

Sterowanie Procesami Dyskretnymi

Laboratorium 0

Zasady zaliczenia

prowadzący: *mgr inż. Radosław Idzikowski*

1 Zasady zaliczenia

- Zajęcia prowadzone w czasie rzeczywistym przy użyciu platformy MS Teams.
- Obecność podczas zajęć jest obowiązkowa. Dopuszczalne są nieobecności bez podania przyczyny oraz 2 uzasadnione nieobecności (z okazaniem odpowiedniego dokumentu np. zwolnienia lekarskiego). Nadmiarowe nieobecności skutkują zmniejszeniem oceny z kursu.
- Kurs obejmuje 5 tematów (na każdy przeznaczone są 2-3 spotkania), w ramach których będą do wykonania zadania.
- Tematy realizowane są w zespołach dwuosobowych.
- Z każdego tematu wystawiana jest ocena cząstkowa (łącznie 5). Przez większość czasu obowiązuje wspólna ocena dla obu członków zespołu, ale w szczególnych przypadkach ich oceny mogą się różnić.
- Ocena końcowa z laboratorium jest średnią arytmetyczną ze ocen cząstkowych. Aby uzyskać zalecenie wszystkie oceny cząstkowe muszą być pozytywne (≥ 3.0).
- Sprawdzanie zadań odbywa się pod koniec (ostatnie 45 minut) ostatnich zajęć z danego tematu.
- Spóźnione zadania można oddawać w trakcie kolejnych zajęć lub na konsultacjach. Pierwszy rozpoczęty tydzień spóźnienia zmniejsza ocenę z tematu o -0.5 , zaś drugi o kolejne -1.0 (łącznie -1.5). Trzeci rozpoczęty tydzień spóźnienie oznacza niezaliczenie kursu.
- Oddanie zadania polega na prezentacji zrealizowanych zadań (kod, model, działanie) oraz rozmowie z prowadzącym.
- Po uzyskaniu pozytywnej oceny z danego tematu należy przesłać zrealizowane zadanie (kod lub model) jako pliki z odpowiednim rozszerzeniem (.cpp, .cs, .py, . itd.). W przypadku więcej niż 2-3 plików można przesłać archiwum ZIP. W szczególności:
 1. **nie** wysyłać plików wykonywalnych lub plików projektowych.
 2. **nie** wysyłać plików bez rozszerzenia.

Dozwolone, a nawet zalecane jest przesłanie linku do repozytorium `github`.

- Maile należy przysyłać z odpowiednim nagłówkiem postaci `[spd][indeks][NR]`, gdzie „indeks” jest 6-cyfrowym numerem indeksu, zaś „NR” to numer tematu (nie zajęć!) od 1 do 5. Mail wysyłają obie osoby z grupy.

2 Harmonogram pracy

Nr zajęć	NR	Tematyka
1	-	Wprowadzenie. Zasady zaliczenia. Konfiguracja wybranego środowiska. Implementacja algorytmu sortR dla problemu $1 r_i C_{\max}$.
2-4	1	Opracowanie i implementacja algorytmów dla problemu $1 r_i, q_i C_{\max}$: <ul style="list-style-type: none">• algorytm Schrage,• algorytm Carlier.
5-6	2	Opracowanie i implementacja algorytmów dla problemu $1 \sum w_i T_i$: <ul style="list-style-type: none">• przegląd zupełny,• programowanie dynamiczne.
7-8	3	Opracowanie i implementacja algorytmów dla problemu $FP C_{\max}$: <ul style="list-style-type: none">• przegląd zupełny• algorytm Johnson,• metoda podziału i ograniczeń.
9-11	4	Opracowanie i implementacja algorytmów dla problemu $FP C_{\max}$ lub $J C_{\max}$: <ul style="list-style-type: none">• algorytm konstrukcyjny NEH ($FP C_{\max}$) lub INSA ($J C_{\max}$),• wybrany algorytm przeszukiwania lokalnego RS, SA, TS.
12-14	5	Opracowanie modeli oraz znalezienie rozwiązania dokładnego przy pomocy pakietu optymalizacyjnego lub inny.
15	-	Wystawienie oceny. Zajęcia odróbkowe.

3 Środowiska pracy

- Zajęcia 1–11 odbywać się będą z wykorzystaniem języka `python`, `C/C++`, `Java` lub `C#`. Inne języki na własną odpowiedzialność.
- Zajęcia 12–14 odbywać się będą z wykorzystaniem IBM ILOG CPLEX Optimization Studio lub z wykorzystaniem języka `python`, ewentualnie (na własną odpowiedzialność) `C/C++`, `Java` oraz `C#`.
- Ponadto do generowania danych do instancji testowych na zajęciach 1–14 posłużą generatory liczb pseudolosowych. Wersje w językach `python`, `C/C++`, `Java` oraz `C#` zostaną dostarczone przez prowadzącego.

opracował: *Radostaw Idzikowski*