



Katedra
Informatyki i Automatyki
Politechniki Rzeszowskiej

Informatyka

Gry i zabawy

Opracował: Maciej Penar

Spis treści

Funkcje	3
Zadania	5
Mało realny problem: Paczkomat Kaczkomat	6
Zadania	6
Systemy liczbowe.....	7

Funkcje

Z Wikipedii [link](#):

Funkcja - dla danych dwóch zbiorów X i Y to przyporządkowanie każdemu elementowi zbioru X dokładnie jednego elementu Y . W matematyce funkcja nie ma stanu (pamięci). Oznacza się ją na ogół f, g, h itd.

Przykładem funkcji jest np. funkcja kwadratowa: $f(x) = x^2, x \in Re$

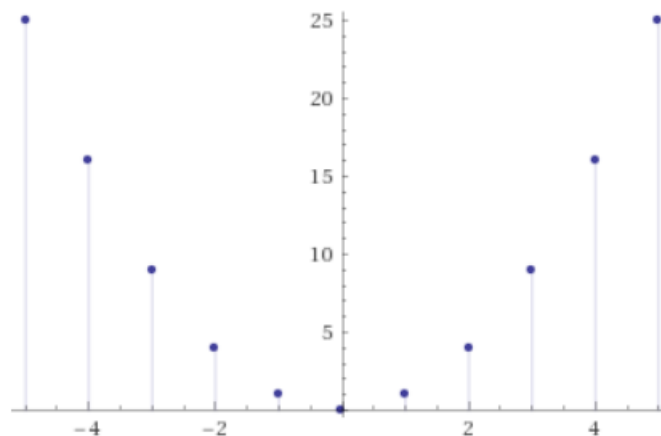
To wszystko o funkcja to prawda, ... ale nie w informatyce LOL. Więc kilka info:

1) w informatyce będzie zależeć nam na czytelności, dlatego niczym nadzwyczajnym jest nazywanie funkcji np.:

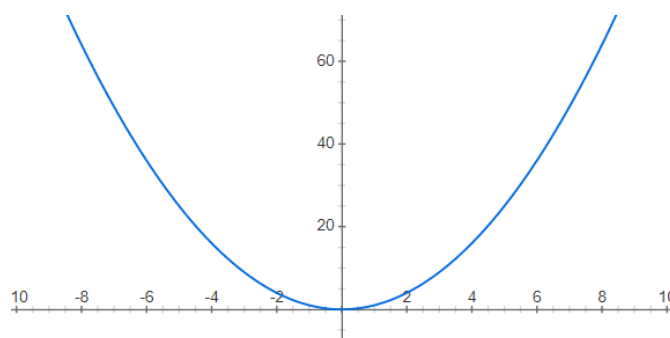
$KwadratLiczby(liczba) = liczba^2$ zamiast $f(x) = x^2$

2) w informatyce funkcje mogą operować na argumentach o **przeróżnym** typie (nie tylko Re, N, Z) i zwyczajowo typ podaje się po dwukropku `:` po nazwie argumentu np.:

$KwadratLiczby(liczba : Integer) = liczba^2$ – argument liczba jest Z



$KwadratLiczby(liczba : Double) = liczba^2$ – argument liczba jest Re



Jeśli to nie jest oczywiste (np. nie wynika z ciała funkcji) to czasem (w praktyce zawsze) podaje się typ zwracany funkcji np.:

KwadratLiczby(liczba : Double) : Integer – funkcja przyjmuje liczby rzeczywiste, a zwraca całkowite. Zauważmy że nie wiemy jakie dokładnie liczby całkowite zostaną zwrócone np. podłoga, sufit.

Przykłady realizacji (implementacji) tej funkcji:

KwadratLiczby(liczba : Double) : Integer = [liczba²]

KwadratLiczby(liczba : Double) : Integer = [liczba²]

3) te typy danych argumentów jak i wartości zwracanej mogą być przeróżne tutaj kilka przykładów funkcji:

- *Palindrom(tekst : String) : String*
 - *Palindrom('troll') = 'llort'*
- *DługośćTekstu(tekst : String) : Int*
 - *DługośćTekstu('studia') = 6*
- *IleZnaków(tekst : String, znak : Character) : Int*
 - *IleZnaków('troll', 'l') = 2*
 - *IleZnaków('troll', 'o') = 1*

4) funkcje mogą być skomplikowane i mogą wymagać **zmiennych pomocniczych** np.:

```
KwadratLiczby(liczba : Double) : Integer = {  
  x = liczba2  
  return [x]  
}
```

Słowem **return** podkreślamy że jest to tzw. **wartość zwracana** z funkcji (wynik).

5) **sygnaturą funkcji** nazywamy zapis na który składają się: nazwa funkcji, argumenty oraz jej typ zwracany (wynikowy) np.:

KwadratLiczby(liczba : Double) : Integer

Palindrom(tekst : String) : String

DługośćTekstu(tekst : String) : Int

Ciało funkcji to jej **definicja** – czyli to co jest po znaku = lub w nawiasach {}.

ZADANIA

1. Napisać sygnaturę funkcji o nazwie *Bezwzględna*, która zwraca wartość bezwzględną argumentu
2. Napisać sygnaturę funkcji o nazwie *Dodaj*, która zwraca sumę argumentów
3. Napisać ciało funkcji *Mnóż*, która zwraca mnożenie argumentów
4. Obliczyć:
 - a. *Palindrom('cześć')*
 - b. *Palindrom(Palindrom('cześć'))*
 - c. *DługośćTekstu(Palindrom('cześć'))*
 - d. *KwadratLiczby(DługośćTekstu('hau'))*
5. Wyjaśnij swoimi słowami dlaczego wyrażenie *DługośćTekstu(Palindrom('cześć'))* ma sens

Mało realny problem: Paczkomat Kaczkomat

Założmy, że mamy firmę która umożliwia klientom wymianę kaczek pomiędzy urządzeniami zwanymi kaczkomatami™. Programiści w naszej firmie szukają pomocy w opracowaniu operacji na kaczkomatach™ które są pod naszą administracją. Na razie nasz start-up jest mały i obsługuje ograniczoną liczbę kaczkomatów™:

- Kaczkomat numer 0 – na ulicy Dąbrowskiego w Rzeszowie
- Kaczkomat numer 1 – na ulicy Hetmańskiej w Rzeszowie
- Kaczkomat numer 2 – na ulicy Langiewicza w Rzeszowie
- Kaczkomat numer 3 – na ulicy Cieplickiego w Rzeszowie
- Kaczkomat numer 4 – na ulicy Krakowskiej w Rzeszowie
- Kaczkomat numer 5 – na ulicy Warszawskiej w Rzeszowie

Programiści wykombinowali że opiszą stan naszych kaczkomatów™ jako niekoniecznie „normalną” liczbę.

$$\text{Kaczkomat} := l_5 l_4 l_3 l_2 l_1 l_0, \text{ gdzie } l_i \in \langle 0, 9 \rangle$$

Gdy kaczkomat™ na ulicy Dąbrowskiego ma 2 kaczki, a kaczkomat™ na ulicy Krakowskiej 7 kaczek to stan kaczkomatu™ to: **070002**

Gdy kaczkomat™ na ulicy Hetmańskiej ma 1 kaczkę, na Langiewicza ma 3 kaczki, a kaczkomat™ na ulicy Warszawskiej 5 kaczek to stan kaczkomatu™ to: **500310**

ZADANIA

1. Rozważyć:
 - a. na czym polega „nienormalność” liczby reprezentującej kaczkomaty™
 - b. ograniczenia kaczkomatu™
2. Napisać funkcję *DodajPaczkę(firma : Kaczkomat, kaczkomat : Int)* która dodaje 1 paczkę wskazaną przez argument kaczkomat. Rozważyć dziedzinę tego argumentu. Na razie nie rozważać przypadków brzegowych. Przykłady:
 - a. *DodajPaczkę*(500310, 0) = 500311
 - b. *DodajPaczkę*(500310, 4) = 510310
3. Policzyc:
 - a. *DodajPaczkę*(110211, 5) = ?
 - b. *DodajPaczkę*(*DodajPaczkę*(110211, 5), 5) = ?
 - c. *DodajPaczkę*(000900, 2) = ?
4. Pomysł pyknął. Ludzie korzystają. Mamy nowy problem, klienci czasem proszą o przeniesienie kaczki z jednego kaczkomatu™ do drugiego kaczkomatu™. Napisać funkcję: *PrzenieśPaczkę(firma : Kaczkomat, kaczkomatZ : Int, kaczkomatDo : Int)*. Rozważyć dziedzinę kaczkomatZ i kaczkomatDo. Na razie nie rozważać przypadków brzegowych. Przykłady:
 - a. *PrzenieśPaczkę*(500310, 1, 0) = 500301
 - b. *PrzenieśPaczkę*(005500, 2, 3) = 006400
5. Kaczkomaty™ rosną. Jak można obejść problem ograniczonej liczby kaczek? Podać dwa sposoby. Czy kaczkomat™ może trzymać nieograniczoną (nieznaną) liczbę kaczek?

6. ! Napisać bezpieczne wersje funkcji: *DodajPaczkę()* i *PrzenieśPaczkę()*. Przykłady:
- DodajPaczkę*(900310, 5) = 900310
 - DodajPaczkę*(990310, 4) = 990310
 - PrzenieśPaczkę*(500310, 0, 1) = 500310, bo źródłowy jest pusty
 - PrzenieśPaczkę*(500319, 1, 0) = 500319, bo docelowy jest pełny

Systemy liczbowe

10 jednostek rzędu niższego tworzy jedną jednostkę rzędu wyższego.

Dla dowolnego systemu liczbowego o podstawie $p \geq 2$ możemy uogólnić powyższe stwierdzenie do;

p jednostek rzędu niższego tworzy jedną jednostkę rzędu wyższego.

Przykłady:

$p = 10$	$p = 3$	$p = 16$	$p = 2$
0	0	0	0
1	1	1	1
2	2	2	10
3	10	3	11
4	11	4	100
5	12	5	101
6	20	6	110
7	21	7	111
8	22	8	1000
9	100	9	1001
10	101	A	1010
11	102	B	1011
12	110	C	1100
13	111	D	1101
14	112	E	1110
15	120	F	1111
16	121	10	10000
17	122	11	10001

Mając liczbę zapisaną przy podstawie p wzór obliczający jej wartość w systemie 10 to:

$$x_n x_{n-1} \dots x_{0(p)} = x_n p^n + x_{n-1} p^{n-1} + \dots + x_0 p^0 = \sum_{i=0}^n x_i p^i \quad (10)$$

Np.:

$$112_{(3)} = 1 * 3^2 + 1 * 3^1 + 2 * 3^0 = 3^2 + 3^1 + 2 = 9 + 3 + 2 = 14_{(10)}$$

$$F1A_{(16)} = 16^2 * 15 + 16^1 * 1 + 16^0 * 10 = 256 * 15 + 16 + 10 = 3866_{(10)}$$

ZADANIA

Obliczyć wartość w $p = 10$:

2120_3	10010_2	562_8	$1011\ 1010_2$	$F0_{16}$
44_5	11110_3	$1A0_{16}$	83_9	1110_8
$A1_{11}$	123_7	$0111\ 1110_2$	22101_3	109_{11}
43_{12}	70_8	$1\ 0001\ 1111_2$	$7E_{16}$	511_{16}
$1000\ 1000_2$	420_8	EF_{16}	101_{16}	$K5_{25}$

Obliczyć wartość w $p = 2$:

$127_{(10)}$	$50_{(10)}$	$89_{(10)}$	$101_{(10)}$	$190_{(10)}$
--------------	-------------	-------------	--------------	--------------

Obliczyć wartość w $p = 4$:

$0111\ 1110_{(2)}$	$1111\ 0010_{(2)}$	$1010\ 1100_{(2)}$	$1110\ 0001_{(2)}$	$1010\ 1010_{(2)}$
--------------------	--------------------	--------------------	--------------------	--------------------

Obliczyć wartość w $p = 8$:

$0111\ 1110_{(2)}$	$1111\ 0010_{(2)}$	$1010\ 1100_{(2)}$	$1110\ 0001_{(2)}$	$1010\ 1010_{(2)}$
--------------------	--------------------	--------------------	--------------------	--------------------

Obliczyć wartość w $p = 16$:

$0111\ 1110_{(2)}$	$1111\ 0010_{(2)}$	$1010\ 1100_{(2)}$	$1110\ 0001_{(2)}$	$1010\ 1010_{(2)}$
--------------------	--------------------	--------------------	--------------------	--------------------