

**Министерство образования и науки Российской Федерации**

**Федеральное агентство по образованию**

**Государственное образовательное учреждение высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э. Баумана)**

**Факультет «Робототехника и комплексная автоматизация» (РК)**

**Кафедра «Системы автоматизированного проектирования» (РК6)**

****

**Практикум по «Теории вероятности».**

**Студент: Сергеева Диана**

**Группа:** РК6-36Б

**Преподаватель:** Берчун Ю.В

Проверил:

Дата:

2020

**Задание:**

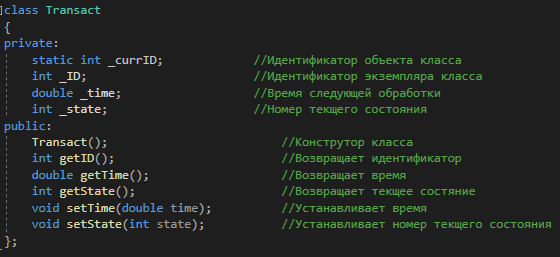
Требуется разработать программу, реализующую дискретно-событийное моделирование системы, рассмотренной в задании 2 домашнего задания №4. Обратите внимание, что все интервалы времени подчиняются законам распределений, носящим непрерывный характер. Поэтому категорически неверными является выбор целочисленных типов данных для моментов и интервалов времени, и тем более инкремент модельного времени с единичным шагом. Нужно реализовать именно переход от события к событию, как это сделано в GPSS и других проблемно-ориентированных системах. Для упрощения можно ограничиться использованием единственного потока случайных чисел для генерации всех необходимых случайных величин. Результатом работы программы должен быть лог-файл, содержащий записи типа: «В момент времени 12.345 транзакт с

идентификатором 1 вошёл в модель», «В момент времени 123.456 транзакт с идентификатором 123 встал в очередь 1», «В момент времени 234.567 транзакт с идентификатором 234 занял устройство 2», «В момент времени 345.678 транзакт с идентификатором 345 освободил устройство 1», «В

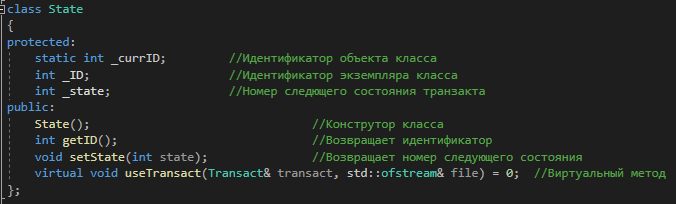
момент времени 456.789 транзакт с идентификатором 456 вышел из модели».

В программе создадим классы:

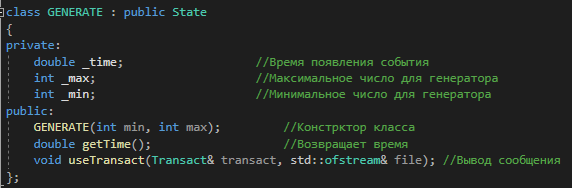
1. Transact с полями:



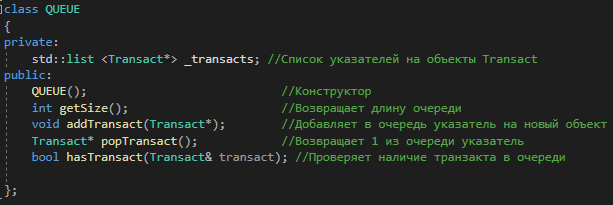
1. State с полями:



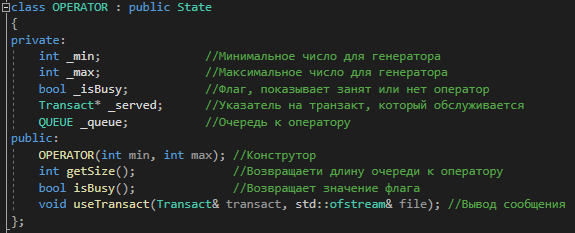
1. GENERATE с полями:



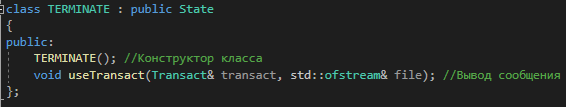
1. QUEUE с полями:



1. OPERATOR с полями:

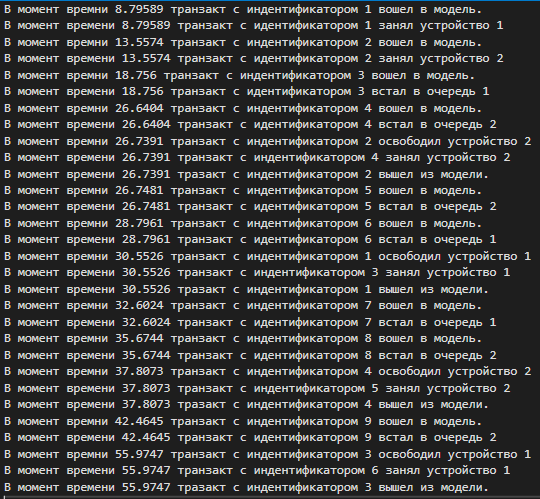


1. TERMINATE с полями:

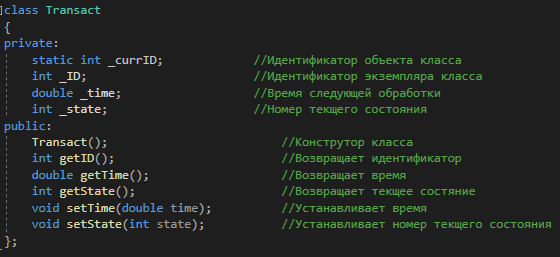


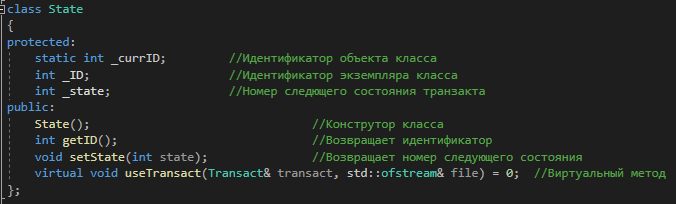
Определяем 2 оператора (2 объекта класса OPERATOR). Создаем список указателей на наши транзакты. Создаём последний транзакт, при котором завершим систему, далее создаём первый транзакт, вносим его в список, создавая при этом новый транзакт. Далее проходим по нашему списку с транзактами и ищем с наименьшем временем, если такой нашёлся меняем его состояние: 0 – транзакт включаем в нашу систему или при необходимости завершаем работу системы; 1 – выбираем оператор помещаем его либо в очередь, либо к оператору; 2 – обработка у 1 оператора; 3 - обработка у 2 оператора; 4 – обработка транзакта закончилась, удаляем его из нашей модели.

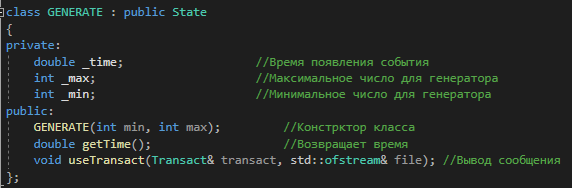
**Результат работы программы:**

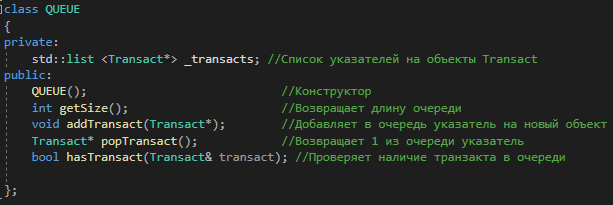
****

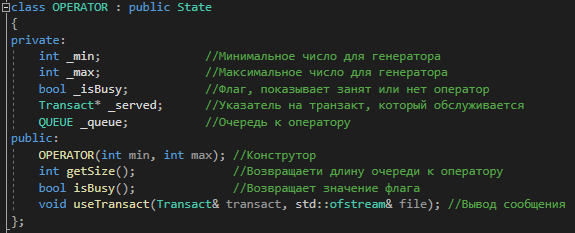
**Текст программы:**

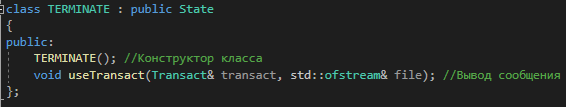


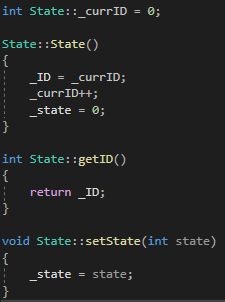
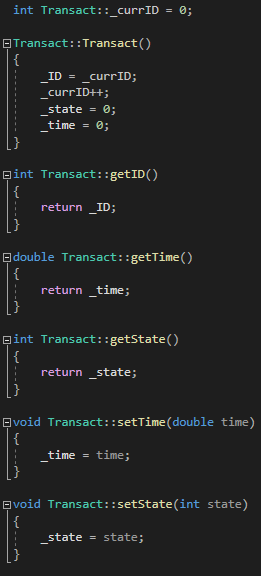


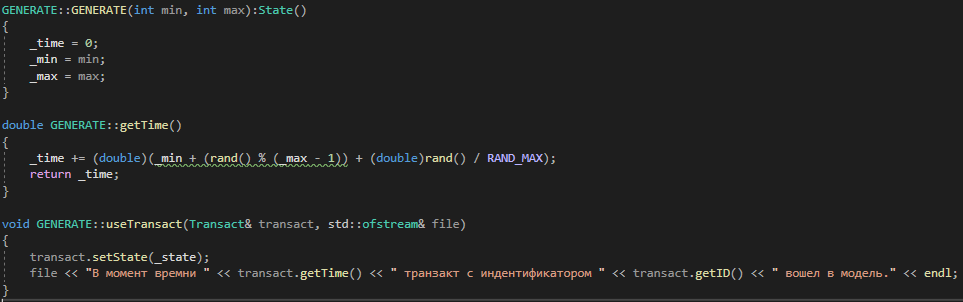


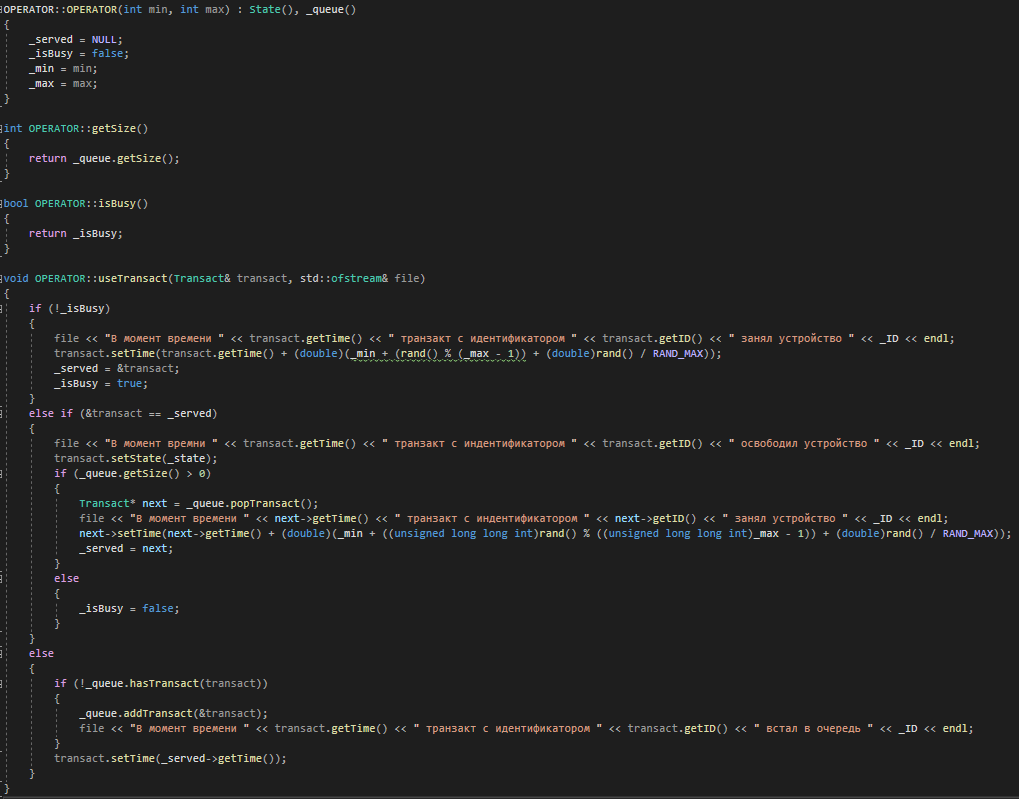


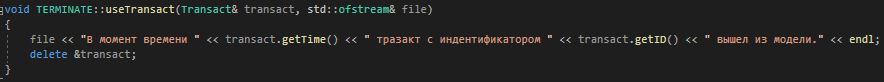


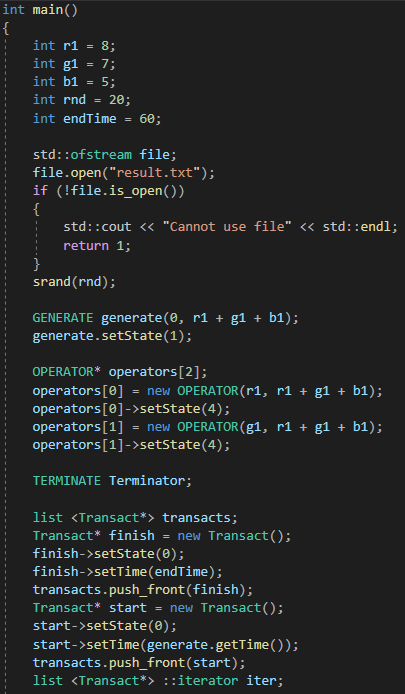


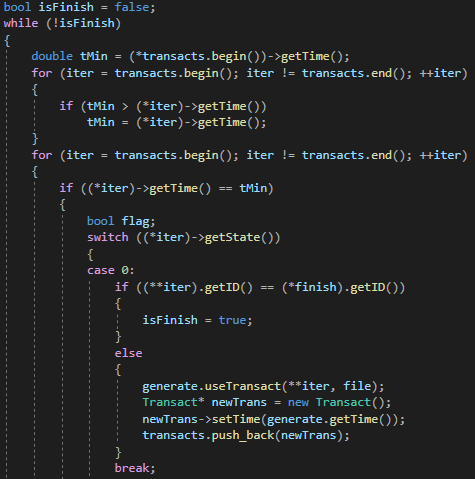
** **

****

****

****

****

****

****