

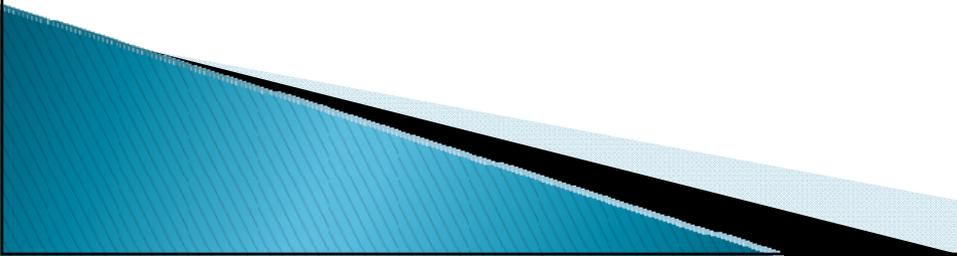
Трансляция описаний конечных автоматов, представленных в формате MS Visio, в исходный код на языке C

Леонид Столяров, 8 класс, лицей «Вторая школа», Москва

Научный руководитель: И. Р. Дединский, лицей «Вторая школа», Москва

Консультант: д.т.н., проф. А. А. Шалыто, ИТМО, Санкт-Петербург

ТРЕБОВАНИЯ К ПРОГРАММЕ

- ▶ Обеспечение эффективности отладки и тестирования автоматных программ
 - ▶ Повышение эффективности ручного и автоматического кодирования программ
 - ▶ Автоматическое документирование логики работы автоматной программы
- 

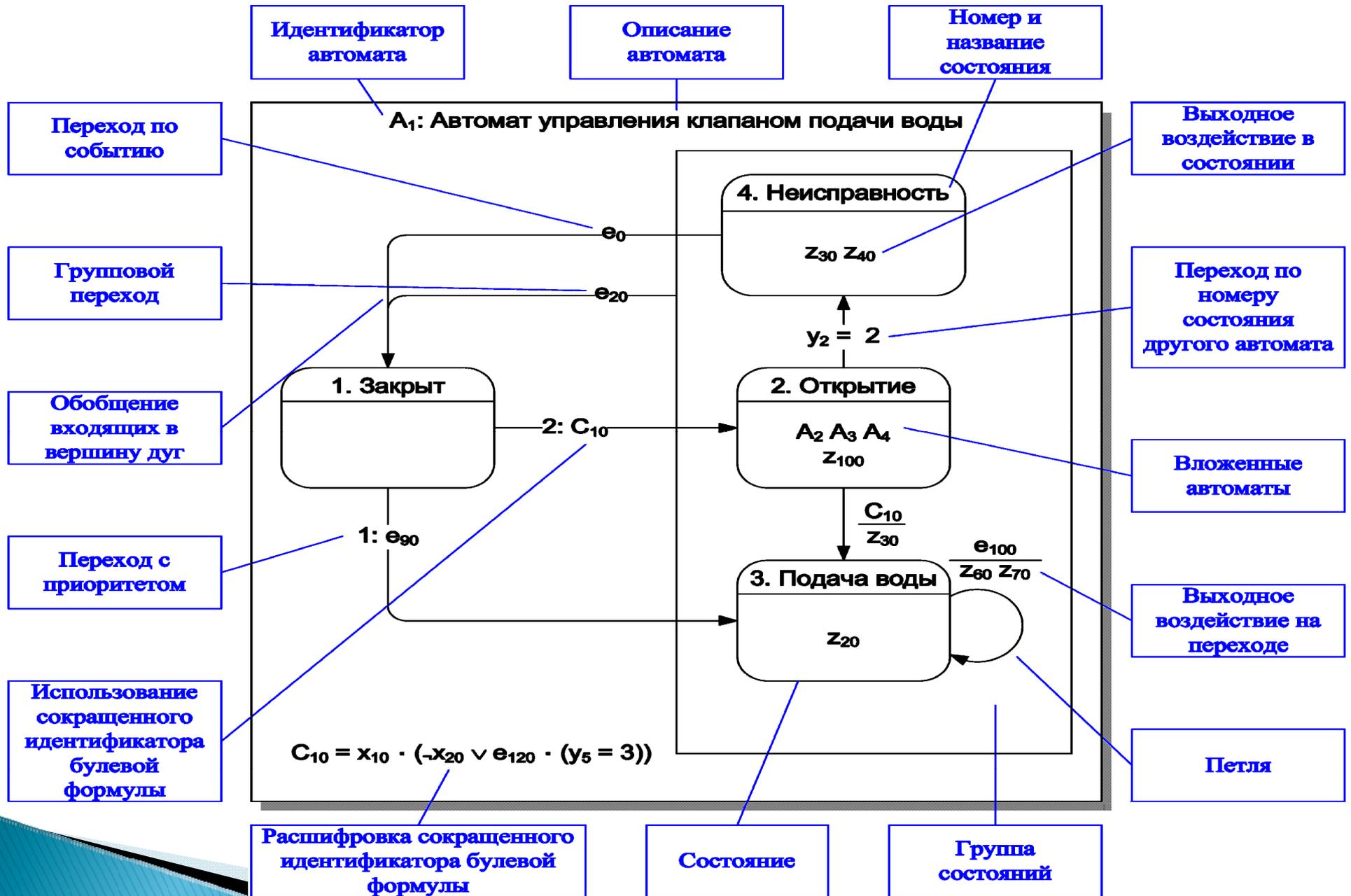
ЦЕЛЬ РАБОТЫ

Разработка программы – транслятора
автоматных графов

ЗАДАЧИ

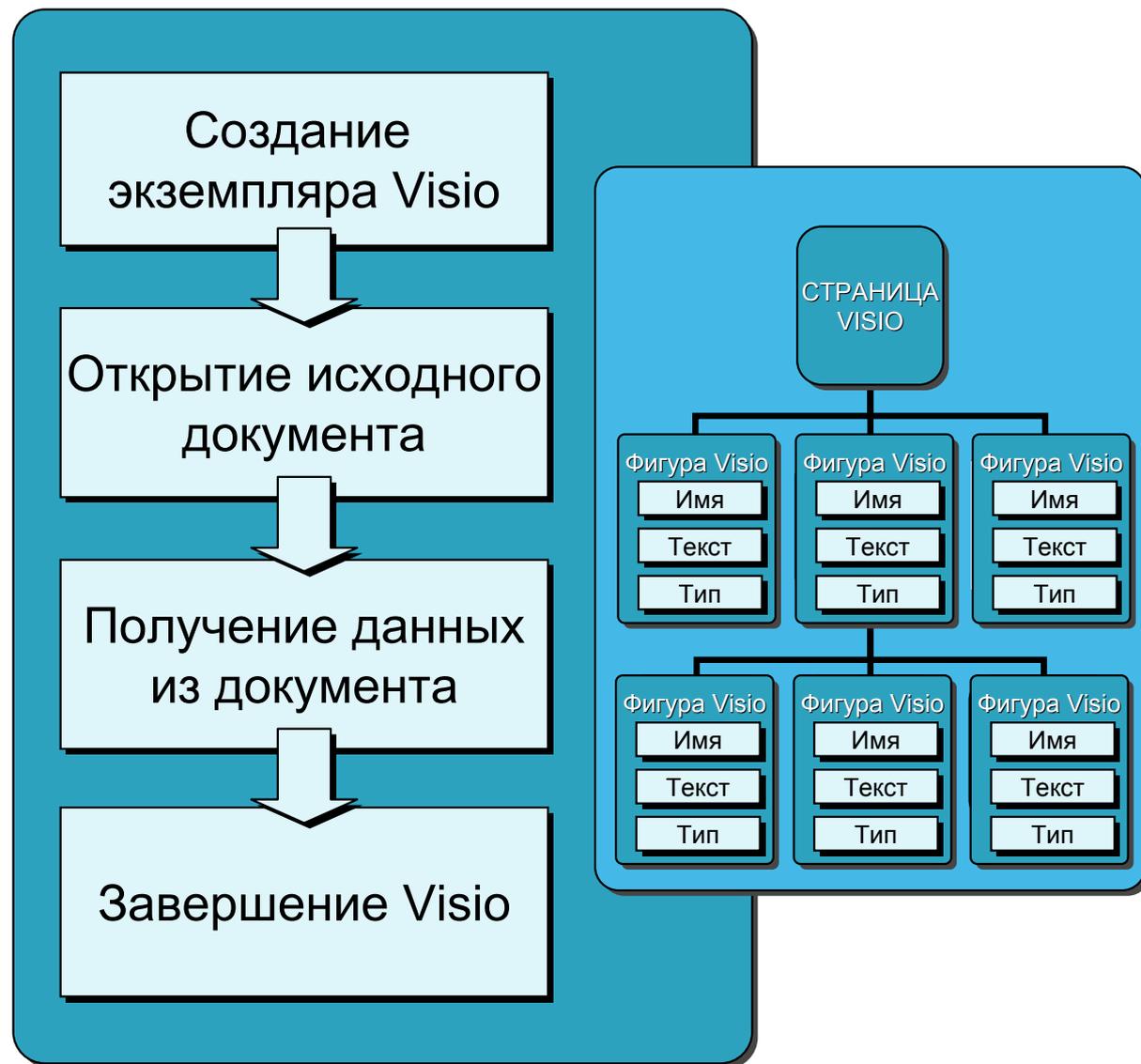
- ▶ Трансляция описаний автоматов, представленных в формате Visio, в код на С
- ▶ Сохранение исходных данных в файл XML
- ▶ Качественная диагностика ошибок в исходных данных
- ▶ Высокое качество и надёжность генерируемого кода
- ▶ Лёгкая интеграция в автоматические процессы сборки приложений

ФОРМАТ ОПИСАНИЯ АВТОМАТА



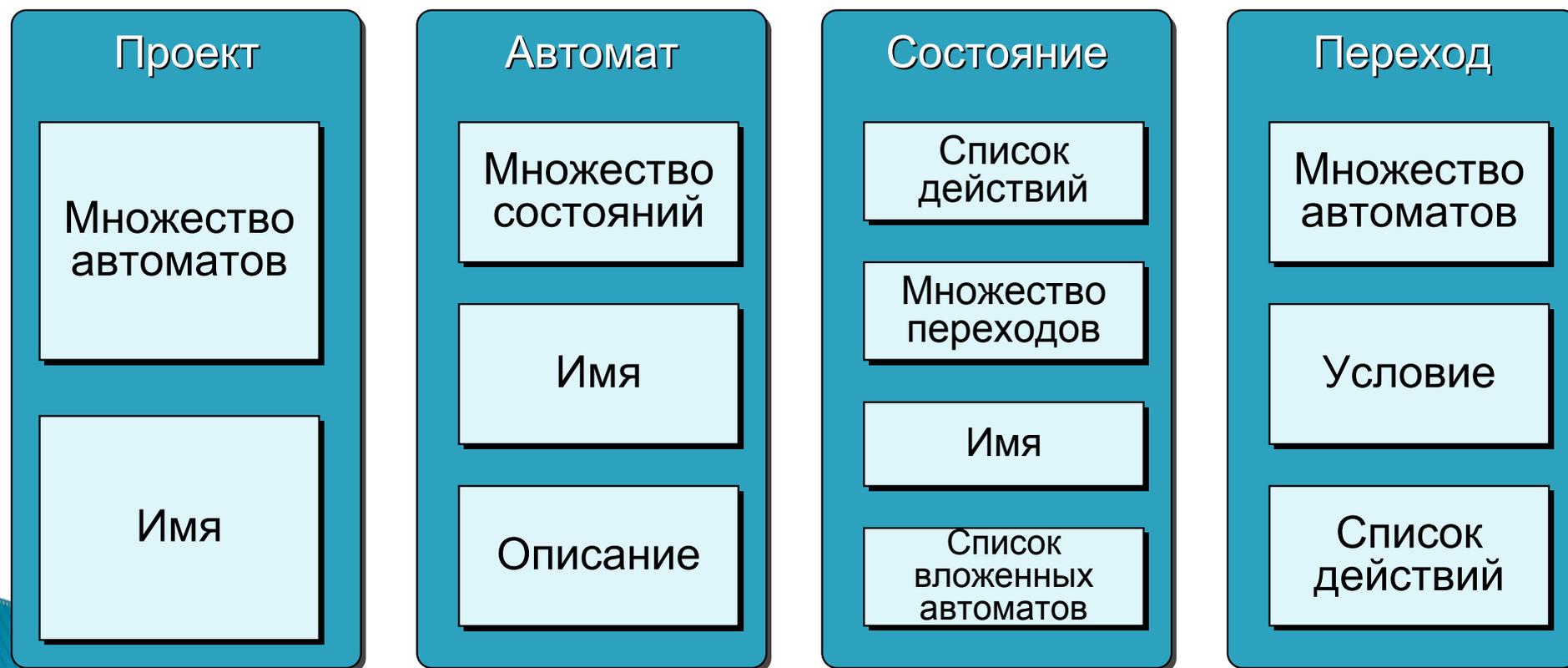
ВЗАИМОДЕЙСТВИЕ С MS VISIO

- ▶ Взаимодействие происходит через COM-объект Visio
- ▶ Внутренняя структура документа Visio – это иерархический набор фигур
- ▶ Импорт схемы из Visio заключается в обходе дерева фигур
- ▶ Фигуры элементов графа переходов загружаются из файла шаблонов Visio



ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ ДАННЫХ

- ▶ Внутренне исходные данные Visio представляются в виде модели, основанной на переходах и состояниях.



АЛГОРИТМ ТРАНСЛЯЦИИ

Чтение данных из исходного документа



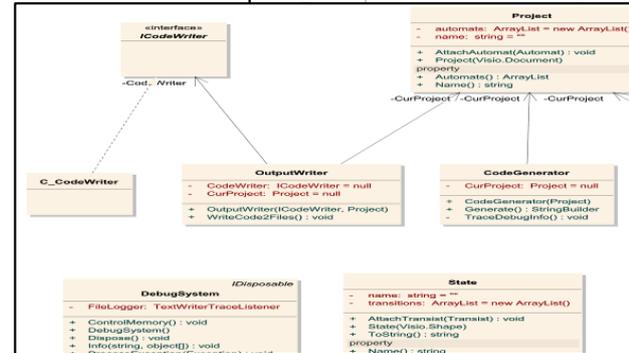
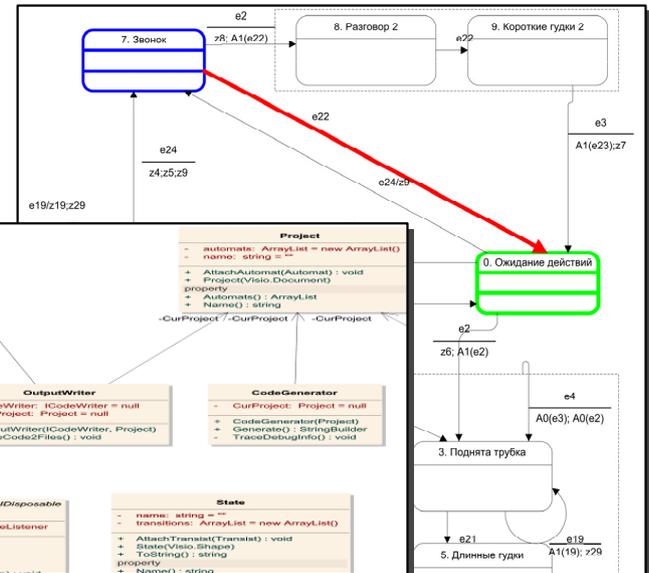
Преобразование исходных данных в модель



Генерация файла XML



Генерация кода на языке C



```

- <model name="Drawing1">
- <stateMashine name="A0" description="Светофор">
  <NodeDescriptions />
- <state name="" description="3.Красный">
  <connection source="3.Красный" target="1.Желтый" condition="e1" actions="z1" />
</state>
- <state name="" description="1.Желтый">
  <connection source="1.Желтый" target="2.Зеленый" condition="e2" actions="z2" />
  <connection source="1.Желтый" target="0.Выключен" condition="x1 > 12.00"
  actions="z4" />
</state>
- <state name="" description="2.Зеленый">
  <connection source="2.Зеленый" target="3.Красный" condition="e3" actions="z3" />
  <connection source="2.Зеленый" target="5.Длинные гудки" condition="x1 > 7.00"
  actions="z5" />
</state>
  
```

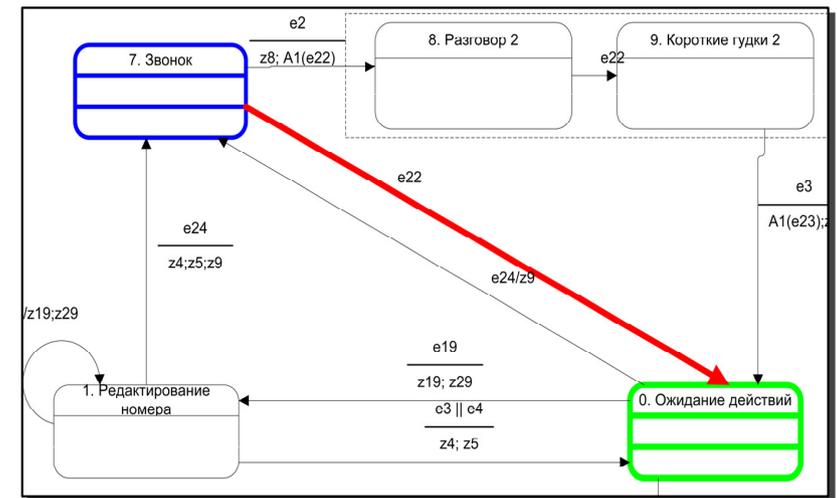
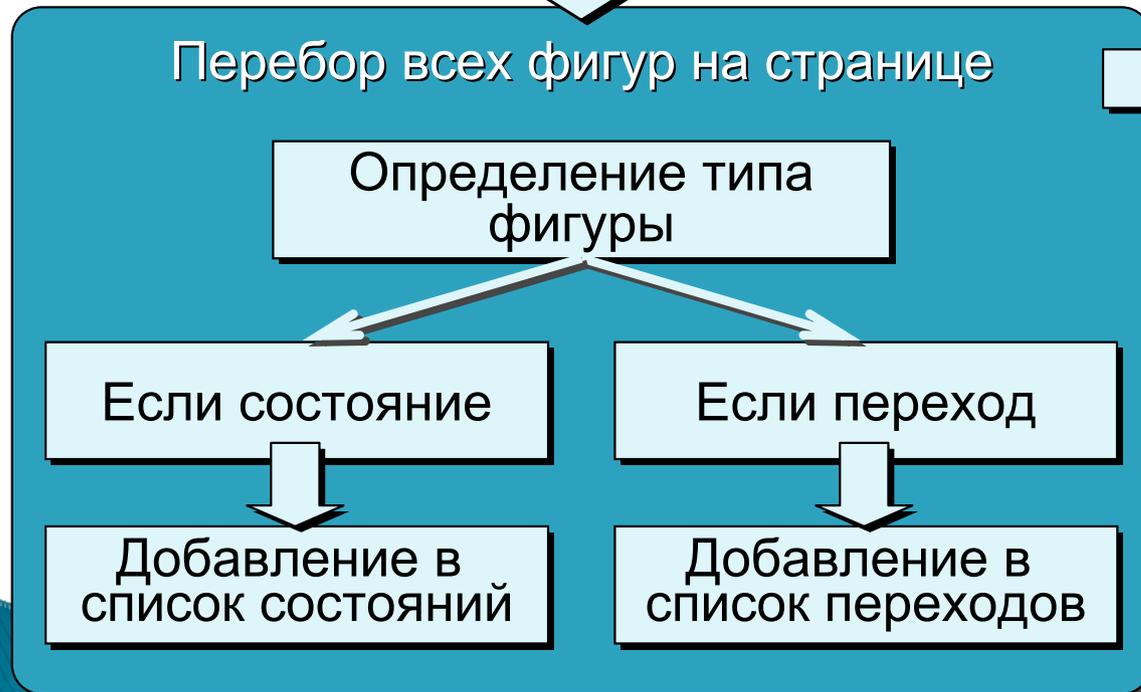
```

18: switch (g_AutomataData.y_A0)
19: {
20:
21:     case 0:
22:         /*3.Красный*/
23:         Automaton_A0 (Event);
24:         if (Event == 0)
25:         {
26:
27:             g_AutomataData.y_A0 = 1;
28:         }
29:
30:         break;
31:
32:     case 1:
33:         /*1.Желтый*/
34:         Automaton_A0 (Event);
35:         if (Event == 1)
36:         {
37:
38:             g_AutomataData.y_A0 = 2;
  
```

ГЕНЕРАЦИЯ МОДЕЛИ ДАННЫХ

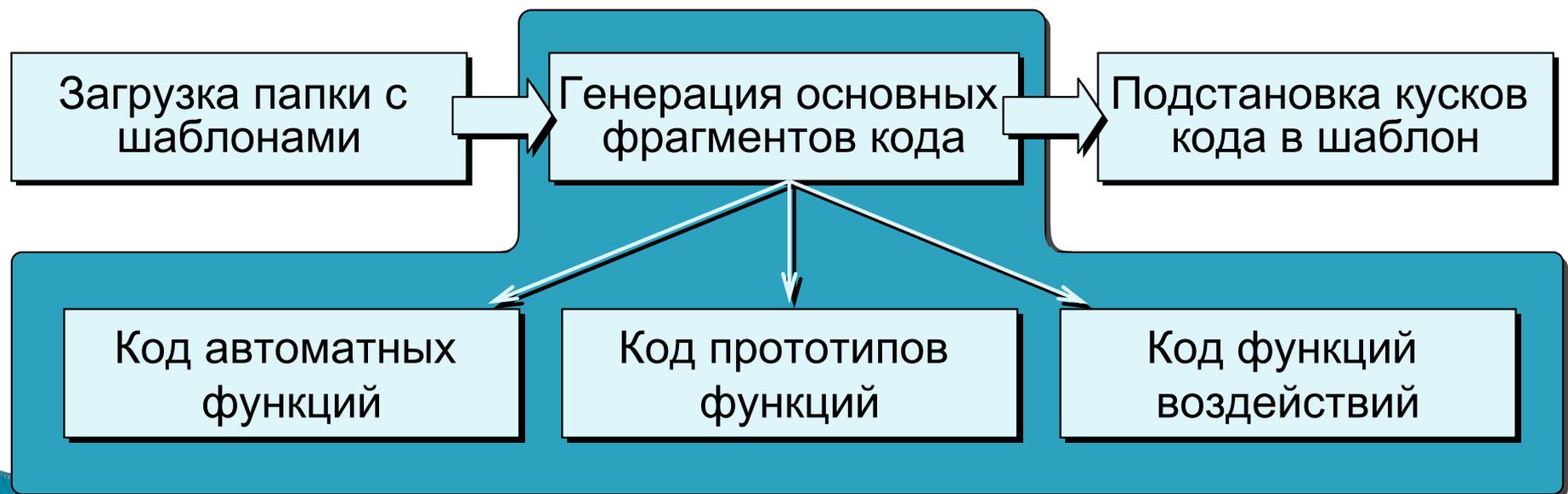
- ▶ Процесс преобразования – однократный

Создание объекта автомата



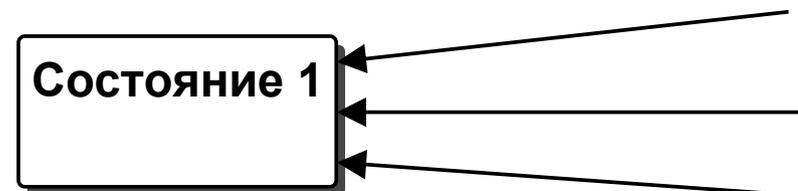
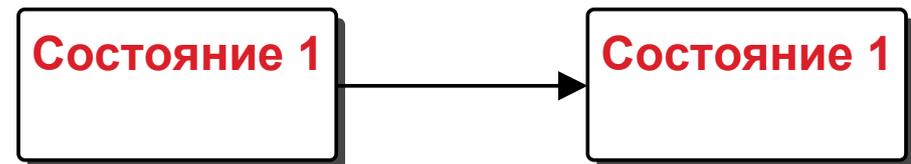
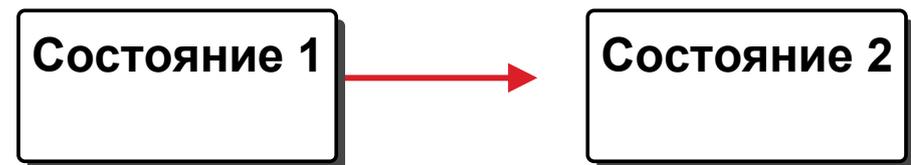
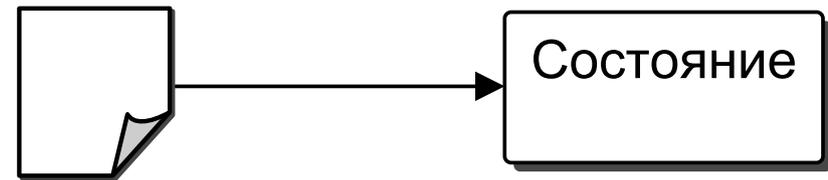
ГЕНЕРАЦИЯ ФАЙЛОВ С КОДОМ

- ▶ Генератор устроен так, что генерацию основных частей исходного кода производит не самостоятельно, а обращаясь к объекту генерации кода
- ▶ Генерация происходит с использованием набора файлов-шаблонов



ТИПЫ ОПРЕДЕЛЯЕМЫХ ОШИБОК

- ▶ Переход соединяет не состояния, а другие фигуры
- ▶ Переход не присоединён к фигуре
- ▶ В автомате найдены одинаковые элементы
- ▶ Из состояния не исходит ни один переход



ПРИМЕР РАБОТЫ ТРАНСЛЯТОРА

The image displays the Microsoft Visual Studio environment with the following components:

- Code Editor:** Shows a C++ code snippet for a state machine. The code includes comments in Russian: `/*2.Зеленый*/` and `/*0.Выключен*/`. It defines a state `Automaton_A0` and uses `if` statements to handle events `Event == 2` and `x1() > 7.00`.
- Solution Explorer:** Shows the project structure for 'AutomatonTest', with the file `test.vsd` circled in red.
- Visio Diagram:** A state transition diagram with four states: `0.Выключен`, `1.Желтый`, `2.Зеленый`, and `3.Красный`. Transitions are labeled with events `e1`, `e2`, and `e4`, and conditions `z1`, `z2`, and `z4`. A legend at the bottom right explains the events: `e1` is 'Время красного истекло' (Red time expired), `e2` is 'Время желтого истекло' (Yellow time expired), `z1` is 'Переключение на желтый' (Switch to yellow), and `z2` is 'Переключение на зеленый' (Switch to green).
- Output Window:** Shows the build process output, including messages like 'Build started: Project: AutomatonTest', 'Translation started', and 'Deleting old generated files...'. This window is also circled in red.

РЕЗУЛЬТАТЫ

- ▶ Разработана версия программы для трансляции описаний автоматов, представленных в формате Visio, в код на C
 - ▶ Разработано сохранение исходных данных в файл XML
 - ▶ Разработаны средства диагностики ошибок в исходных данных
 - ▶ Разработаны средства интеграции в автоматические процессы сборки приложений (make-файлы, MS Visual Studio build system)
- 