceduralnie, w C)

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main()
5  {
6      double a, b, c, delta, x1, x2;
7      printf("a = "); scanf("%lf", &a);
8      printf("b = "); scanf("%lf", &b);
9      printf("c = "); scanf("%lf", &c);
10     if ((delta = b*b-4*a*c) >= 0)
11     {
12         x1 = (-b-sqrt(delta))/(2*a);
13         x2 = (-b+sqrt(delta))/(2*a);
14         printf("x1 = %lf\nx2 = %lf\n", x1, x2);
15     }
16     else
17         printf("Brak rozwiazan rzeczywistych\n");
18     return 0;
19 }
```

Klasy cout i cin (obiektowo w C++)

Strumień – to najprościej mówiąc jest to ciąg bajtów o nieokreślonej długości.

Wyróżniamy trzy rodzaje strumieni:

1. **Strumień konsoli – wczytanie z klawiatury i wypisanie na ekran**
2. Strumień plikowe
3. Strumień napisów

Do obsługi strumieni służą obiekty **cin** oraz **cout**

Domyślnym strumieniem jest strumień konsoli, którym będziemy posługiwać się w tym wykładzie.

Klasy cout i cin (obiektowo w C++)

Wyprowadzenie wartości do strumienia wyjściowego (stdout)

```
cout << „tekst”;
```

```
cout << zmienna;
```

Wczytanie ze strumienia wejściowego (stdin)

```
cin >> zmienna;
```

Prototypy cin i cout znajdują się w bibliotece iostream.h

```
#include <iostream>
```

Klasy cout i cin (obiektoowo w C++)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Hej tam.\n";
8      cout << "To jest 5: " << 5 << "\n";
9      cout << "Manipulator endl ";
10     cout << "wypisuje nowa linie na ekranie.";
11     cout << endl;
12     cout << "To jest bardzo duza liczba:\t" << 70000;
13     cout << endl;
14     cout << "To jest suma 8 i 5:\t";
15     cout << 8+5 << endl;
16     cout << "To jest ulamek:\t\t";
17     cout << (float) 5/8 << endl;
18     cout << "I bardzo, bardzo duza liczba:\t";
19     cout << (double) 7000 * 7000 << endl;
20     return 0;
21 }
```

Hej tam.

To jest 5: 5

Manipulator endl wypisuje nowa linie na ekranie.

To jest bardzo duza liczba: 70000

To jest suma 8 i 5: 13

To jest ulamek: 0.625

I bardzo, bardzo duza liczba: 4.9e+007



Instrukcje sterujące: Instrukcja IF

Prawda - Fałsz

W języku C++ nie ma osobnych zmiennych przechowujących dane typu prawda-Fałsz.

Tę rolę pełnić może każda zmienna, wyrażenie lub funkcja , która przyjmuje (lub zwraca) wartość zero lub różną od zera.

Wartość zero -

FAŁSZ

Wartość inna niż zero -

PRAWDA

Instrukcja warunkowa if

```
if (wyrażenie) instrukcja;
```

```
if (wyrażenie) instrukcja_1;  
else instrukcja_2;
```

```
if (wyrażenie)  
{  
    instrukcja_1;  
    instrukcja_2;  
}  
else instrukcja_3;
```

Instrukcja warunkowa if

Przykład:

```
cin >> i;  
if (i!=0) cout << „i rozne od zera”;  
else cout << „i rowne zero”;
```

Można i tak:

```
cin >> i;  
if (i) cout << „i rozne od zera”;  
else cout << „i rowne zero”;
```


Literatura:

W prezentacji wykorzystano przykłady i fragmenty:

- Grębosz J.: **Symfonia C++**, **Programowanie w języku C++ orientowane obiektowo**, Wydawnictwo Edition 2000.
- Jakubczyk K.: *Turbo Pascal i Borland C++ Przykłady*, Helion.

Warto zajrzeć także do:

- Kerninghan B.W., Ritchie D. M.: **język ANSI C**, Wydawnictwo Naukowo Techniczne.

Dla bardziej zaawansowanych:

- Grębosz J.: **Pasja C++**, Wydawnictwo Edition 2000.
- Meyers S.: **język C++ bardziej efektywnie**, Wydawnictwo Naukowo Techniczne