

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА
Факультет прикладної математики та інформатики

**Паралельні та розподілені обчислення
ЛАБОРАТОРНА РОБОТА №4**

Тема: «Форми».

Виконала:
Ст. *Пелєщак Вероніка*
ПМІ-35с

Завдання: Створити програму, що моделює роботу трьох потоків. Кожен потік працює певний час t_1 , t_2 , t_3 , після чого передає наступному потоку новий часовий інтервал для роботи. Під час простою потік очікує на дані від іншого потоку. Робота потоків відображається у вигляді трьох циліндрів (прямокутників), де:

- синім кольором позначено час роботи потоку;
- червоним кольором — час простою.

Передбачено можливість введення початкових інтервалів, випадкову генерацію та покрокову анімацію виконання.

Хід роботи:

1. `__init__(self, root)`

Ініціалізує графічний інтерфейс: створює вікно, полотно (Canvas) для малювання, панель керування з кнопками, полями введення та чекбоксом “Animate”.

2. `clear_canvas(self)`

Очищає екран і малює три порожніх “цилінди” (по 20 клітинок у кожному). Використовується для початкового стану або при повторному запуску.

3. `randomize_initial(self)`

Генерує випадкові значення початкових часових інтервалів t_1, t_2, t_3 у заданих межах і записує їх у відповідні поля введення.

4. `start_simulation(self)`

Зчитує введені або згенеровані інтервали, запускає симуляцію (метод `run_discrete_simulation`) і відображає результат — одразу (`draw_states`) або покроково (`animate_draw`).

5. **run_discrete_simulation(self, initials, max_random)**

Основна логічна частина. Моделює роботу трьох потоків по кроках:

1. потік “працює”, якщо має залишок часу;
2. по завершенню роботи передає новий інтервал наступному потоку;
3. якщо потік чекає — він “простоює”.

Повертає матрицю станів (робота / простій) для кожного потоку.

6. **draw_states(self, states)**

Відображає результати симуляції без анімації, одразу зафарбовуючи клітинки синім або червоним кольором.

7. **animate_draw(self, states, step=0)**

Виконує покрокове заповнення “циліндрів”, поступово додаючи рядки з певною затримкою (self.root.after(150, ...)) — створює ефект анімації.

Ключові моменти програми:

- Використано бібліотеку **tkinter** для побудови графічного інтерфейсу.
- Потоки представлені у вигляді вертикальних стовпчиків (по 20 рівнів).
- Моделювання відбувається дискретно у часі — кожен рядок відповідає одному часовому кроку.
- Передача часу між потоками відбувається по колу:
Thread 1 → Thread 2 → Thread 3 → Thread 1.
- Простій (червоний) з’являється тоді, коли потік чекає, поки попередній завершить свою роботу.
- Є можливість рандомізації початкових параметрів і анімації процесу.

Результаты:



