Задача практикума – самостоятельная реализация алгоритма имитационного событийного моделирования дискретных систем (на примере систем массового обслуживания).

Рассмотрим ключевые слова, вынесенные в формулировку задачи. Речь идёт о **дискретных** системах – это значит, что система может быть описана набором дискретных значений. Например, некоторый ресурс может быть занят или свободен, доступен или недоступен, в буферном элементе того или иного объёма могут быть занята определённое число единиц этого объёма, при наличии ситуации ожидания некоторого условия число ожидающих объектов также носит дискретный характер. Дискретных характер описания состояния системы в целом позволяет, во-первых, свести её модель к набору правил взаимодействия, носящих **логико-алгоритмический** характер и не требующий решения обыкновенных дифференциальных, алгебро-дифференциальных уравнений или дифференциальных уравнений в частных производных, характерных для других видов математического и компьютерного моделирования. Во-вторых, изменение состояния дискретной системы производится условно мгновенно, т.е. в процессе моделирования можно выделить лишь отдельные моменты времени, в которые происходят **события**, изменяющие состояние системы. И в процессе моделирования необходимо и достаточно рассматривать поведение системы только в эти моменты времени.

Поэтому в основе процедуры моделирования лежит **дискретно-событийный** механизм продвижения модельного времени. Он состоит из итерационно меняющихся процедур, которые традиционно называются фазами. В **фазе коррекции таймера** осуществляется переход к очередному моменту времени, когда предполагается реализация хотя бы одного события. В **фазе просмотра** при остановленном модельном времени выполняются изменения состояния системы и, при необходимости, прогнозируются события, связанные с будущими изменениями. Кроме того, иногда выделяют ещё **фазу ввода**, в ходе которой производится инициализация состояния системы и выполняется первичное прогнозирование событий.

С точки зрения модели данных оказывается удобным использовать вспомогательную структуру данных, описывающую текущий статус прохождения заявкой (которая в терминах имитационного моделирования обычно называется **транзактом**) маршрута обработки с привлечением ресурсов системы. Взаимодействуя с ресурсами, транзакт меняет их состояние, а значит и состояние системы в целом. Например, в GPSS такая структура данных содержит следующие поля:

* идентификатор транзакта;
* приоритет транзакта;
* момент времени наступления очередного события, связанного с транзактом;
* текущий блок (номер строки в модели), где находится транзакт;
* очередной блок (номер строки в модели), куда транзакт должен быть перемещён.

Номера строк в GPSS-модели в более общем случае можно рассматривать как состояния исполнения процесса обработки, которые удобно визуально представить как точки, расставленные на блок-схеме алгоритма исполнения заявки.

Рассмотренные структуры данных объединяются в наборы (списки), в качестве которых могут выступать такие типовые контейнеры, как связные списки, очереди, очереди с приоритетами и деревья. Традиционно в имитационном моделирования эти списки называются **цепями**. Как минимум говорят о двух цепочках – цепи **будущих событий** (FEC, Future Event Chain, на неё уже обращали внимание в видео про основы работы в среде GPSS) и цепи **текущих событий** (CEC, Current Event Chain).

В **фазе коррекции таймера** в цепи будущих событий находят запись с минимальным прогнозным временем, осуществляют переключение модельного времени на этот момент и переносят все записи с указанным временем из FEC в CEC. В фазе просмотра транзакт продвигается по маршруту обработки (на примере GPSS – изменяются номера строк, через которые проходит транзакт). Это продолжается до тех пор, пока не возникнет одна из следующих ситуаций:

* транзакт полностью завершил обслуживание – в этом случае соответствующая запись просто удаляется из цепи текущих событий, выделенная память освобождается (это важно, поскольку при моделировании систем с большим числом запросов может потребоваться выделение значительных объёмов памяти; кстати, по той же причине, новые транзакты из одного и того же источника вводятся в систему по одному, а не планируются моменты времени ввода всех транзактов на весь планируемый период моделирования);
* транзакт инициировал операцию, требующую времени (в GPSS – это прежде всего попадание в блок ADVANCE) – в этом случае к текущему модельному времени добавляется прогнозируемое время на выполнение этой операции, это значение фиксируется в соответствующем поле рассмотренной выше структуры данных, и она переносится из CEC в FEC;
* транзакт претендует на занятие какого-либо ресурса, который сейчас занят/не доступен (или ждёт выполнения какого-либо логического условия, которое на данный момент не выполняется) – в наиболее простом случае (подробнее рассмотренном здесь - <http://bigor.bmstu.ru/?cnt/?doc=gpss_model/sost_zep.mod/?cou=gpss_model/gpss_lab2.cou>) транзакт остаётся в CEC, но в качестве момента времени прописывается признак необходимости продвижения транзакта «как можно раньше»; тогда при каждом изменении состояния системы вследствие продвижения других транзактов, будут анализироваться условия продвижения транзактов, помеченных признаком «как можно раньше»; в более сложном случае, транзакты перемещаются из CEC в отдельные цепочки, которые могут быть ассоциированы с конкретным ресурсом; это рациональнее, поскольку не надо анализировать все ожидающие транзакты в системе, а при изменении состояния ресурса опрашивать только ассоциированную с ним цепочку; кроме того, введение отдельных цепочек может быть инициировано разработчиком для реализации более сложных дисциплин обслуживания (отличающихся от классической очереди типа FIFO).

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».