Пояснительная записка к курсовой работе

Постановка задачи

Разработать планировщик обработки процессов по подобию планировщика задач в операционных системах (ОС).

Вариант № 5

Условие в табл. № 1.

Таблица № 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n/n |  | Стратегия планирования | Наличие вытеснения | Способ организации очереди | | | | Динамическое повышение приоритета |
|  | упорядоченный  список | не упорядочен.  список | список частично упорядочивается  через t тактов | каждому  приоритету своя очередь |
|  |
| метод планирования памяти |
| 5 | 2 | HPF | - | + |  |  |  | + |

Размещение в ОП происходит по первому подходящему.

Данный планировщик задач был разработан и написан на языке Java в IntelliJ IDEA.

Ниже на рис 1-4 представлена схема UML для данной программы.

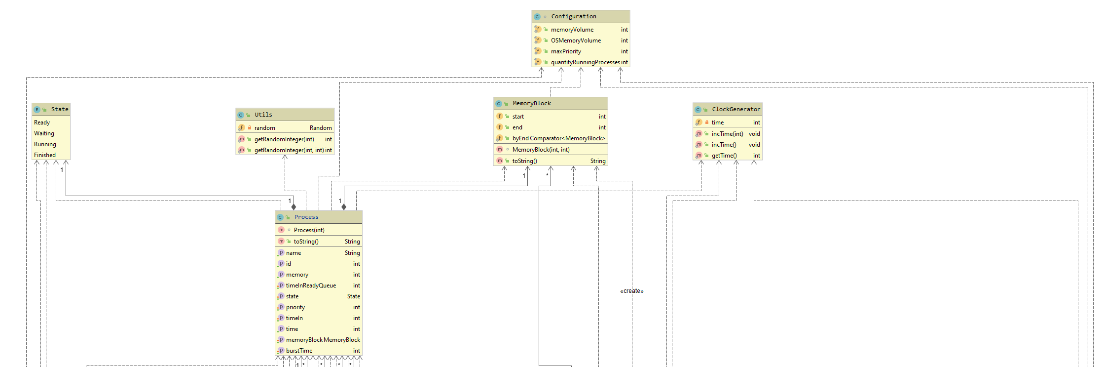


Рис. 1. Схема UML для разработанного планировщика задач (фрагмент 1).

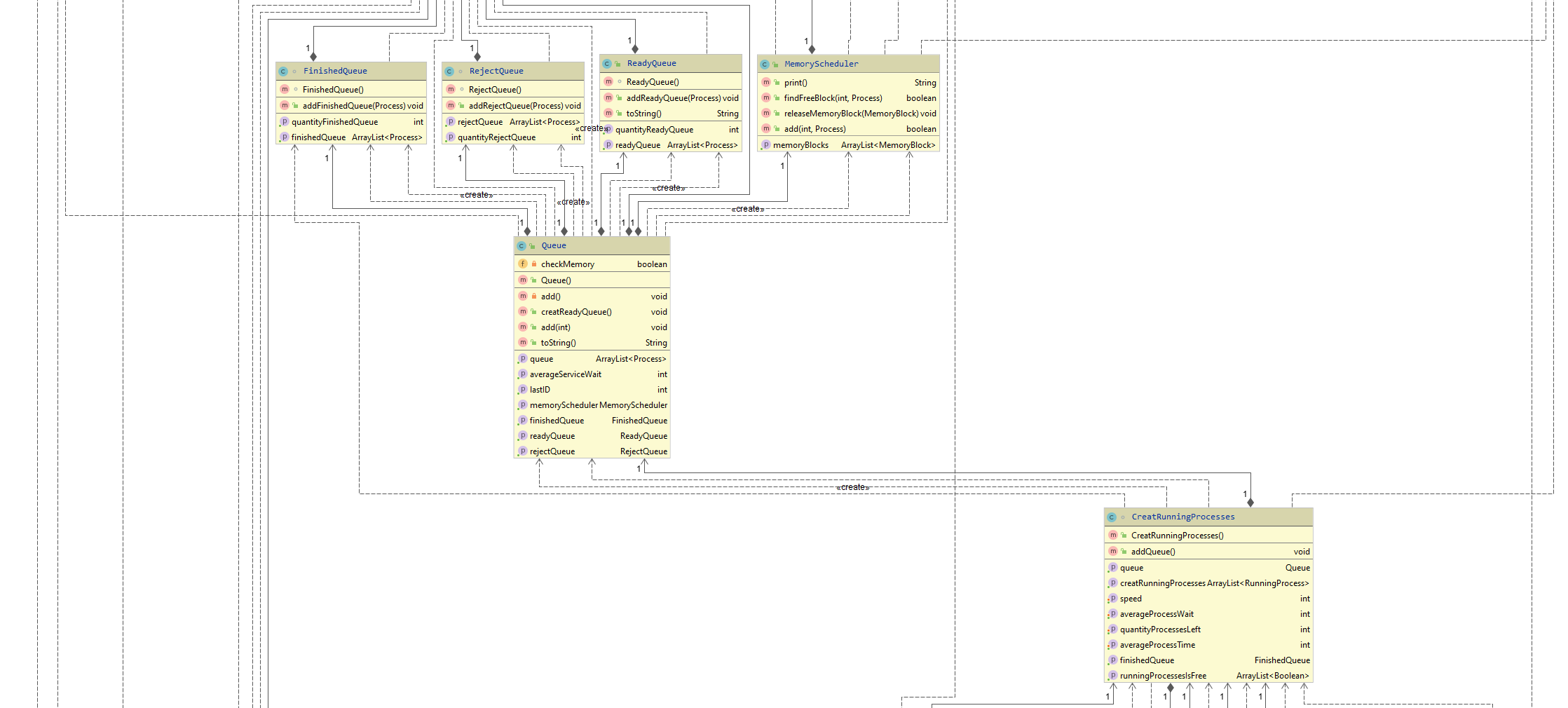


Рис. 2. Схема UML для разработанного планировщика задач (фрагмент 2).

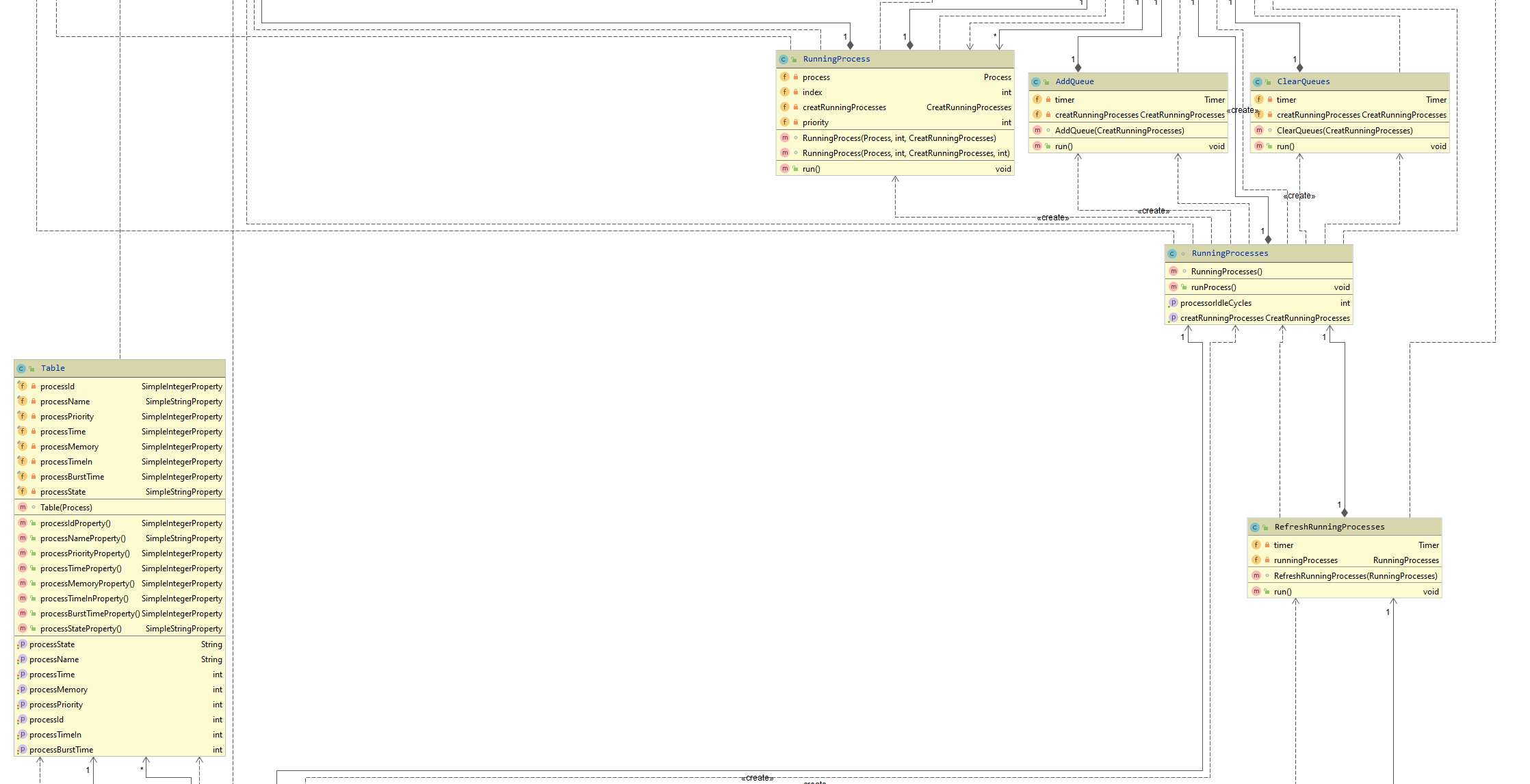


Рис. 3. Схема UML для разработанного планировщика задач (фрагмент 3).

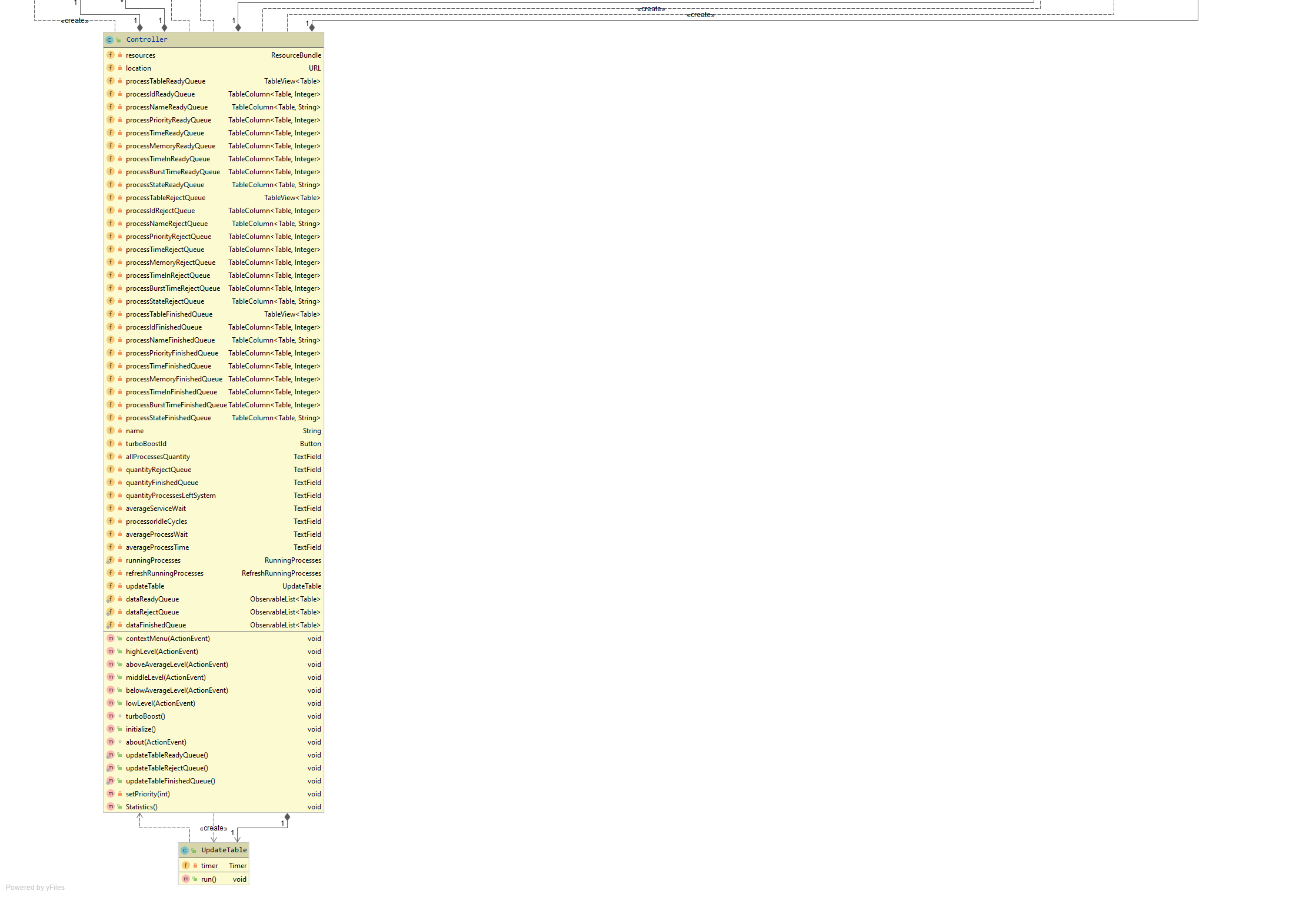


Рис. 4. Схема UML для разработанного планировщика задач (фрагмент 4).

Комментарии каждого элемента написанны в самих классах программы.

Использование программы.

Для большего понимание ниже прикреплён скриншот данной программы рис. 5.

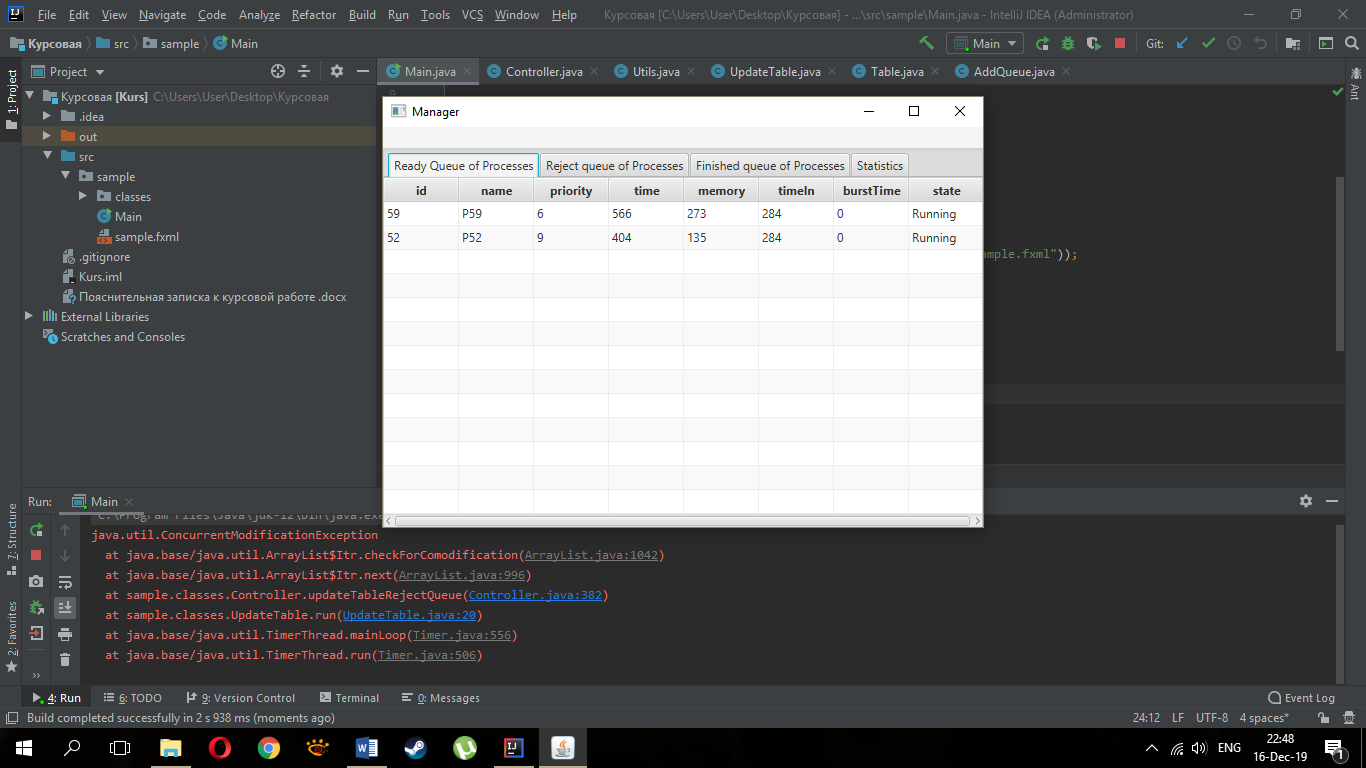


Рис. 5. GUI планировщика задач

Первая вкладка содержит таблицу очереди готовых процессов, в ней отображается свойства процесса.

Вторая вкладка содержит таблицу очереди отказов, туда попадают процессы, которые не поместились в ОП.

Третья вкладка содержит таблицу очереди завершенных процессов.

Четвертая вкладка содержит статистику о планировщики задач рис. 6.

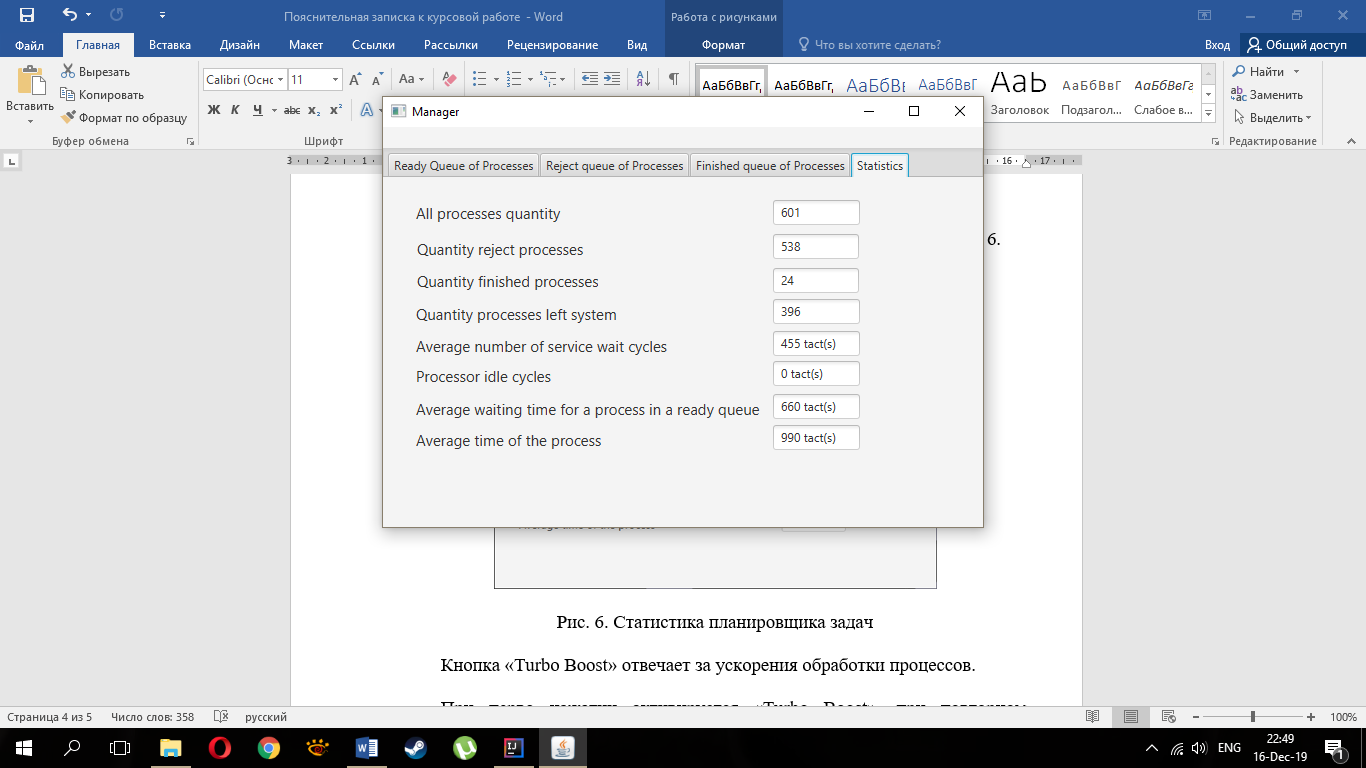


Рис. 6. Статистика планировщика задач

Так же на рис. 7 показано динамическое изменение памяти.

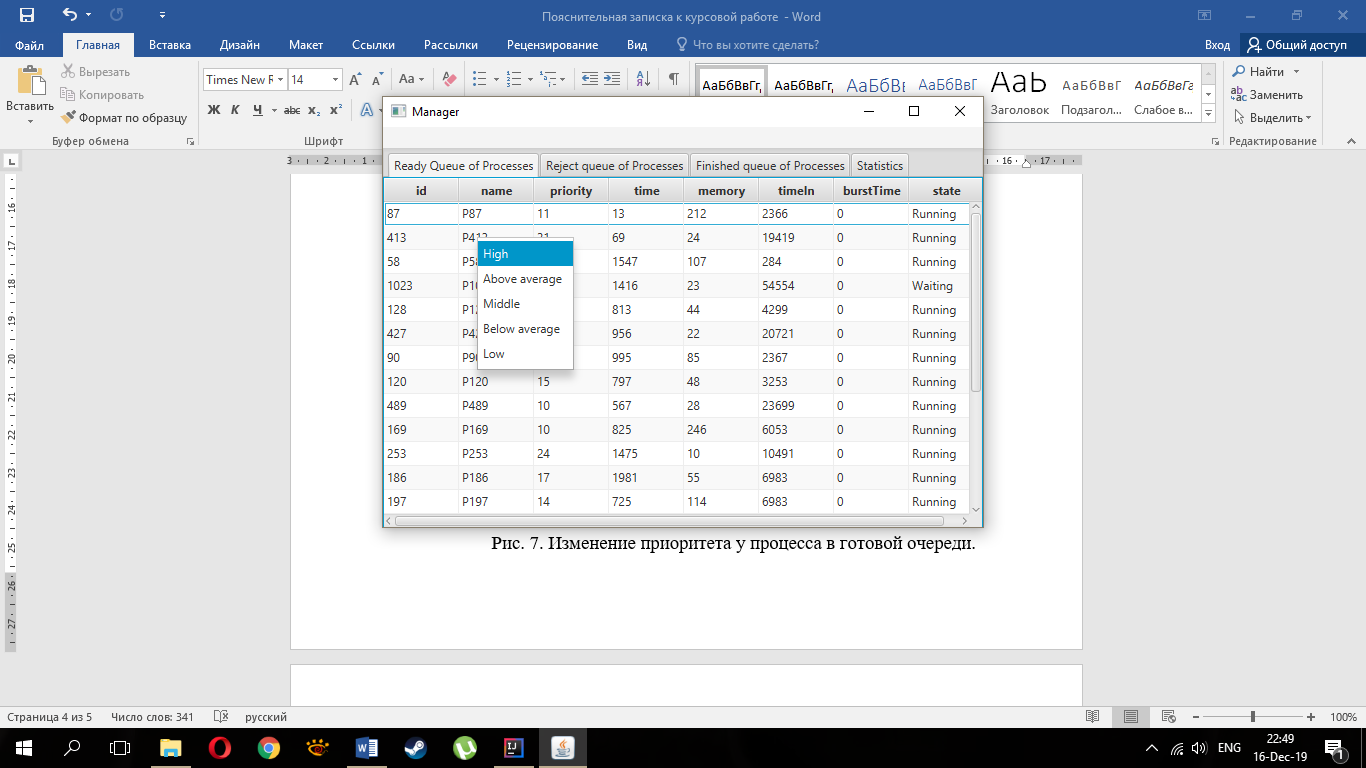


Рис. 7. Изменение приоритета у процесса в готовой очереди.

Выводы

В данной курсовой работе был разработан планировщик задач согласно табл. № 1. На рис. 1-4 представлена схема UML. На рис. 5-7 показано как работает планировщик. В ходе моделирования можно заметить, что отсутствие вытеснения повышает скорость обработки процессов, из-за того, что потоки, которые обрабатывают, требуют поиска в готовой очереди процессов с более высшим приоритетом.

Литература

1. <https://www.jetbrains.com/help/idea/>
2. <https://docs.oracle.com/en/>
3. Таненбаун Э., Бос Х. Современные операционные системы 4-е изд. – СПб.: Питер, 2019. – 1120 с.
4. <https://www.w3schools.com/java/>
5. <https://stackoverflow.com/>
6. Сетевые операционные системы / В. Г. Олифер, Н. А. Олифер. –СПб.: Питер, 2002. – 544 с.
7. Ахо А., Хопкрофт Д., Ульман Д. – Структуры данных и алгоритмы.: Пер. с англ.: Уч. пос.- М., Издательский дом «Вильямс», 2016.– 400 с.