

## **ЛАБОРАТОРНАЯ РАБОТА №3**

### **КОНСТРУИРОВАНИЕ ДИСКРЕТНО-СОБЫТИЙНЫХ МОНИТОРОВ ДЛЯ СИСТЕМ МОДЕЛИРОВАНИЯ**

Сконструировать дискретно-событийный монитор, реализующий набор операторов, соответствующих одному из перечисленных ниже методов управления. Отобразить ход часов модельного времени и проиллюстрировать динамику функционирования монитора моделирования.

Отчет по работе должен содержать:

1. Задание.
2. Описание реализации дискретно-событийного монитора (классов).
3. Листинг программы.
4. Результаты работы программы, реализующей дискретно-событийную модель (визуализация динамики) с пояснениями и выводами.

**Класс LIST не использовать! Для реализации динамических структур данных разрабатывать собственные классы!**

## **ЗАДАНИЯ**

### **1. Императивное управление без приоритетов уведомлений**

1. Включить в календарь уведомление о событии  $EV_i$  в абсолютном времени.
2. Включить в календарь уведомление о событии  $EV_i$  перед событием  $EV_j$ .
3. Отменить все события  $EV_i$ , запланированные в календаре.
4. Определить, сколько событий запланировано на заданный момент времени.
5. В течение заданного промежутка времени выполнить действия, соответствующие событиям, запланированным в календаре.

### **2. Императивное управление без приоритетов уведомлений**

1. Включить в календарь уведомление о событии  $EV_i$  в относительном времени.
2. Включить в календарь уведомление о событии  $EV_i$  после события  $EV_j$ .
3. Отменить все события  $EV_i$ , запланированные в календаре на заданный момент времени.
4. Определить запланированное время наступления события  $EV_i$ .
5. Выполнить все действия, соответствующие событиям, запланированным в календаре.

### **3. Императивное управление с приоритетами уведомлений**

1. Включить в календарь уведомление о событии  $EV_i$  в абсолютном времени в заданном приоритетном классе.
2. Включить в календарь уведомление о событии  $EV_i$  перед событием  $EV_j$  в заданном приоритетном классе.
3. Отменить все события  $EV_i$ , запланированные в календаре в заданном приоритетном классе.
4. Определить, сколько событий запланировано на заданный момент времени в каждом из приоритетных классов.
5. В течение заданного промежутка времени выполнить действия, соответствующие событиям, запланированным в календаре.
6. Изменить приоритет события  $EV_i$ .

#### **4. Императивное управление с приоритетами уведомлений**

1. Включить в календарь уведомление о событии  $EV_i$  в относительном времени в заданном приоритетном классе.
2. Включить в календарь уведомление о событии  $EV_i$  после события  $EV_j$  в заданном приоритетном классе.
3. Отменить все события  $EV_i$ , запланированные в календаре на заданный момент времени в каждом из приоритетных классов.
4. Определить запланированное время наступления события  $EV_i$  и его приоритет .
5. Выполнить все действия, соответствующие событиям, запланированным в календаре.
6. Изменить приоритет события  $EV_i$ .

#### **5. Императивное управление без приоритетов уведомлений при наличии в системе нескольких таймеров**

В системе имеется один таймер с нулевым смещением показаний, называемый системным. Все остальные таймеры задаются смещением их показаний относительно текущих показаний системного таймера.

1. Включить в календарь уведомление о событии  $EV_i$  под таймер  $T_j$  в абсолютном времени этого таймера.
2. Определить значение текущего времени в шкале таймера  $T_j$ .
3. Оцифровать таймер  $T_j$  абсолютным значением времени  $TT$ , т.е. определить смещение показаний таймера  $T_j$  относительно показаний системного таймера и изменить положение меток соответствующих событий в календаре.
4. Отменить все события, связанные с таймером  $T_j$ .
5. В течение заданного промежутка времени выполнить действия, соответствующие событиям, запланированным в календаре.

## **6. Императивное управление без приоритетов уведомлений при наличии в системе нескольких таймеров**

В системе имеется один таймер с нулевым смещением показаний, называемый системным. Все остальные таймеры задаются смещением их показаний относительно текущих показаний системного таймера.

1. Включить в календарь уведомление о событии  $EVi$  под таймер  $Tj$  в относительном времени этого таймера.
2. Определить число событий под таймером  $Tj$ .
3. Оцифровать таймер  $Tj$  абсолютным значением времени  $TT$ , т.е. определить смещение показаний таймера  $Tj$  относительно показаний системного таймера и изменить положение меток соответствующих событий в календаре.
4. Отменить событие  $EVi$ , связанное с таймером  $Tj$ .
5. Выполнить все действия, соответствующие событиям, запланированным в календаре.

## **7. Интеррогативное управление с использованием сигнальной синхронизации**

1. Связать наступление события  $EVi$  с подачей сигнала  $Sj$  (ожидание подачи сигнала  $Sj$ ).
2. Подать сигнал  $Sj$ .
3. Сбросить сигнал  $Sj$ .
4. Определить статус сигнала  $Sj$ .
5. Определить, имеются ли в системе уведомления, ожидающие подачи сигнала  $Sj$ .
6. Запланировать выполнение действий, соответствующих событиям, до тех пор, пока в системе имеется хотя бы один активный сигнал, подачи которого ожидает хотя бы одно уведомление.

## **8. Интеррогативное управление с использованием сигнальной синхронизации в сочетании с императивным управлением без приоритетов уведомлений**

1. Связать наступление события  $EVi$  с подачей сигнала  $Sj$  (ожидание подачи сигнала  $Sj$ ).
2. Подать сигнал  $Sj$ .
3. Сбросить сигнал  $Sj$ .
4. Включить в календарь уведомление о событии  $EVi$  в абсолютном времени.
5. Запланировать выполнение действий, соответствующих событиям, до тех пор, пока календарь событий не пуст и в системе имеется хотя бы один активный сигнал, подачи которого ожидает хотя бы одно уведомление.

## **9. Интеррогативное управление с использованием ситуационного управления (управления по условию)**

1. Создать ситуацию  $Si$ , определяемую логическим выражением  $Cj$ .
2. Связать наступление события  $EVi$  с ситуацией  $Si$ .
3. Определить статус ситуации  $Si$ .
4. Определить, есть ли уведомления, ожидающие активизации ситуации  $Si$ .
5. Отменить ситуацию  $Si$ , определяемую логическим выражением  $Cj$ .
6. Разрушить связь между событием  $EVi$  и ситуацией  $Si$ .
7. Запланировать выполнение действий, соответствующих событиям, до тех пор, пока в системе имеется хотя бы одна активная ситуация, с которой связано хотя бы одно уведомление.

## **10. Интеррогативное управление с использованием ситуационного управления (управления по условию) в сочетании с императивным управлением без приоритетов уведомлений**

1. Создать ситуацию  $Si$ , определяемую логическим выражением  $Cj$ .
2. Связать наступление события  $EVi$  с ситуацией  $Si$ .
3. Определить, есть ли уведомления, ожидающие наступления ситуации  $Si$ .
4. Отменить ситуацию  $Si$ , определяемую логическим выражением  $Cj$ .

5. Включить в календарь уведомление о событии  $EV_i$  в относительном времени.
6. Запланировать выполнение действий, соответствующих событиям, до тех пор, пока календарь событий не пуст и в системе имеется хотя бы одна активная ситуация, с которой связано хотя бы одно уведомление.

## **11. Схема состояний (конечный автомат, функционирующий в модельном времени)**

1. Определить состояние  $S_i$  автомата (т.е., идентификатор данного состояния, идентификаторы состояний, в которые возможны переходы из данного, и признаки переходов).
2. Определить действия  $F_i$ , которые должны выполняться в состоянии  $S_i$ .
3. Определить условный переход, т.е. запланировать переход из состояния  $i$  в состояние  $j$  в связи с выполнением условия, задаваемого логическим выражением  $C_j$ .
4. Определить текущее состояние автомата.
5. Запустить автомат (автомат функционирует до тех пор, пока он не перейдет в такое состояние, из которого нет ни одного перехода).

## **12. Схема состояний (конечный автомат, функционирующий в модельном времени) в сочетании с императивным управлением без приоритетов уведомлений**

1. Определить состояние  $S_i$  автомата (т.е., идентификатор данного состояния, идентификаторы состояний, в которые возможны переходы из данного, и признаки переходов).
2. Определить действия  $F_i$ , которые должны выполняться в состоянии  $S_i$ .
3. Определить безусловный переход, т.е. запланировать переход из состояния  $i$  в состояние  $j$  через промежуток времени, задаваемый значением арифметического выражения  $A_i$ .
4. Определить множество алгоритмов формирования уведомлений.
5. Запустить автомат (автомат функционирует до тех пор, пока он не перейдет в такое состояние, из которого нет ни одного перехода).

### **13. Схема процессов (очереди в блоках имеют ограниченную длину и ограниченное время пребывания транзактов)**

1. Реализовать процесс, определяющий модель пути движения транзакта:
  - ввести транзакт в модель (сгенерировать транзакт),
  - транзакт занимает устройство,
  - задержка движения транзакта на заданное время (обслуживание в устройстве),
  - транзакт освобождает устройство,
  - вывести транзакт из модели (уничтожить транзакт).
2. Определить множество алгоритмов генерации транзактов и формирования задержек.
3. Транзакты, выводимые из очереди по причине переполнения или превышения времени пребывания, покидают модель. Определить вероятность того, что транзакт покинет модель необслуженным.
4. Сгенерировать и провести по блокам модели 100 транзактов.
5. Отобразить таблицу транзактов и таблицу блоков.

### **14. Схема процессов (очереди в блоках имеют неограниченную длину и неограниченное время пребывания транзактов)**

1. Реализовать процесс, определяющий модель пути движения транзакта:
  - ввести транзакт в модель (сгенерировать транзакт),
  - транзакт занимает устройство,
  - задержка движения транзакта на заданное время (обслуживание в устройстве),
  - транзакт освобождает устройство,
  - вывести транзакт из модели (уничтожить транзакт).
2. Определить множество алгоритмов генерации транзактов и формирования задержек.
3. Определить среднее время пребывания транзакта в модели.
4. Сгенерировать и провести по блокам модели 100 транзактов.
5. Отобразить таблицу транзактов и таблицу блоков.

## **15. Императивное управление без приоритетов уведомлений**

1. Включить в календарь уведомление о событии  $EVi$  в абсолютном времени.
2. Включить в календарь уведомление о событии  $EVi$  на заданный момент времени перед другими событиями с таким же значением времени (в абсолютном времени).
3. Отменить события всех типов, запланированные в календаре на заданный момент времени.
4. Определить, сколько событий каждого типа запланировано на заданный момент времени.
5. В течение заданного промежутка времени выполнить действия, соответствующие событиям, запланированным в календаре.

## **16. Императивное управление без приоритетов уведомлений**

1. Включить в календарь уведомление о событии  $EVi$  в относительном времени.
2. Включить в календарь уведомление о событии  $EVi$  на заданный момент времени после других событий с таким же значением времени (в относительном времени).
3. Отменить все события заданного типа  $EVi$ , запланированные в календаре.
4. Определить запланированное время наступления события  $EVi$ .
5. Выполнить все действия, соответствующие событиям, запланированным в календаре.

## **17. Схема процессов (предотвращение тупика методом линейного упорядочивания типов ресурсов, необходимых процессу)**

1. Построить список процессов.
2. Построить иерархию типов ресурсов, необходимых процессу.
3. Реализовать операцию выделения ресурса заданного типа процессу.
4. Реализовать операцию использования процессом ресурса заданного типа.
5. Реализовать операцию освобождения процессом ресурса заданного типа.
6. Смоделировать последовательность операций выделения, использования и освобождения ресурсов для нескольких процессов.



## **18. Схема процессов (предотвращение тупика с использованием матриц запросов и матриц используемых ресурсов)**

1. Построить список процессов.
2. Построить матрицу запросов и матрицу используемых ресурсов для процесса.
3. Реализовать операцию выделения ресурса заданного типа процессу.
4. Реализовать операцию использования процессом ресурса заданного типа.
5. Реализовать операцию освобождения процессом ресурса заданного типа.
6. Смоделировать последовательность операций выделения, использования и освобождения ресурсов для нескольких процессов.