



Katedra  
Informatyki i Automatyki  
Politechniki Rzeszowskiej

# Informatyka

## Schematy Blokowe i inne zabawy

Opracował: Maciej Penar

## Spis treści

Funkcje przedostatni raz .....	3
Typ funkcyjny .....	3
Argument innej funkcji .....	3
typ zwracany innej funkcji .....	3
Uproszczona arytmetyka modularna .....	4
Zadania .....	4
Schematy Blokowe .....	5
Zadania .....	5

## Funkcje przedostatni raz

### TYP FUNKCYJNY

Funkcja może być traktowana jako specjalny rodzaj danych. Tak samo jak mówiliśmy o *Int*, *Double* czy *String*, tak samo możemy mówić o typie: *Function(...)*. Taki typ możemy podać jako argument innej funkcji albo jak typ zwracany (funkcja która zwraca inne funkcje).

### ARGUMENT INNEJ FUNKCJI

Przykładem pierwszego przypadku (tj. argument innej funkcji) może być filtrowanie tablicy:

$$\text{filter}(\text{arr}: \text{Array} < \text{Int} >, \text{predicate}: \text{Function} < \text{Int} >: \text{Bool})$$

Mając tablicę:  $x = [10, -5, 12, 4, 5]$

Wywołania mają następujący efekt:

- $\text{filter}(x, (\text{arg}: \text{Int})\{ \text{return } \text{arg} \geq 10 \}) = [10, 12]$
- $\text{filter}(x, (\text{arg}: \text{Int})\{ \text{return } \text{arg} < 0 \}) = [-5]$

Zwróćmy uwagę że:

- podczas wywołania funkcji *filter*, funkcja podana w parametrze *predicate* nie musi mieć podanej nazwy – jest ona istotna tylko podczas definiowania funkcji *filter*
- podczas wywoływania funkcji *filter*, funkcja podana w parametrze *predicate* musi mieć nazwane argumenty – co nie jest wymagane przy definiowaniu funkcji *filter*

### TYP ZWRACANY INNEJ FUNKCJI

Drugi przypadek ma miejsce gdy chcemy sterować implementacją funkcji (wybrać odpowiednie dla nas zachowanie). Przykładem może być szyfrowanie. Dane są funkcje wykonujące szyfrowanie dwoma algorytmami:

$$\text{cesar}(\text{text}: \text{String}, \text{key}: \text{Int}) : \text{String}$$
$$\text{xor}(\text{text}: \text{String}, \text{key}: \text{Int}) : \text{String}$$

Chcemy wybrać funkcje na podstawie ich nazwy:

$$\text{getCipher}(\text{name}: \text{String}) : \text{Function}(\text{String}, \text{String}) : \text{String}$$

W ten sposób możliwe wywołania to:

- $\text{getCipher}(\text{'cesar'}) (\text{'troll'}, 13) = \text{gebyy}$
- $\text{getCipher}(\text{'xor'}) (\text{'ABC'}, 64) = 00\ 03\ 02_{(16)}$

## Uproszczona arytmetyka modularna

Operacja % czyli wyznaczania reszty z dzielenia. Użycie funkcji/operatora dzielenia modulo gwarantuje że wartości będą całkowite i z zakresu  $[0, mod)$ . Jeśli  $f(x) \in R$ , to:

- $f(x) \% 10 \in [0, 10) \cap \mathbb{Z}$
- $f(x) \% 24 \in [0, 23) \cap \mathbb{Z}$
- $(f(x) \% 6) + 10 \in [10, 16) \cap \mathbb{Z}$

W informatyce operacja ta znajduje zastosowanie jako zamiennik na max/min/if oraz przy generowaniu liczb losowych. Tak naprawdę w informatyce każdy prymitywny typ danych (np. *Int*, *Double*, *Char*) możemy traktować jako podlegający arytmetyce modularnej – ze względu na przepełnienie wartości (np. Integer Overflow).

### ZADANIA

Policzyć:

1. Jest dzień 27 listopad 12:00. Dodać 15 godzin wprzód.
2. Jest dzień 27 listopad 11:00. Dodać 4 dni i 6 godzin wprzód.
3. Jest dzień 27 listopad 11:00. Dodać 7 dni i 10 godzin w tył.

Mamy grupę:  $\{A, B, C, \dots, Z\}$ . Załóżmy, że  $A + 1 = B, B + 1 = C, \dots, Z + 1 = A$

4. Wyznaczyć moduł

Policzyć:

5.  $A + 10$
6.  $T + 7$
7.  $Z + 20$
8.  $M + 26$
9.  $K - 25$
10.  $K + 27$

## Schematy Blokowe

### ZADANIA

Narysuj schemat blokowe następujących problemów:

1. Wczytaj liczbę do zmiennej k. Na wyjściu wypisz liczbę przeciwną.
2. Wczytaj liczbę do zmiennej k. Na wyjściu wypisz czy jest parzysta
3. Wczytaj liczby do zmiennej k dopóki użytkownik nie wprowadzi -1. Na wyjściu wypisz:
  - a. Sumę
  - b. Ile elementów użytkownik wprowadził
  - c. Średnią
4. Wczytaj liczbę do zmiennej k. Na wyjściu wypisz z ilu cyfr się składa.
5. Wczytaj liczby do zmiennych k, m. Na wyjściu wypisz resztę z dzielenia k przez m. Komentarz: założmy że nie mamy operacji modulo % (link: <https://pl.wikipedia.org/wiki/Modulo>)
6. Wczytać tablicę liczb do zmiennej t. Znaleźć oraz wypisać na wyjściu: a. Minimalną wartość b. Maksymalność wartość
7. Założmy że użytkownik wczytuje ciąg znaków do zmiennej s o długości n oraz mamy zdefiniowaną operację s[i] zwracającą i-ty znak. Np. dla s="Informatyka" operacja s[0] -> „I”, s[1] -> „n”, s[2] -> „f”, itd. Wczytać ciąg znaków oraz wypisać na wyjściu czy dany ciąg jest palindromem.
8. Dla modelu z zadania 7. Wczytać ciąg znaków s oraz wypisać cały wyraz wspak.
9. Dla modelu z zadania 7. Wczytać ciągi znaków s1, s2. Określić najdłuższy wspólny prefix tych wyrazów i wypisać na wyjściu (liczbę, nie prefix)