

Тестовое задание “Process Manager”

Под “Process Manager”(PM) мы понимаем компонент, предназначенный для управления процессами внутри операционной системы.

Каждый процесс содержит два поля:

- PID - уникальный неизменяемый идентификатор процесса
- Priority - приоритет процесса(низкий/средний/высокий)

Процесс неизменяемый. Его PID и Priority создаются при рождении и умирают с ним же. Для этого у каждого процесса есть метод kill(), который уничтожает процесс.

Мы хотим, чтобы Process Manager покрывал следующую функциональность: Add new process, List running processes, Kill/KillGroup/KillAll.

Add new process - case 1 (1/3)

В PM должно быть указано максимальное количество процессов, которым он может управлять. Это значение задается в момент старта компонента.

Для добавления нового процесса используется метод add(process). По умолчанию мы можем добавлять новые процессы, пока не превышен лимит, в противном случае мы не принимаем любой новый процесс.

Add new process - case 2 (2/3)

Другой заказчик хочет другое поведение: он хочет, чтобы PM принимал все процессы через метод add(process), но если максимальный лимит процессов исчерпан, то PM должен убивать самые старые процессы (First-In, First-Out).

Add new process - case 3 (3/3)

И снова очередной заказчик и снова новые требования. Теперь при вызове метода add(process), PM должен посмотреть на список текущих процессов. И если новый процесс имеет приоритет выше любого существующего, то убивается самый старый процесс из самых низко-приоритетных. В противном случае мы не добавляем процесс.

List running processes

PM должен иметь возможность вывести список всех текущих процессов через метод list(), с возможностью сортировки их по времени создания(порядку добавления), PID и Priority.

Kill/KillGroup/KillAll

PM предоставляет возможность убивать процессы:

- убить выбранный процесс по PID
- убить все процессы с выбранным приоритетом
- убить все процессы

Требуется реализовать PM на языке Kotlin, поддерживающий вышеприведённые операции.