

Algoritmy

Funkcje

Wprowadzenie

Mirosław Ochodek

Miroslaw.Ochodek@pwsz.pila.pl

Miroslaw.Ochodek@cs.put.poznan.pl



PAŃSTWOWA
WYŻSZA
SZKOŁA
ZAWODOWA
IM. ST. STASZICA
W PILE

Funkcje (i procedury)

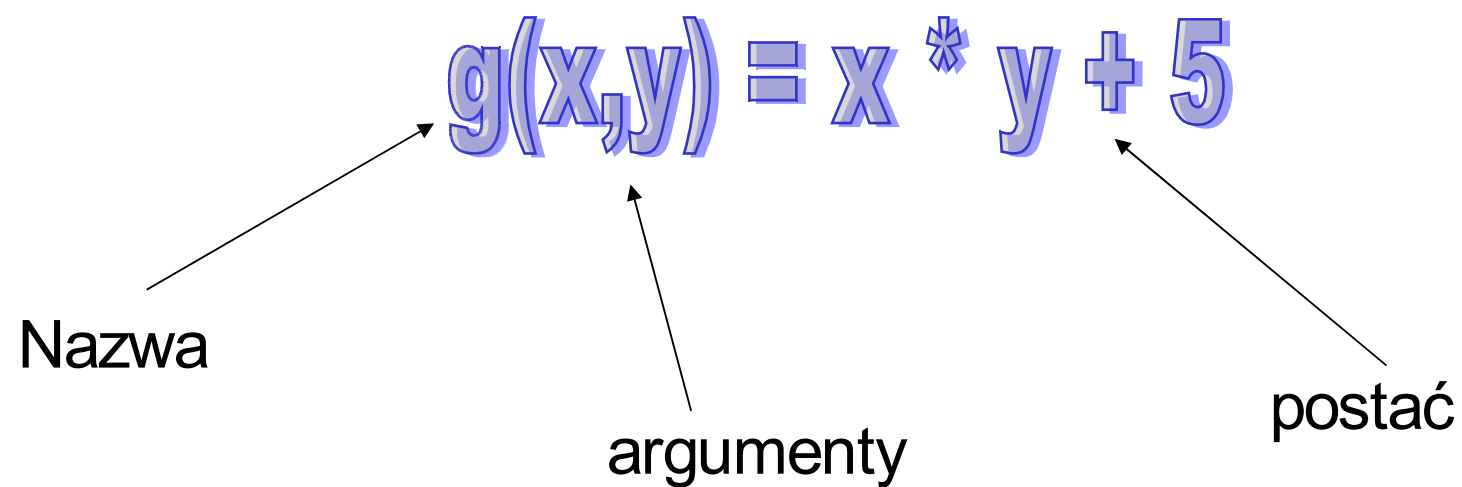
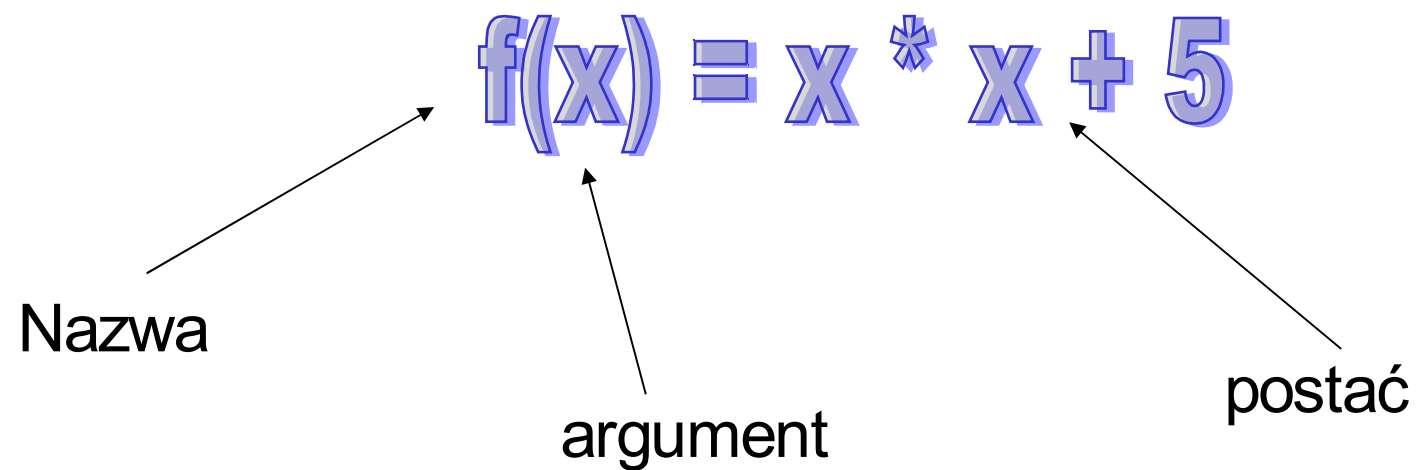
■ Problem

- Zbyt długi ciąg instrukcji sprawia, że kod jest nieczytelny
- Często w kodzie powtarzają się takie same ciągi instrukcji (algorytmy)

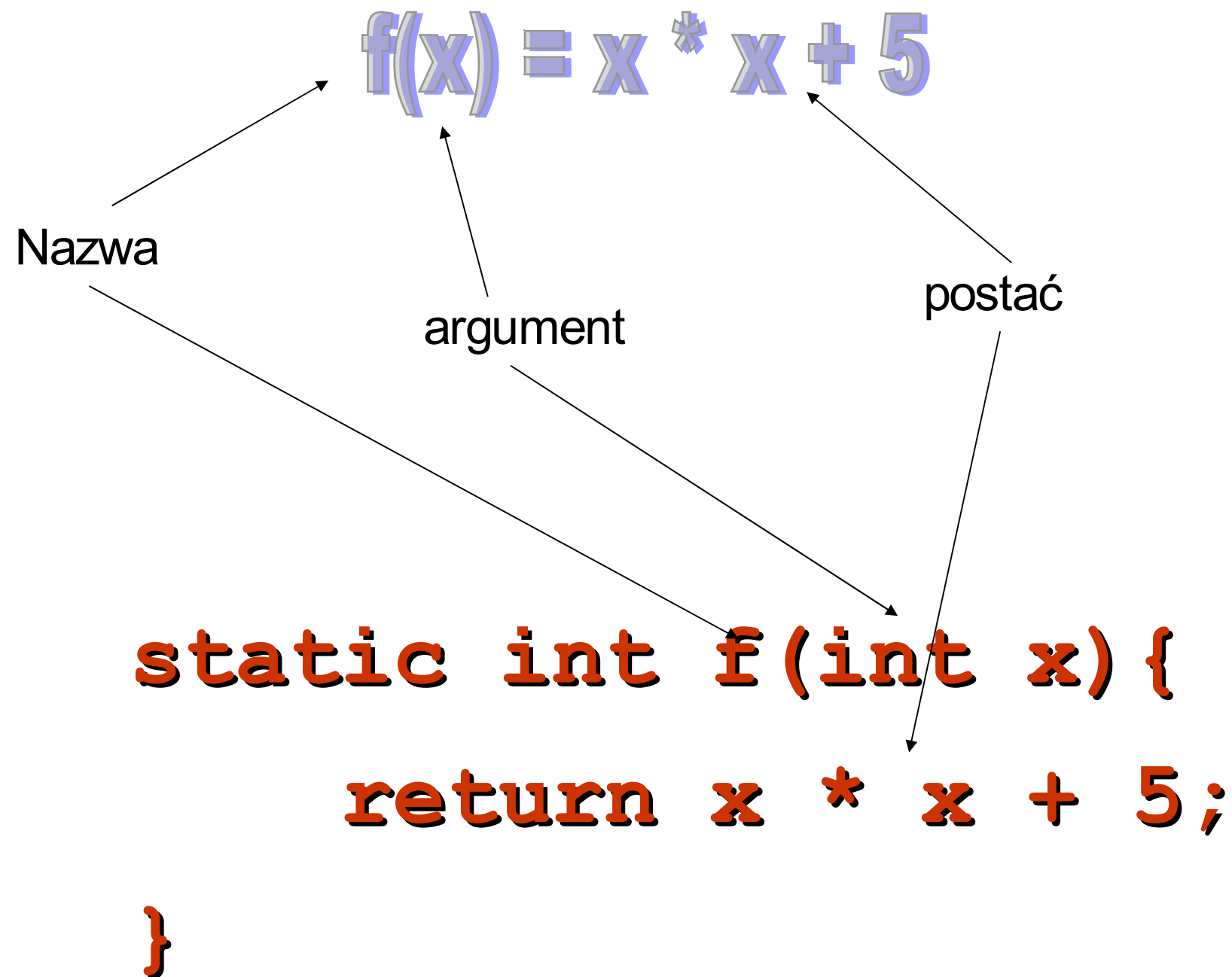
■ Rozwiązanie

- Podziel program na mniejsze podprogramy

Funkcja w matematyce



Funkcje w programowaniu

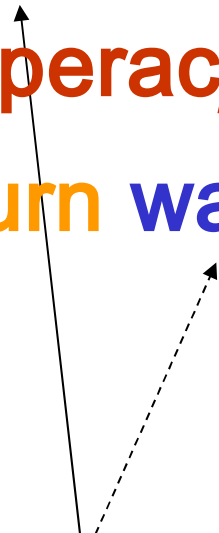


Funkcje - składnia

```
static typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```

Funkcje - składnia

```
static typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```



■ typ_zwracany

- typ wynikowy zwracany z funkcji (wartość takiego typu musi być zwrócona w instrukcji return)
- void – oznacza, że funkcja nie zwraca wartości (zwraca wartość pustą) - procedura

Funkcje - składnia

```
static typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```



■ nazwa funkcji

- Identyfikator funkcji – jej nazwa (prawidłowy identyfikator, czyli np. nie może być spacji w nazwie!)

Funkcje - składnia

```
typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```



- typ argumentu argument
 - oddzielona przecinkami lista argumentów
 - każdy argument: typ nazwa
 - może być funkcja bezparametrowa – pusty nawias

Funkcje - składnia

```
static typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```

- operacje
 - dowolne operacje

Funkcje - składnia

```
static typ_zwracany nazwa_funkcji(typ_argumentu argument, ...){  
    // operacje  
    return wartosc_zwracana;  
}
```

■ return wartość

- oznacza zakończenie w tym miejscu działania funkcji i zwrócenia jako wynik działania wartości
- W przypadku jeśli funkcja zwraca typ void (czyli pustą wartość) wyrażenie return występuje bez zwracanej wartości i oznacza przerwanie działania programu

Przykład

- Napisz program, który sumuje dwie liczby

Przykład

- Stare podejście

```
int n = 5;
```

```
int m = 6;
```

```
int z = n + m;
```

Przykład

- Z funkcją

```
static int suma(int a, int b) {  
    return a+b;  
}
```

```
int n = 5;  
int m = 6;
```

```
int z = suma(n,m)
```

Procedura przykład

- Wypisz zawartość 3 tablic

Procedura przykład

■ Stare podejście

```
int[] tab = new int[5];  
int[] tab2 = new int[7];  
int[] tab3 = new int[10];  
  
for (int i=0; i < tab.length; i++) {  
    System.out.println(tab[i]);  
}  
for (int i=0; i < tab2.length; i++) {  
    System.out.println(tab2[i]);  
}  
for (int i=0; i < tab3.length; i++) {  
    System.out.println(tab3[i]);  
}
```

Procedura przykład

■ Nowe podejście

```
static void wyswietlTab(int[] tablica) {  
    for (int i=0; i < tablica.length; i++) {  
        System.out.println(tablica[i]);  
    }  
}
```

```
int[] tab = new int[5];  
int[] tab2 = new int[7];  
int[] tab3 = new int[10];
```

```
wyswietlTab(tab);  
wyswietlTab(tab2);  
wyswietlTab(tab3);
```

Zadanie

- Napisz funkcję, która sprawdza czy dana liczba jest parzysta
 - zwraca wartość typu boolean
 - przyjmuje jeden parametr – liczbę całkowitą
- Użyj funkcji w programie, który pobiera z wejścia od użytkownika liczbę i wypisuje
 - „Podana liczba jest parzysta”
 - „Podana liczba nie jest parzysta”

Zadanie

- Napisz funkcję, która oblicza średnią z wartości jako w tablicy podanej jako parametr
 - Typ zwracany double
 - Parametr tablica double
- Użyj funkcji w programie pobierającym od użytkownika n-liczb i wyliczających średnią

Zadanie

- Napisz procedurę (funkcja, która zwraca typ void), która wypisze na ekran zawartość tablicy dwuwymiarowej liczb całkowitych

Zadanie

- Napisz funkcję BMI, która przyjmuje dwa parametry:
 - Wzrost – double
 - Waga – double
- Zwraca wartosc typu double – współczynnik BMI

Zadanie

- Program tablice matematyczne
 - Użytkownik podaje granice przedziału $\langle a, b \rangle$
 - int a
 - Int b
 - Oraz „krok” – czyli co ile ma się zwiększać w każdej iteracji
 - int krok
- Napisz program, który wypisze w wierszach kolejne liczby z przedziału $\langle a, b \rangle$ z krokiem – krok
- Przykład - $\langle 3, 8 \rangle$, krok – 2
 - 3
 - 5
 - 7

Zadanie

Stwórz funkcję, podzielnaPrzez

- Przyjmuje dwa parametry
 - int – liczba
 - int – dzielnik
- Zwraca true jeśli liczba jest podzielna bez reszty przez dzielnik
- Rozbuduj zeszły program, tak żeby dodatkowo pytał się o przedział wartość podzielników do sprawdzenia $\langle x, y \rangle$ i z – jako krok sprawdzenia
- Przykład
 - przedział liczb $\langle 3, 8 \rangle$, krok – 2
 - Do sprawdzenia $\langle 2, 4 \rangle$, z – 1
 - Liczba | 2 | 3 | 4 |
 - 3 | Nie | Tak | Nie |
 - 5 | Nie | Nie | Nie |
 - 7 | Nie | Nie | Nie |

Liczby doskonałe

- Liczba doskonała – liczba naturalna, która jest sumą wszystkich swych dzielników właściwych.
- Wejście:
 - Liczba naturalna
- Wyjście
 - Odpowiedź czy jest doskonała?
- Stwórz odpowiednią funkcję sprawdzającą!

Licznik

- Stwórz funkcję, która przyjmuje jako parametry
 - Tablicę double'i – tab
 - Liczbę typu double – liczba
- Wynikiem działania funkcji licznik, ma być liczba wystąpień „liczby” w tablicy
- Napisz także program, główny pozwalający na podanie zawartości tablicy (n liczb double) oraz liczby szukanej
- Na ekranie wyświetl podane liczby w tablicy, liczbę szukaną oraz liczbę wystąpień

Ćwiczenie

■ Wejście:

- Predefiniowana tablica String'ów 3x3 wypełniona wartościami:
 - X – gracz krzyżyk
 - O – gracz kółko
 - - - pole puste

■ Wyjście:

- Wypisz stan planszy
- Sprawdź czy któryś z graczy nie wygrał
 - Wygrana w pionie

Ćwiczenie

■ Wejście:

- Predefiniowana tablica String'ów 3x3 wypełniona wartościami:
 - X – gracz krzyżyk
 - O – gracz kółko
 - - - pole puste

■ Wyjście:

- Wypisz stan planszy
- Sprawdź czy któryś z graczy nie wygrał
 - Wygrana w pionie
 - **Wygrana w poziomie**

Ćwiczenie

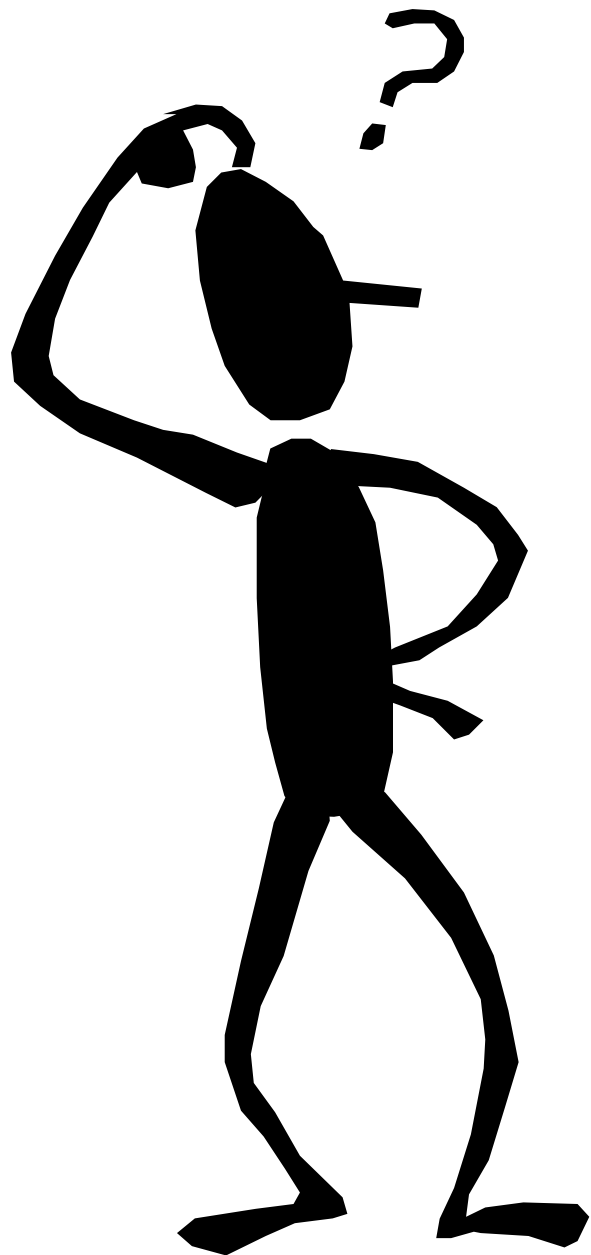
■ Wejście:

- Predefiniowana tablica String'ów 3x3 wypełniona wartościami:
 - X – gracz krzyżyk
 - O – gracz kółko
 - - - pole puste

■ Wyjście:

- Wypisz stan planszy
- Sprawdź czy któryś z graczy nie wygrał
 - Wygrana w pionie
 - Wygrana w poziomie
 - **Wygrana po przekątnej**

Koniec



Dziękuję za uwagę

