

## Менеджер-бот и round-robin

---

Популярный алгоритм распределения нагрузки между исполнителями round-robin подразумевает наличие  $N$  исполнителей и  $M$  задач, обычно  $N$  много меньше чем  $M$ . Алгоритм заключается в простом распределении задач по исполнителям в цикле. Первому исполнителю назначается первая задача, второму -- вторая и т. д.  $N+1$  - я задача снова назначается первому исполнителю, и так далее по кругу.

Каждая задача характеризуется своей сложностью в условных единицах работы (например, 42).

Каждый исполнитель характеризуется производительностью -- сколько единиц работы в единицу времени он выполняет (например, 5). Каждому исполнителю может быть назначено несколько задач, он последовательно выполняет первую в списке задач.

Реализуем алгоритм round-robin в виде менеджер-бота -- настольной программы с графическим пользовательским интерфейсом.

Кнопка New начинает новый сеанс моделирования: случайно генерируются исполнители (каждый со своей производительностью) и задачи (каждая со своей сложностью), стартует таймер. Диапазон различных характеристик берётся из настроек программы (см. ниже).

Сделайте метод A, который в начале работы (по нажатию на New) распределяет список существующих задач между исполнителями по принципу round-robin.

Сделайте метод В, который перераспределяет задачи: все первые задачи каждого исполнителя переходят к следующему исполнителю (помещаются в начало списка задач). Задача последнего исполнителя переходит к первому исполнителю. Остальные задачи не трогаются.

Когда срабатывает таймер, каждый исполнитель вычитает из сложности первой задачи в списке свою производительность. Если сложность становится меньше или равной нулю, то задача удаляется из списка. Затем с вероятностью 50% срабатывает метод В.

Кнопка Pause останавливает моделирование -- ставит таймер на паузу.

Придумайте способ генерации случайных имён исполнителей и случайных названий задач.

На экранной форме разместите три списка. В первом выводите список исполнителей (имя и производительность) с текущей (первой) задачей (название и сложность). Во втором списке выводите список всех задач конкретного исполнителя в случае, когда в первом списке мышью выбирается некоторый исполнитель. В третьем списке выводите построчно информацию о процессе работы, когда какая-то из задач была завершена (имя исполнителя, название задачи и номер цикла моделирования).

При каждом срабатывании таймера обновляйте первый список, перерисовывая его без моргания.

Кнопка Settings вызывает отдельное окно настроек программы, которое позволяют задавать следующие параметры:

- время срабатывания таймера, в секундах;
- минимальное и максимальное количество исполнителей;
- минимальная и максимальная производительность исполнителя;
- минимальное и максимальное количество задач;
- минимальная и максимальная сложность задачи.

Обеспечьте исходное хранение в программе всех этих настроек по умолчанию.

Изучите, как в вашей среде программирования организуется работа таймера. Как правило, инициализируется стандартный объект Таймер, задаётся периодичность его срабатывания (например, секунда), и указывается функция, которая автоматически вызывается через этот период.