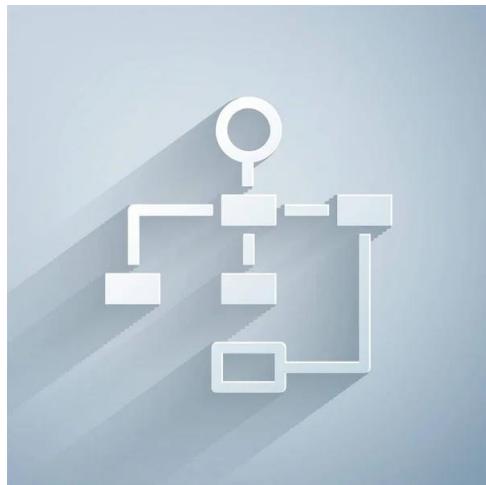


ALGORYTMY I STRUKTURY DANYCH



WARUNKI ZALICZENIA PRZEDMIOTU I KRYTERIUM WYSTAWIENIA OCENY KOŃCOWEJ

Na przedmiocie zdobywamy punkty, które na koniec semestru będą przeliczone na ocenę końcową.

Punkty można zdobywać następującymi sposobami:

➤ Aktywność na zajęciach

Na zajęciach będziemy rozwiązywali problemy algorytmiczne zamieszczone na kolejnych listach z zadaniami. Na zajęciach można się zgłaszać do tablicy, rozwiązywać zadania i przekonując wyjaśniać sposób działania algorytmów. Za poprawnie rozwiązane jedno zadanie, w zależności od stopnia jego trudności, można uzyskać od 3 do 6 punktów. Za bardziej dogłębne wyjaśnienie zadania i/lub przedstawienie więcej niż jednego wariantu jego rozwiązania i/lub omówienie jego złożoności obliczeniowej, ilość punktów może być zwiększoana (np. z 3 do 4 lub 5). Punkty można także uzyskać wskazując błędy lub istotne poprawki w zadaniu rozwiązranym na tablicy.

➤ Kolokwia

W semestrze odbędą się dwa kolokwia zaliczeniowe, obejmujące głównie zadania przerabiane na zajęciach. Pierwsze kolokwium odbędzie się na zajęciach około połowy semestru, drugie na ostatnich zajęciach. Na kolokwiach tych można będzie zdobyć 50 punktów.

➤ Ekstra problemy

Oprócz zadań znajdujących się na listach zadań i rozwiązywanych na zajęciach, rozwiązać będzie można zadania dodatkowe, pisząc programy w języku C/C++. W trzecim miesiącu semestru zostaną podane dwa takie problemy. Na ich rozwiązanie będzie kilka dni, a za poprawne rozwiązanie każdego z tych problemów algorytmicznych będzie można uzyskać, w zależności od stopnia jego trudności, od 10 do 20 punktów.

➤ Nieobecności

Bez żadnych konsekwencji każdemu przysługują trzy nieobecności. Warunkiem uzyskania zaliczenia jest także nieprzekroczenie pięciu nieobecności. Za każdą nieobecność powyżej trzech i poniżej sześciu, odejmowane będzie 6 punktów, czyli przy maksymalnie pięciu nieobecnościami, będzie to odjęcie 12 punktów.

Za 100% frekwencję na zajęciach przysługuje dodatkowe 6 punktów.

Skala ocen:

- od 28 pkt: dostateczny
- od 35 pkt: dostateczny plus
- od 42 pkt: dobry
- od 49 pkt: dobry plus
- od 56 pkt: bardzo dobry

Mariusz Sobol

DEFINICJA PSEUDOKODU JAKI BĘDZIEMY STOSOWAĆ NA ĆWICZENIACH



- Przypisanie do zmiennej wyliczonej wartości:

```
zmienna ← wartość
```

- Operatory arytmetyczne zapisujemy są za pomocą powszechnie używanych w matematyce symboli, czyli:

$$+, -, \cdot, /, \neq, \geq, \leq, \frac{2a}{b-c}, \sqrt{a \cdot b}, \text{ itd.}$$

część całkowitą i resztę z dzielenia oznaczamy jako: div i mod

- Instrukcja warunkowa:

```
if warunek then  
    ciąg instrukcji
```

lub

```
if warunek then  
    ciąg instrukcji-1  
else  
    ciąg instrukcji-2
```

- Pętla while:

```
while warunek do  
    ciąg instrukcji
```

- Pętla for:

```
for zmienna ← wart_pocz to wart_końc do  
    ciąg instrukcji
```

lub

```
for zmienna ← wart_pocz to wart_końc step krok do  
    ciąg instrukcji
```

w pierwszym przypadku zmienna zmienia się w każdym kroku o 1, w drugim przypadku o wartość podaną jako krok

- Warunek logiczny ograniczamy nawiasami okrągłymi **()**, a operatory logiczne alternatywy, koniunkcji i negacji oznaczamy słowami kluczowymi: **or**, **and** i **not**
- Blok instrukcji wyróżniamy nawiasami klamrowymi **{ }**
- Dostęp do kolejnych elementów tablicy uzyskuje się przez podanie nazwy tablicy i indeksu:
dla tablicy jednowymiarowej **A[i]**, dla tablicy dwuwymiarowej **A[i, j]**
Pierwszy element tablicy posiada indeks **i=1**.
- Operacje wprowadzania danych wejściowych i wyprowadzania wyników (operacje we/wy) zapisujemy jako:
read(n) oraz **write(x, y)** a także
write("to nie jest liczba parzysta")
write("silnia = ", s)
- Zapis algorytmu poprzedzamy podaniem nazwy algorytmu