

Санкт-Петербургский государственный университет информационных
технологий, механики и оптики

Кафедра «Компьютерные технологии»

М.А. Лукин, А.В. Яковлев, А.А. Шалыто

Реализация классической игры *Ним* на основе автоматного
подхода

Программирование с явным выделением состояний

Проектная документация

Проект создан в рамках «Движения за открытую проектную документацию»
<http://is.ifmo.ru/>

Санкт-Петербург

2006

Оглавление

Введение	3
1. Постановка задачи.....	3
1.1. Правила игры «Ним»	4
2. Автоматная реализация	5
2.1. Автоматы.....	5
2.2. Источники событий.....	7
2.3. Объекты управления	7
3. Реализация программы	8
3.1. Интерпретационный подход	8
3.2. Компилятивный подход	8
Источники	9

Введение

В проекте реализована классическая игра *Ним* на основе автоматного подхода. Эта популярная игра появилась в Китае. Название ей дал Чарльз Бутон, который разработал современную математическую теорию [1,2].

Проект реализован на языке *Java* в среде *Eclipse* [3] с использованием инструментального средства *Unimod* [4]. В проекте используются три автомата.

1. Постановка задачи

Полный анализ игры впервые опубликовал в 1901 г. профессор математики Гарвардского университета Чарльз Л. Бутон [1,2]. Открытая им оптимальная стратегия основана на двоичной системе счисления.

Каждую комбинацию фишек (камней) он назвал либо опасной, либо безопасной. Если позиция, создаваемая после очередного хода игрока, гарантирует ему победу, то она называется безопасной. В противном случае, позиция опасная.

Ч. Бутон строго доказал, что любую опасную позицию всегда можно превратить в безопасную с помощью соответствующего хода. С другой стороны, если перед очередным ходом игрока уже сложилась безопасная позиция, то любой его ход превращает позицию в опасную. Таким образом, оптимальная стратегия игрока состоит в том, чтобы каждым ходом опасную позицию превращать в безопасную и заставлять соперника превращать позицию в опасную. Использование оптимальной стратегии гарантирует победу игроку тогда и только тогда, когда он ходит первым и начальная позиция фишек опасна, или он когда он ходит вторым, а начальная позиция безопасна.

Для того, чтобы определить, опасна ли позиция, или она безопасна, требуется количество фишек в каждом ряду записать в двоичной системе счисления и вычислить сумму чисел в каждом столбце (разряде). Если эта сумма четна, то позиция безопасна. Если сумма хотя бы в одном разряде нечетна, то позиция опасна. Эквивалентный, но более простой способ оценки позиции состоит в том, чтобы представить количество фишек в каждой кучке (в каждом ряду) в виде суммы степеней двойки, а затем вычеркнуть все пары одинаковых степеней и просуммировать оставшиеся степени. В результате получается так называемая «ним-сумма» для данной позиции. Иногда это число называют также «числом Ганди», или «числом Спрэга-Ганди» в честь Р. Спрэга и П. Ганди, которые независимо друг от друга разработали общую теорию такого рода игр, основанную на численных оценках каждой игровой позиции.

Предположим, например, что в начале игры имеются три кучки – из трех, пяти и семи фишек. Запишем эти числа в следующем виде.

$$3 = 2 + 1;$$

$$5 = 4 + 1;$$

$$7 = 4 + 2 + 1.$$

Вычеркнем, как показано выше, соответствующие пары четверок, двоек и единиц. Сумма того, что осталось равна единице – это и есть ним-сумма для данной позиции. Позиция безопасна в том и только в том случае, если ним-сумма для нее равна нулю. В противном случае позиция оказывается опасной (как в рассмотренном

примере). Для того, чтобы обеспечить свой выигрыш, имея перед собой опасную позицию, игроку следует превратить ее в безопасную. В данном случае если взять одну фишку из любой кучки, то ним-сумма позиции уменьшится до нуля. Как и во всех играх такого рода, в игре *Ним* можно играть «наоборот», когда тот игрок который забирает последнюю фишку считается проигравшим. Для многих игр со взятием фишек стратегия игры «наоборот» чрезвычайно сложна, однако для игры «Ним» в этом случае требуется ввести в стратегию лишь достаточно тривиальные изменения, касающиеся только конца партии. В самом деле, для того, чтобы выиграть требуется просто придерживаться обычной стратегии, причем таким образом, чтобы оставить нечетное число кучек, состоящих из одной фишки.

1.1. Правила игры Ним

Ним — игра для двух игроков, каждый из которых по очереди делает ход. Перед игроками располагается поле с камнями. Известны различные варианты игры *Ним* [5, 6]. В данном проекте правила игры таковы:

- фишки раскладываются в несколько рядов;
- игроки по очереди забирают камни из любого ряда;
- не разрешается за один ход брать камни из нескольких рядов;
- за один ход игрок должен взять хотя бы один камень;
- выигрывает тот, кто возьмет последний камень.

В наиболее известном варианте игры 12 фишек раскладываются так, как показано на рис. 1.

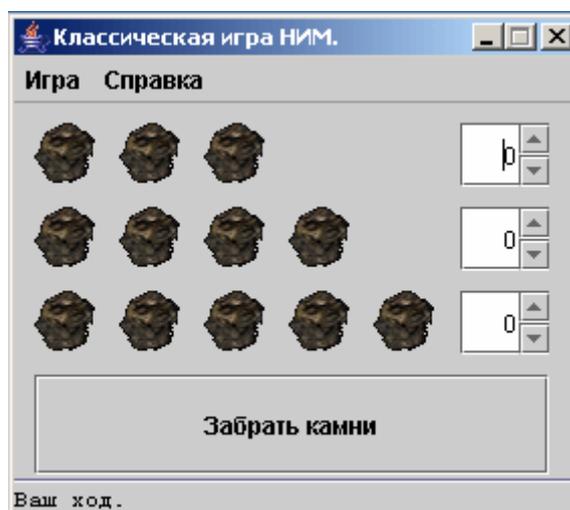


Рис. 1. Традиционная раскладка

В средней части окна располагаются оставшиеся камни. Справа от каждого ряда камней находится поле ввода количества камней, которое игрок хочет забрать. Каждое из полей ввода работает в двух режимах: можно ввести число непосредственно, либо воспользоваться кнопками «Взять на один больше» (стрелка вверх) и «Взять на один меньше» (стрелка вниз).

Под камнями располагается кнопка «Забрать камни». Под кнопкой «Забрать камни» находится поле информации.

Кроме традиционной раскладки имеется до фига разных (рис. 2.).

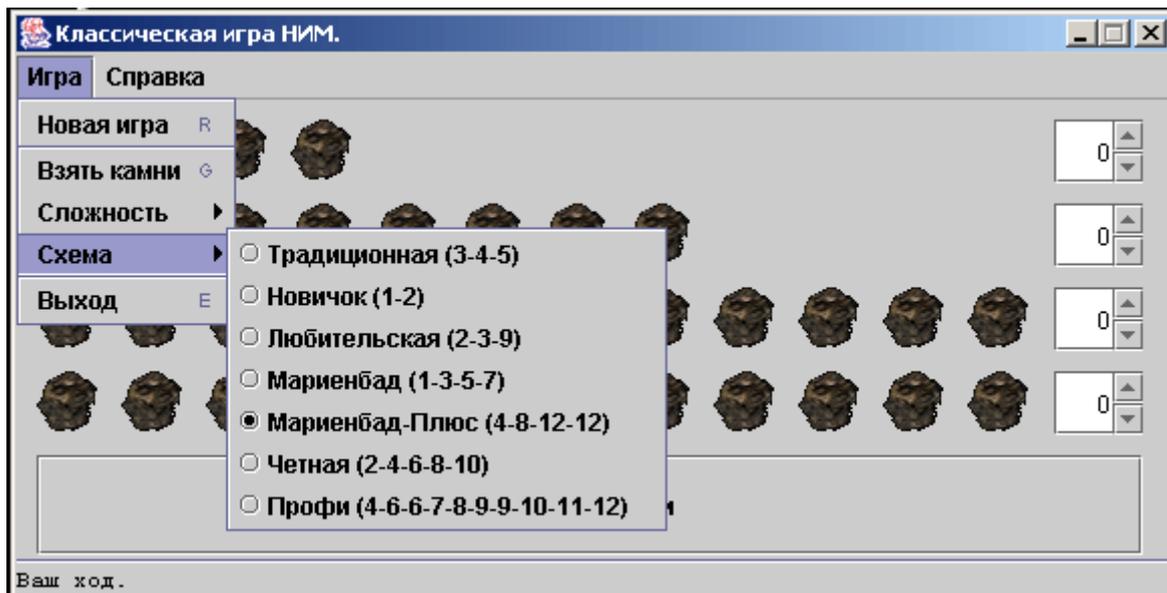


Рис. 2. Виды раскладок

2. Автоматная реализация

Схема связей системы представлена на рис.3.

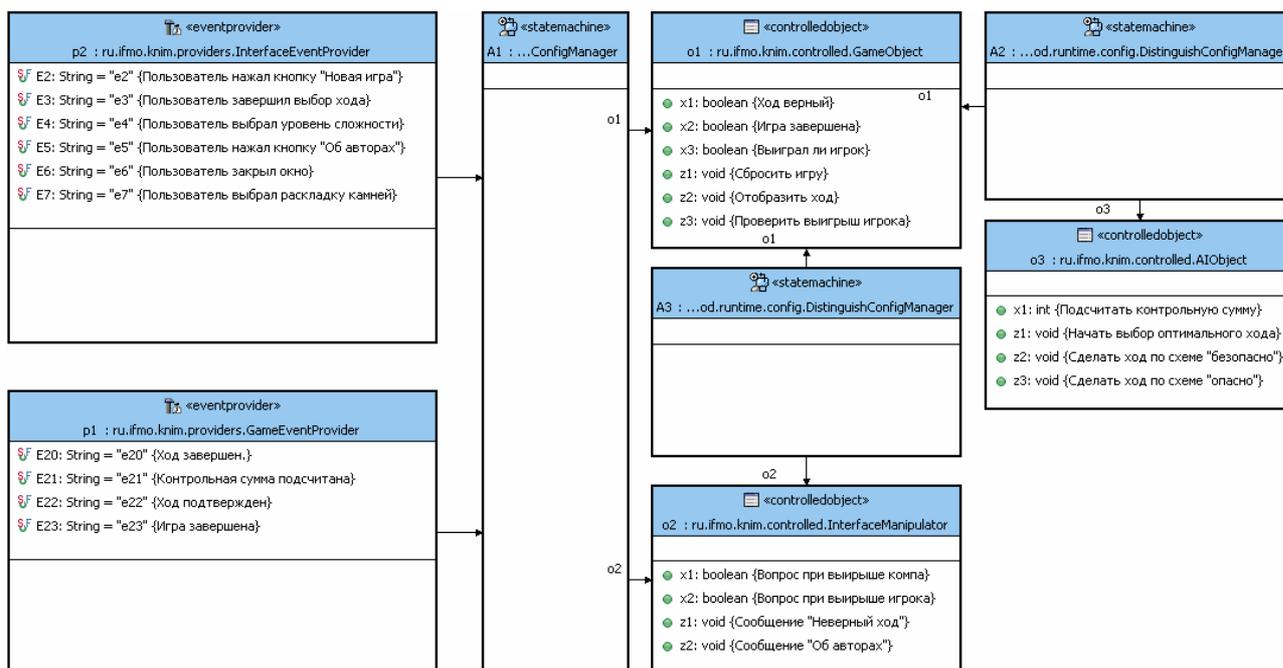


Рис. 3. Схема связей

2.1. Автоматы

Использование вложенных автоматов оправдано тем, что это сильно упрощает систему. Автомат *A1* — управляющий автомат игры. На рис.4 изображена его схема.

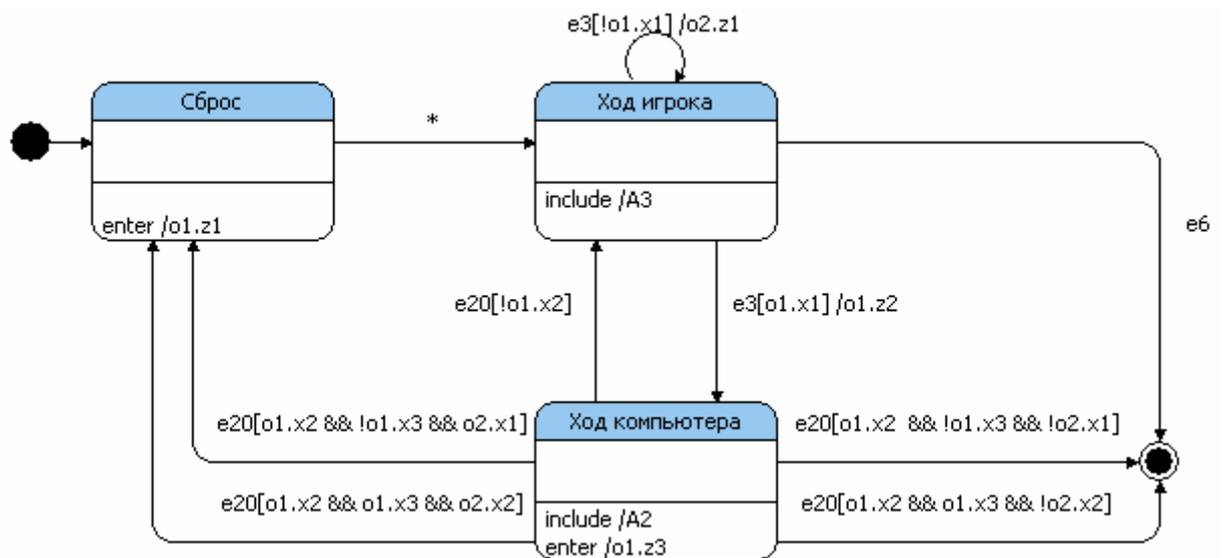


Рис. 4. Автомат $A1$

Автомат $A2$ — автомат искусственного интеллекта. Схема автомата изображена на рис.5.

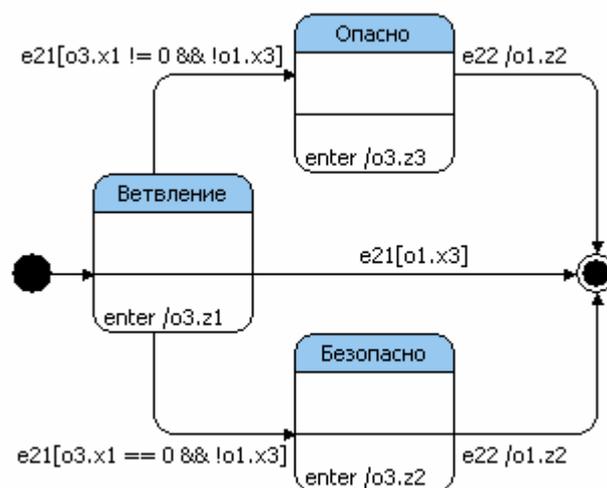


Рис. 5. Автомат $A2$

Автомат $A3$ осуществляет управление интерфейсной частью игры.

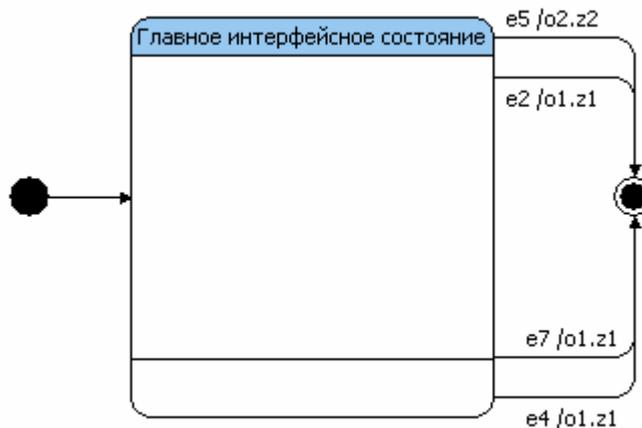


Рис. 6. Автомат *A3*

2.2. Источники событий

Опишем физический смысл источников событий данной системы.

Источник событий *p1*

Этот объект описывает события, поступающие от объекта, управляющего игрой. События:

- e20** – ход завершен;
- e21** – контрольная сумма подсчитана;
- e22** – ход подтвержден.
- e23** – игра завершена.

Источник событий *p2*

Этот объект соответствует интерфейсу. Он может формировать события:

- e2** – пользователь нажал кнопку «Новая игра»;
- e3** – пользователь завершил выбор хода;
- e4** – пользователь выбрал уровень сложности;
- e5** – пользователь нажал кнопку «Об авторах»;
- e6** – пользователь закрыл окно;
- e7** – пользователь выбрал раскладку камней.

2.3. Объекты управления

Объект управления *o1*

Объект используется для управления игрой:

- z1** – сбросить игру;
- z2** – отобразить ход;
- z3** – выигрыш игрока;
- x1** – ход верный?
- x2** – игра завершена?
- x3** – выиграл ли игрок?

Объект управления o2

Объект используется для управления интерфейсом:

z1 – сообщение «неверный ход»;

z2 – сообщение «об авторах»;

x1 – будет ли продолжена игра после проигрыша игрока?

x2 – будет ли продолжена игра после выигрыша игрока?

Объект управления o3

Объект искусственного интеллекта:

z1 – начать выбор оптимального хода;

z2 – сделать ход по схеме «безопасно»;

z3 – сделать ход по схеме «опасно»;

x1 – подсчитывается контрольная сумма и в дальнейшем сравнивается с нулем для определения позиции;

3. Реализация программы

Программа реализована с помощью инструментального средства *Unimod*. Это средство позволяет построить схему связей и автоматы (рис.2 – 4), проверив их свойства. Далее вручную реализуются объекты управления, источники событий и интерфейс программы на языке *Java*.

Существуют два подхода для создания приложения: *интерпретационный* и *компилятивный*.

3.1. Интерпретационный подход

В этом подходе используется большое число библиотек *Unimod*.

Построенные автоматы сохраняются в формате *XML*. Затем компилируются классы, реализующие работу приложения. Теперь можно запускать приложение, используя набор библиотек *Unimod*, автоматы и указанные классы.

Данный подход невыгоден тем, что требует подключения большого числа библиотек, которые для работы данного конкретного приложения не являются необходимыми.

3.2. Компилятивный подход

В данном подходе с помощью средств *Unimod* из *XML*-файла, описывающего диаграммы, генерируется *Java*-файл. В проекте используется система сборки *Ant* [7], для которой создан сборочный файл *build.xml*, описанный в приложении 2. В процессе сборки проекта в лог записывается отладочная информация. В случае успешной компиляции, в нем окажется следующее:

```
Buildfile: build.xml
```

```
clean:
```

```
[delete] Deleting directory F:\java\knim-game\build
```

```
[delete] Deleting directory F:\java\knim-game\assemble

BUILD SUCCESSFUL
Total time: 0 seconds
Buildfile: build.xml

clean:

compile:
  [echo] Compiling sources.
  [mkdir] Created dir: F:\java\knim-game\build\classes
  [echo]      Compiling      with      debug      [${compile.debug}]
[${compile.debug.level}]
  [javac] Compiling 13 source files to F:\java\knim-game\build\classes

package:
  [mkdir] Created dir: F:\java\knim-game\assemble
  [copy] Copying 1 file to F:\java\knim-game\assemble
  [copy] Copying 8 files to F:\java\knim-game\assemble

package.bin:
  [echo] Assembling module.
  [jar] Building jar: F:\java\knim-game\assemble\classical-nim.jar

build:

BUILD SUCCESSFUL
Total time: 1 second
```

В заключение отметим, что приложение 1 содержит исходные коды программы, приложение 2 – XML-описание, а приложение 3 – логи работы программы в режиме интерпретации.

Источники

1. *Гарднер М.* Крестики-нолики. М.: Мир, 1988.
2. *Фаронов В.В.* Турбо Паскаль 7.0. Начальный курс. Учебное пособие. Москва: Нолидж, 1998.
3. *Инструментальное средство Unimod.* <http://unimod.sourceforge.net/intro.html>
4. *Среда разработки Eclipse.* <http://www.eclipse.org/>
5. <http://golovolomka.hobby.ru/otvet/games10.htm>
6. <http://www.schoolnet.by/~stankovo/polza/game1.html>
7. <http://www.jakarta.org/ant>

Приложение 1. Исходные коды программы

htmlExport\ru\ifmo\knim\screens\GameScreen.java.html

Приложение 2. XML-описание

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE model PUBLIC "-//evelopers Corp.//DTD State machine model V1.0//EN"
"http://www.evelopers.com/dtd/unimod/statemachine.dtd">
<model name="Modell">
  <controlledObject name="o1" class="ru.ifmo.knim.controlled.GameObject"/>
  <controlledObject name="o2"
class="ru.ifmo.knim.controlled.InterfaceManipulator"/>
  <controlledObject name="o3" class="ru.ifmo.knim.controlled.AIObject"/>
  <eventProvider name="p2"
class="ru.ifmo.knim.providers.InterfaceEventProvider">
    <association targetRef="A1" clientRole="p2"/>
  </eventProvider>
  <eventProvider name="p1" class="ru.ifmo.knim.providers.GameEventProvider">
    <association targetRef="A1" clientRole="p1"/>
  </eventProvider>
<rootStateMachine>
  <stateMachineRef name="A1"/>
</rootStateMachine>
<stateMachine name="A1">
  <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
  <association targetRef="o2" clientRole="A1" supplierRole="o2"/>
  <association targetRef="o1" clientRole="A1" supplierRole="o1"/>
  <state name="Top" type="NORMAL">
    <state name="Ход игрока" type="NORMAL">
      <stateMachineRef name="A3"/>
    </state>
    <state name="Ход компьютера" type="NORMAL">
      <stateMachineRef name="A2"/>
      <outputAction ident="o1.z3"/>
    </state>
    <state name="s2" type="FINAL"/>
    <state name="s1" type="INITIAL"/>
    <state name="Сброс" type="NORMAL">
      <outputAction ident="o1.z1"/>
    </state>
  </state>
  <transition sourceRef="Ход игрока" targetRef="Ход компьютера" event="e3"
guard="o1.x1">
    <outputAction ident="o1.z2"/>
  </transition>
  <transition sourceRef="Ход игрока" targetRef="Ход игрока" event="e3"
guard="!o1.x1">
    <outputAction ident="o2.z1"/>
  </transition>
  <transition sourceRef="Ход игрока" targetRef="s2" event="e6"/>
  <transition sourceRef="Ход компьютера" targetRef="Ход игрока" event="e20"
guard="!o1.x2"/>
  <transition sourceRef="Ход компьютера" targetRef="Сброс" event="e20"
guard="o1.x2 && !o1.x3 && o2.x1"/>
  <transition sourceRef="Ход компьютера" targetRef="s2" event="e20"
guard="o1.x2 && !o1.x3 && !o2.x1"/>
  <transition sourceRef="Ход компьютера" targetRef="Сброс" event="e20"
guard="o1.x2 && o1.x3 && o2.x2"/>
  <transition sourceRef="Ход компьютера" targetRef="s2" event="e20"
guard="o1.x2 && o1.x3 && !o2.x2"/>

```

```

    <transition sourceRef="s1" targetRef="Сброс"/>
    <transition sourceRef="Сброс" targetRef="Ход игрока" event="*/>
</stateMachine>
<stateMachine name="A2">
    <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
    <association targetRef="o3" clientRole="A2" supplierRole="o3"/>
    <association targetRef="o1" clientRole="A2" supplierRole="o1"/>
    <state name="Top" type="NORMAL">
        <state name="s2" type="FINAL"/>
        <state name="Безопасно" type="NORMAL">
            <outputAction id="o3.z2"/>
        </state>
        <state name="s1" type="INITIAL"/>
        <state name="Опасно" type="NORMAL">
            <outputAction id="o3.z3"/>
        </state>
        <state name="Ветвление" type="NORMAL">
            <outputAction id="o3.z1"/>
        </state>
    </state>
    <transition sourceRef="Безопасно" targetRef="s2" event="e22">
        <outputAction id="o1.z2"/>
    </transition>
    <transition sourceRef="s1" targetRef="Ветвление"/>
    <transition sourceRef="Опасно" targetRef="s2" event="e22">
        <outputAction id="o1.z2"/>
    </transition>
    <transition sourceRef="Ветвление" targetRef="Опасно" event="e21"
guard="o3.x1 != 0 & & !o1.x3"/>
    <transition sourceRef="Ветвление" targetRef="Безопасно" event="e21"
guard="o3.x1 == 0 & & !o1.x3"/>
    <transition sourceRef="Ветвление" targetRef="s2" event="e21" guard="o1.x3">
        <outputAction id="o3.z4"/>
    </transition>
</stateMachine>
<stateMachine name="A3">
    <configStore
class="com.evelopers.unimod.runtime.config.DistinguishConfigManager"/>
    <association targetRef="o2" clientRole="A3" supplierRole="o2"/>
    <association targetRef="o1" clientRole="A3" supplierRole="o1"/>
    <state name="Top" type="NORMAL">
        <state name="s2" type="INITIAL"/>
        <state name="s1" type="FINAL"/>
        <state name="Главное интерфейсное состояние" type="NORMAL"/>
    </state>
    <transition sourceRef="s2" targetRef="Главное интерфейсное состояние"/>
    <transition sourceRef="Главное интерфейсное состояние" targetRef="s1"
event="e4">
        <outputAction id="o1.z1"/>
    </transition>
    <transition sourceRef="Главное интерфейсное состояние" targetRef="s1"
event="e7">
        <outputAction id="o1.z1"/>
    </transition>
    <transition sourceRef="Главное интерфейсное состояние" targetRef="s1"
event="e5">
        <outputAction id="o2.z2"/>
    </transition>
    <transition sourceRef="Главное интерфейсное состояние" targetRef="s1"
event="e2">
        <outputAction id="o1.z1"/>
    </transition>
</stateMachine>

```

</model>

Приложение 3. Логи работы программы в режиме интерпретации

Рассмотрим лог работы программы в случае победы компьютера в раскладке «Новичок».

```
[info] Start event [e2] processing. In state [/A1:Top]
[info] Transition to go found [s1#Сброс##true]
[info] Start on-enter action [o1.z1] execution
[info] Finish on-enter action [o1.z1] execution
[debug] Try transition [Сброс#Ход игрока##true]
[info] Transition to go found [Сброс#Ход игрока##true]
[info] Start event [e2] processing. In state [/A1:Ход игрока/A3:Top]
[info] Transition to go found [s2#Главное интерфейсное состояние##true]
[debug] Try transition [Главное интерфейсное состояние#s1#e2#true]
[info] Transition to go found [Главное интерфейсное состояние#s1#e2#true]
[info] Start output action [o1.z1] execution
[info] Finish output action [o1.z1] execution
[info] State machine came to final state [/A1:Ход игрока/A3:s1]
[info] Finish event [e2] processing. In state [/A1:Ход игрока/A3:s1]
[info] Finish event [e2] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока/A3:Top]
[info] Transition to go found [s2#Главное интерфейсное состояние##true]
[info] Finish event [e24] processing. In state [/A1:Ход игрока/A3:Главное
интерфейсное состояние]
[info] Finish event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока/A3:Главное
интерфейсное состояние]
[info] Finish event [e24] processing. In state [/A1:Ход игрока/A3:Главное
интерфейсное состояние]
[info] Finish event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e7] processing. In state [/A1:Ход игрока]
[info] Start event [e7] processing. In state [/A1:Ход игрока/A3:Главное
интерфейсное состояние]
[debug] Try transition [Главное интерфейсное состояние#s1#e7#true]
[info] Transition to go found [Главное интерфейсное состояние#s1#e7#true]
[info] Start output action [o1.z1] execution
[info] Finish output action [o1.z1] execution
[info] State machine came to final state [/A1:Ход игрока/A3:s1]
[info] Finish event [e7] processing. In state [/A1:Ход игрока/A3:s1]
[info] Finish event [e7] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e24] processing. In state [/A1:Ход игрока/A3:Top]
[info] Transition to go found [s2#Главное интерфейсное состояние##true]
[info] Finish event [e24] processing. In state [/A1:Ход игрока/A3:Главное
интерфейсное состояние]
[info] Finish event [e24] processing. In state [/A1:Ход игрока]
[info] Start event [e3] processing. In state [/A1:Ход игрока]
[debug] Try transition [Ход игрока#Ход компьютера#e3#o1.x1]
[info] Start input action [o1.x1] calculation
[info] Finish input action [o1.x1] calculation. Its value is [true]
[info] Transition to go found [Ход игрока#Ход компьютера#e3#o1.x1]
[info] Start output action [o1.z2] execution
[info] Finish output action [o1.z2] execution
[info] Start on-enter action [o1.z3] execution
[info] Finish on-enter action [o1.z3] execution
[info] Start event [e3] processing. In state [/A1:Ход компьютера/A2:Top]
[info] Transition to go found [s1#Ветвление##true]
```

```
[info] Start on-enter action [o3.z1] execution
[info] Finish on-enter action [o3.z1] execution
[info] Finish event [e3] processing. In state [/A1:Ход компьютера/A2:Ветвление]
[info] Finish event [e3] processing. In state [/A1:Ход компьютера]
[info] Start event [e21] processing. In state [/A1:Ход компьютера]
[info] Start event [e21] processing. In state [/A1:Ход компьютера/A2:Ветвление]
[debug] Try transition [Ветвление#Опасно#e21#o3.x1 != 0 && !o1.x3]
[info] Start input action [o3.x1] calculation
[info] Finish input action [o3.x1] calculation. Its value is [1]
[info] Start input action [o1.x3] calculation
[info] Finish input action [o1.x3] calculation. Its value is [false]
[info] Transition to go found [Ветвление#Опасно#e21#o3.x1 != 0 && !o1.x3]
[info] Start on-enter action [o3.z3] execution
[info] Finish on-enter action [o3.z3] execution
[info] Finish event [e21] processing. In state [/A1:Ход компьютера/A2:Опасно]
[info] Finish event [e21] processing. In state [/A1:Ход компьютера]
[info] Start event [e22] processing. In state [/A1:Ход компьютера]
[info] Start event [e22] processing. In state [/A1:Ход компьютера/A2:Опасно]
[debug] Try transition [Опасно#s2#e22#true]
[info] Transition to go found [Опасно#s2#e22#true]
[info] Start output action [o1.z2] execution
[info] Finish output action [o1.z2] execution
[info] State machine came to final state [/A1:Ход компьютера/A2:s2]
[info] Finish event [e22] processing. In state [/A1:Ход компьютера/A2:s2]
[info] Finish event [e22] processing. In state [/A1:Ход компьютера]
[info] Start event [e20] processing. In state [/A1:Ход компьютера]
[debug] Try transition [Ход компьютера#Ход игрока#e20#!o1.x2]
[info] Start input action [o1.x2] calculation
[info] Finish input action [o1.x2] calculation. Its value is [true]
[debug] Try transition [Ход компьютера#Сброс#e20#o1.x2 && !o1.x3 && o2.x1]
[info] Start input action [o1.x3] calculation
[info] Finish input action [o1.x3] calculation. Its value is [false]
[info] Start input action [o2.x1] calculation
[info] Finish input action [o2.x1] calculation. Its value is [false]
[debug] Try transition [Ход компьютера#s2#e20#o1.x2 && !o1.x3 && !o2.x1]
[info] Transition to go found [Ход компьютера#s2#e20#o1.x2 && !o1.x3
&& !o2.x1]
[info] State machine came to final state [/A1:s2]
[info] Finish event [e20] processing. In state [/A1:s2]
```

