

**Національний Технічний Університет України КПІ**

Факультет Інформатики та Обчислювальної Техніки  
Кафедра інформатики та програмної інженерії

**Практична робота №3**

З дисципліни «Моделювання систем»

ПОБУДОВА ІМІТАЦІЙНОЇ МОДЕЛІ СИСТЕМИ З ВИКОРИСТАННЯМ

ФОРМАЛІЗМУ МОДЕЛІ МАСОВОГО ОБСЛУГОВУВАННЯ

**Перевірив:**

Асистент

Дифучин А.Ю.

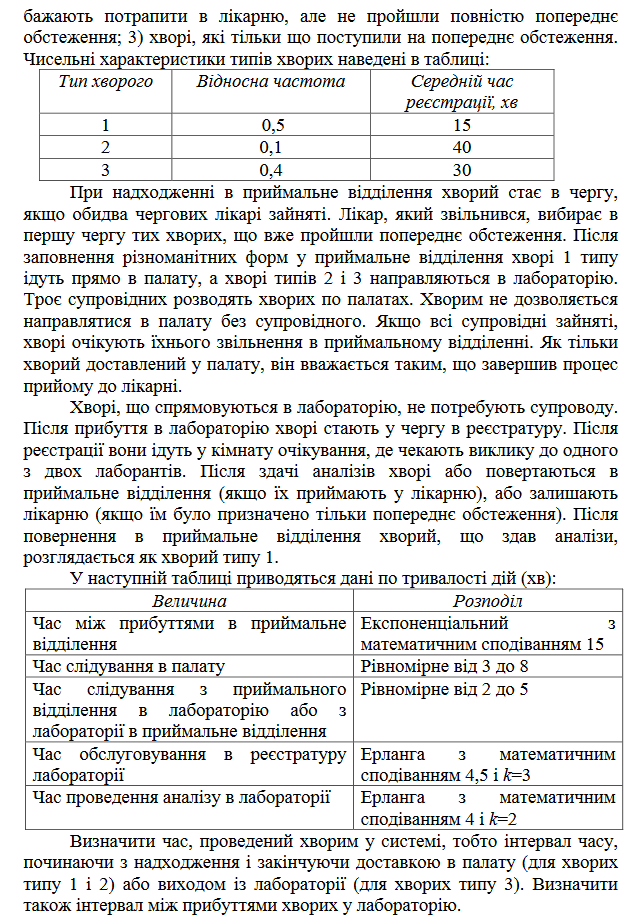
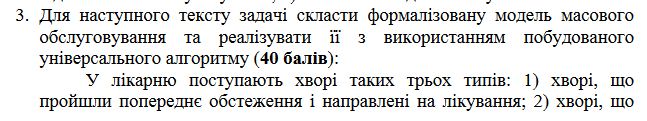
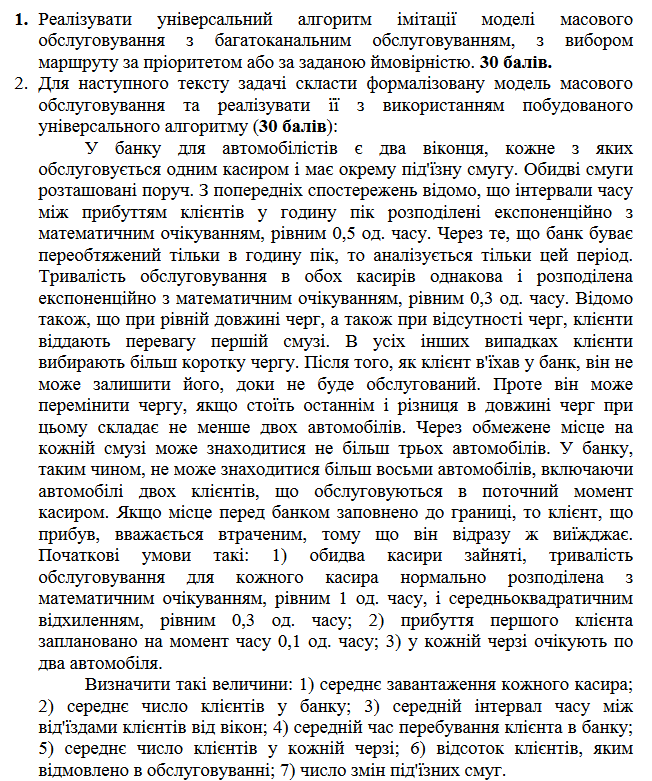
Оцінка:\_\_\_\_\_\_\_\_\_\_\_**Виконав:**

Студент групи ІТ-04

Панов В. Є.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_

**Завдання до практичної роботи**

****

**Виконання**

* Завдання 1

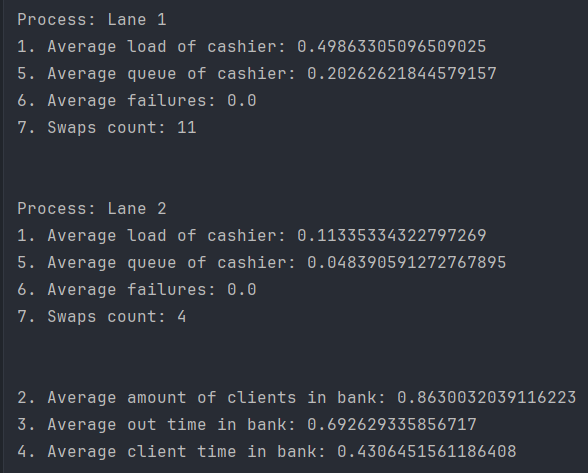
Задачу було реалізовано, продемонстровано та описано у лаб. роботі 2.

* Завдання 2

Виконані модифікації:

Елемент Create направляє процес до процесу, який має найкоротшу чергу. Під час створення нового об'єкту tnext відбувається перегляд пріоритетної черги, де зберігаються всі елементи, які можуть бути наступними, і обирається той, який має найменше значення.

Модель заповнюється автомобілями з відхиленням від типового розподілу затримок. Це досягається через виклики функції inAct() об'єктів класу Process із встановленими потрібними законами розподілу перед початком симуляції.

****

* Завдання 3

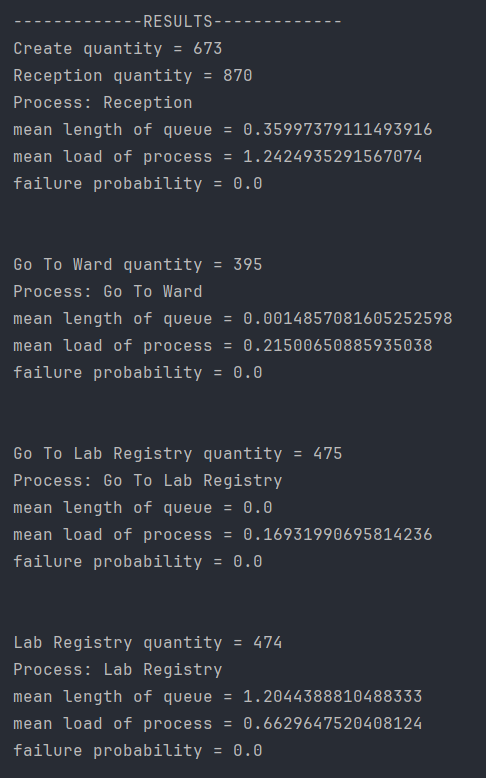
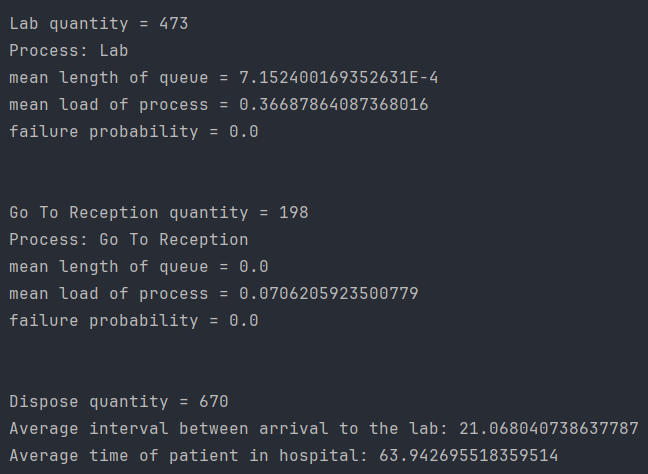
Виконані модифікації:

Замість створення однакових пацієнтів, у функції Create створюються три типи пацієнтів, які направляються кожен у свій процес. Спершу в функції Create створюється об'єкт Patient. Цей об'єкт передається в спеціальний зв'язаний список (який може здатися зайвим на перший погляд), а потім передається у наступний процес. Такий же механізм передачі об'єкта Patient використовується і в процесі.

Час обслуговування пацієнтів у конкретному процесі («Reception») не вибирається випадковим чином, а фіксується в залежності від типу Patient.

Якщо пацієнт є правильного типу та перебуває в конкретному процесі («Lab»), то замість того, щоб йти до Dispose, він змінює свій тип та направляється до іншого процесу.

В класі Patient фіксується час прибуття до лікарні та час, який він там провів.

**Висновок**

Був створений універсальний алгоритм імітації моделі масового обслуговування, який був використаний для вирішення наступних завдань. Для другого завдання була сформальована модель масового обслуговування та реалізована за допомогою універсального алгоритму імітації. Для третього завдання також були внесені модифікації до класів системи масового обслуговування для врахування особливостей, викладених у вимогах завдання.