

Тестовое задание

на позицию Java-разработчик
(Специалист по имитационному
моделированию)

Введение

Необходимо разработать инструмент для решения задачи распределения сотрудников между производственными центрами.

Инструмент должен быть разработан на языке Java без использования сторонних библиотек (допускается Apache POI для работы с excel документами) или в системе имитационного моделирования AnyLogic.

Конечное решение может представлять из себя как десктопное, так и консольное приложение.

Написать несколько [Unit-тестов](#) к разработанному инструменту.

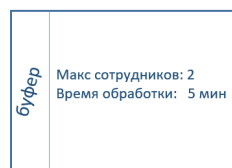
Описание задачи и предметной области

- 1) Существует цех, в котором присутствуют различные производственные центры (далее ПЦ), которые обрабатывают детали. Все детали одинаковые, дискретные и неделимые. На каждом производственном центре могут работать один или несколько сотрудников, занимающиеся обработкой деталей. Производственные центры связаны друг с другом, и деталь после обработки попадает на следующий ПЦ.

Все сотрудники могут выполнять задачи на всех ПЦ и перемещаться между ними (перемещение происходит мгновенно) в любое время, если в этот момент он не занят обработкой детали. Общее количество сотрудников ограничено. [Перед перемещением сотрудник должен завершить обработку детали, если она начата, и только потом переместиться в другой центр.](#)

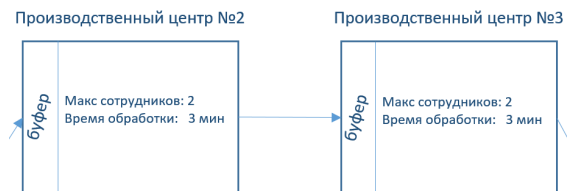
- 2) ПЦ представляет собой объект, на котором один или несколько сотрудников обрабатывают детали. Несколько сотрудников могут работать параллельно в одном производственном центре. Один сотрудник может обрабатывать только одну деталь одновременно. Т.е. приведенный ниже центр может обрабатывать 2 детали в 5 минут, если на нем работают два человека и только 1 в 5 минут, если сотрудник один. [Время обработки одной детали – рациональное число больше нуля](#) (может быть и 2, и 0.5, и 11.123 минут). Для каждого ПЦ задается максимальное количество одновременно работающих сотрудников.

Производственный центр №1



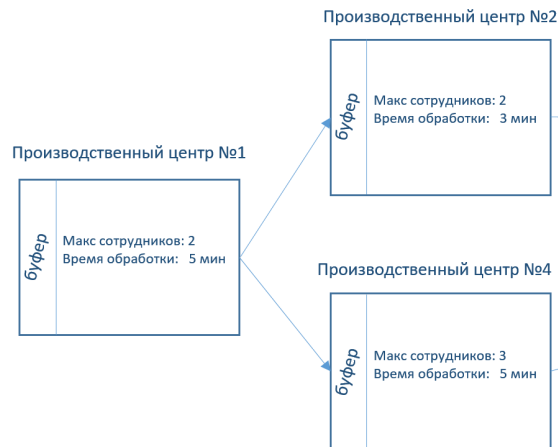
У каждого ПЦ есть буфер, в который попадают детали, ожидающие обработки. Размер буфера неограничен. Детали, ожидающие обработки, скапливаются в буфере. Если в конкретный момент времени нет сотрудников в производственном центре, то обрабатываться детали не могут.

- 3) Связи между ПЦ отображают процесс перехода детали между ПЦ, на котором она прошла обработку, и буфером следующего по процессу ПЦ.



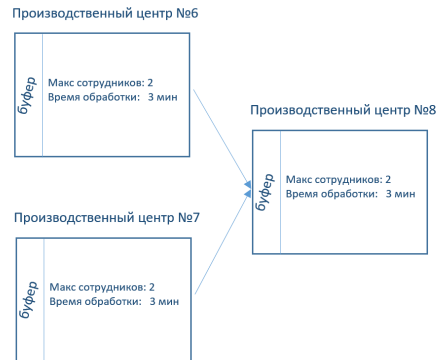
Детали после завершения обработки деталь попадет в буфер следующего центра без задержки, мгновенно и становится доступной для следующей обработки.

Существует возможность задать несколько связей между производственными центрами:



В таком случае детали распределяются равномерно и последовательно по одному на каждую исходящую связь, т.е. первая обработанная деталь попадет в буфер центра №2, вторая в буфер центра №4, третья в центр №2 и так далее.

Возможна и обратная ситуация, когда в буфер ПЦ поступают детали от нескольких предшествующих центров.

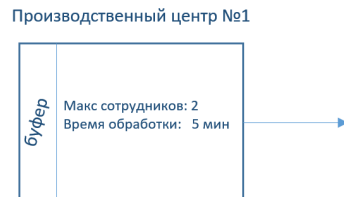


Таких связей, как приведенные выше, может быть неограниченное количество, не только по две.

Циклические цепочки связей не должны допускаться.

4) Начальный производственный центр

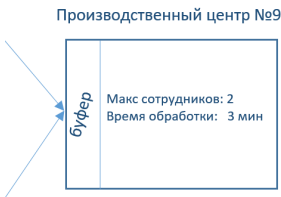
Центр без единой выходящей связи является начальным. Он может быть только в единственном экземпляре.



В буфере этого центра изначально находятся детали, помещённые туда при инициализации, которые в последующем будут проходить обработку на всех остальных центрах.

5) Конечный производственный центр

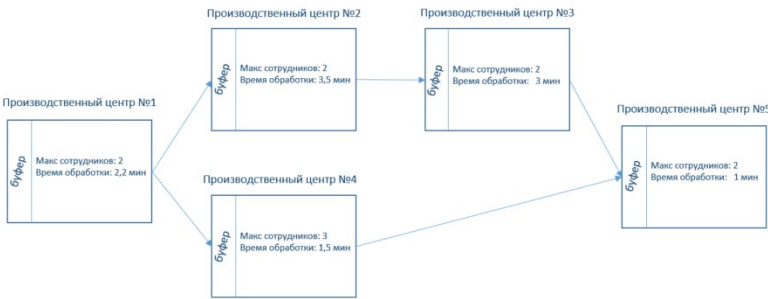
Центр без единой исходящей связи является конечным. Он может быть только в единственном экземпляре.



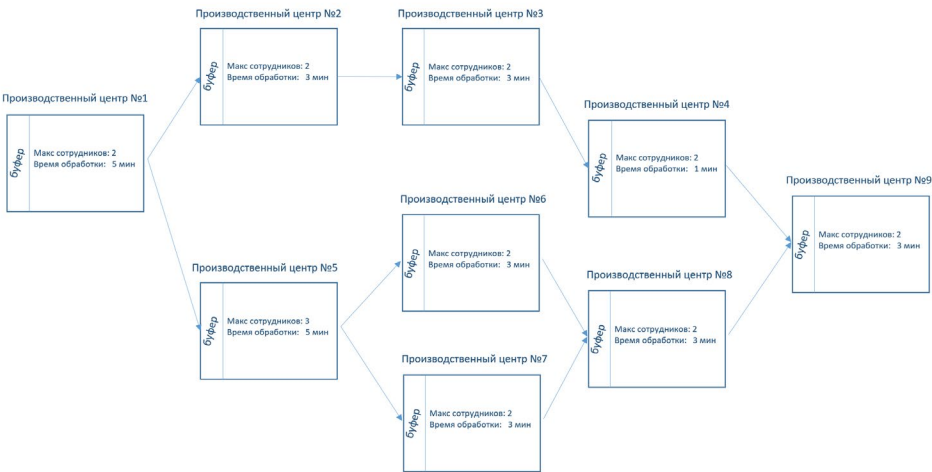
После обработки на этом центре деталь считается до конца обработанной и больше никуда не поступает.

Так могут выглядеть диаграммы различных цехов:

Пример 1



Пример 2



Задача состоит в том, чтобы в зависимости от нагрузки на различные центры перераспределять сотрудников между ними так, чтобы обработать все детали максимально быстро, с учетом заданного общего количества сотрудников.

Разработанный инструмент должен уметь решать эту задачу для любого количества сотрудников, любого количества производственных центров, связей между ними и произвольными параметрами на каждом центре – максимальное количество сотрудников и время обработки одной детали.

Ожидаемая размерность данных:

- 1) Количество сотрудников - до 40
- 2) Количество ПЦ – до 20
- 3) Макс. время на обработку одной детали – 10 мин
- 4) Количество деталей - до 2000

Формат входных данных

Сценарий, содержащий в себе входные данные передается в инструмент в формате Excel с тремя листами следующего наполнения:

A	B	C	D
workersCount	detailsCount		
6	500		

A	B	C	D
id	name	performance	maxWorkersCount
1	Производственный центр №1	2,2	2
2	Производственный центр №2	3,5	2
3	Производственный центр №3	3	2
4	Производственный центр №4	1,5	3
5	Производственный центр №5	1	2

A	B
sourceCenter	destCenter
Производственный центр №1	Производственный центр №2
Производственный центр №1	Производственный центр №4
Производственный центр №2	Производственный центр №3
Производственный центр №3	Производственный центр №5
Производственный центр №4	Производственный центр №5

На скриншотах выше изображен сценарий для цеха из [примера 1](#)

Формат выходных данных

После обработки всех деталей разработанное решение должно записать в CSV файл данные следующего содержания:

Для каждой минуты и каждого ПЦ вывести количество работающих сотрудников и количество деталей в буфере.

Заголовки должны быть следующие - Time, ProductionCenter, WorkersCount, BufferCount

В качестве разделителя выступает «, ».

Пример:

Time, ProductionCenter, WorkersCount, BufferCount

0.0, Производственный центр №1, 2, 498

0.0, Производственный центр №2, 2, 0

0.0, Производственный центр №3, 0, 0

0.0, Производственный центр №4, 2, 0

0.0, Производственный центр №5, 0, 0

1.0, Производственный центр №1, 2, 498

1.0, Производственный центр №2, 2, 0

1.0, Производственный центр №3, 0, 0

1.0, Производственный центр №4, 2, 0

1.0, Производственный центр №5, 0, 0

2.0, Производственный центр №1, 2, 498

2.0, Производственный центр №2, 2, 0

2.0, Производственный центр №3, 0, 0

2.0, Производственный центр №4, 2, 0

2.0, Производственный центр №5, 0, 0

3.0, Производственный центр №1, 3, 495

.

.

.

622.0, Производственный центр №5, 1, 1

623.0, Производственный центр №1, 0, 0

623.0, Производственный центр №2, 2, 0

623.0, Производственный центр №3, 3, 0

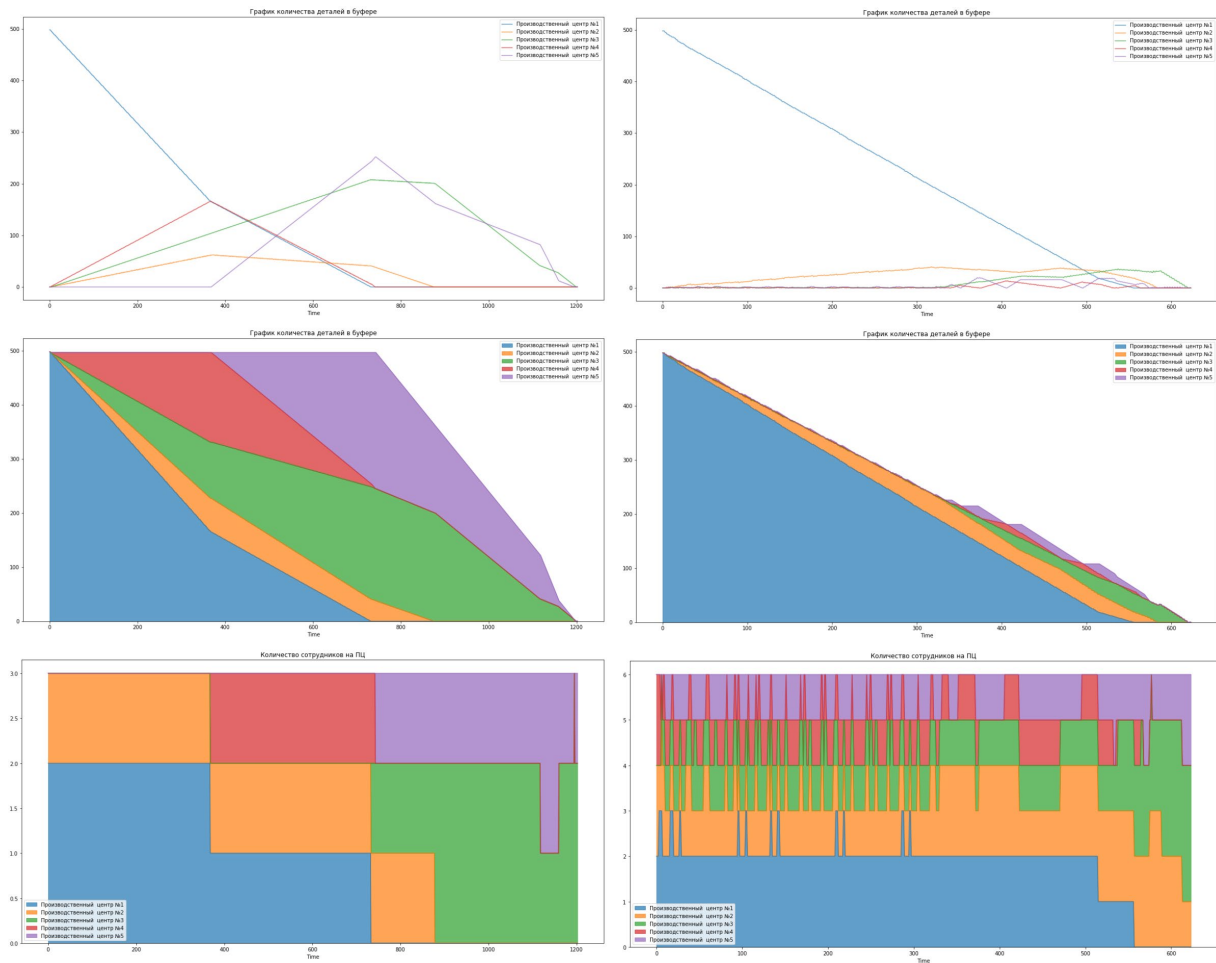
623.0, Производственный центр №4, 0, 0

623.0, Производственный центр №5, 1, 0

Визуализация и статистика

К тестовому заданию мы написали небольшой скрипт в JupyterNotebook, который может помочь вам визуализировать результаты из CSV файла, в случае, если вы будете делать консольное приложение – в нем надо прописать путь к CSV файлу и запустить, после чего он построит графики.

Графики, построенные по результатам для цеха из [примера 1](#) для трёх и шести сотрудников соответственно.



Приложенные к заданию результаты для тестовых сценарии могут не являться эталонными и оптимальными. Они больше нужны для демонстрации.

Отправка решения

Исходный код вашего решения необходимо загрузить в публичный репозиторий, например, GitHub. В README написать инструкцию, как ваше решения запускать и работать с ним.