

ТЕМА: Структура от данни ОПАШКА

**Задача 1:** Да се създаде на езика Java клас **TimedQueue**, който представя опашка със следните свойства: при добавяне на нов елемент в опашката този елемент се снабдява с печат за време. Освен нормалния итератор, опашката трябва да има още три итератора:

- итератор ***iterateBefore(Date time)***, който обхожда всички елементи на опашката с печат за време по-малък или равен на параметъра по възходящ ред на печатите;
- итератор ***iterateAfter(Date time)***, който обхожда всички елементи на опашката с печат за време по-голям или равен на параметъра по възходящ ред на печатите;
- итератор ***iterateBetween(Date from, Time to)***, който обхожда всички елементи на опашката с печат за време по-голям или равен на параметъра ***from*** и по-малък или равен на параметъра ***to*** по възходящ ред на печатите;
- итератор ***iterateBackwards ( )***, който обхожда всички елементи на опашката по низходящ ред на печатите.

<http://java.sun.com/javase/6/docs/api/java/util/Date.html>

**Задача 2:** Да се създаде клас на езика Java, който описва обекти – процеси, характеризиращи се със следните атрибути:

- идентификатор на процес – цяло положително число;
- име на процеса;
- приоритет на процеса – цяло число в интервала [0,15];
- състояние, в което се намира процеса.

Всеки един процес се намира в едно от следните състояния – „*спящ*”, „*готов*” за изпълнение и „*работещ*”.

Във всеки момент има най-много един „*работещ*” процес и произволен брой „*готови*” и „*спящи*” процеси.

Класът **Process** да съдържа:

- конструктор, който създава нов процес с определен приоритет и име и уникален идентификатор и привежда процеса в състояние „*готов*”;

Класът **Condition**, който описва условие, трябва да съдържа:

- опашка, в която се поместват свързаните с условието процеси;
- метод ***wait()***, който добавя процес в опашката като го привежда от състояние „*работещ*” в състояние „*спящ*”;

- метод **signal()**, който отстранява процеса от началото на опашката и сменя състоянието му от „спящ” на „готов”.

Класът **ProcessManager** , който създава необходимите структури за пазене на информацията за процесите и има следните методи:

- метод **create**(String name, int priority), който създава нов процес с указаното име и приоритет и състояние „готов”;
- метод **pick()** - привежда процес, който е в състояние „готов” и има най-висок приоритет в състояние „работещ”;
- метод **preempt**(boolean flag) – привежда процес от състояние „работещ” в състояние „готов” , като намалява приоритета му с единица, ако това е указано.