

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Рукович Александр Владимирович

Должность: Директор

Дата подписания: 11.03.2017

Уникальный программный ключ:

f45eb7c44954caac03ea7d4f52eb8d7d6b5cb9bae6d9b4bda094afddaf1b70511

Министерство образования и науки Российской Федерации  
Технический институт (филиал) федерального государственного  
автономного образовательного учреждения высшего образования  
«Северо-Восточный федеральный университет  
имени М.К. Аммосова» в г. Нерюнгри

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
К ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ  
**«МАТЕМАТИЧЕСКОЕ И ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ»**  
ДЛЯ СТУДЕНТОВ НАПРАВЛЕНИЙ ПОДГОТОВКИ  
09.03.03 «ПРИКЛАДНАЯ ИНФОРМАТИКА»,  
01.03.02 «ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»  
(очная форма обучения)  
Часть I

Нерюнгри  
Издательство ТИ(ф) СВФУ  
2017

Утверждено учебно-методическим советом ТИ (ф) ФГАОУ ВО «СВФУ»

Составитель:

**В.В. Юданова**, старший преподаватель кафедры математики и информатики

Рецензент:

**В.М. Самохина**, к.п.н., зав. кафедрой математики и информатики

Подготовлено на кафедре математики и информатики

Печатается в авторской редакции

*Методические указания по курсу «Математическое и имитационное моделирование» (Часть I) предназначены для студентов направлений обучения 09.03.03 "Прикладная информатика" и 01.03.02 "Прикладная математика и информатика". Методические указания посвящены разделу имитационного моделирования и содержат лабораторные работы по разработке всех основных видов имитационных моделей средствами программы Anylogic.*

*Материал предназначен для студентов, преподавателей и аспирантов, интересующихся современными подходами к построению моделей сложных систем.*

© Технический институт (ф) СВФУ, 2017

## ОГЛАВЛЕНИЕ

Введение	4
Лабораторная работа 1. Моделирование динамических систем	5
Практические задания к лабораторной работе 1	7
Лабораторная работа 2. Дискретно-событийная модель	7
Практические задания к лабораторной работе 2	11
Лабораторная работа 3. Использование анимации при моделировании в программе Anylogic	12
Лабораторная работа 4. Использование стейтчарта и диаграммы процесса при моделировании в программе Anylogic	14
Практические задания к лабораторным работам 3, 4	18
Лабораторная работа 5. Системно-динамическая модель	20
Практические задания к лабораторной работе 5	22
Лабораторная работа 6. Агентная модель	23
Практические задания к лабораторной работе 6	26
Список литературы	27

## **ВВЕДЕНИЕ**

Программа имитационного моделирования Anylogic является одним из наиболее востребованных инструментов разработки имитационных моделей. Использование данного программного продукта при изучении раздела имитационного моделирования позволит ознакомить студентов с возможностью разработки моделей динамической системы и системной динамики, дискретно-событийным и агентным моделированием.

В содержательной части методических указаний приводится подробное описание по разработке того или иного вида модели на примере решения различных типов задач. Также по каждой теме имеются практические задания по вариантам для самостоятельного выполнения на лабораторных работах по дисциплине «Математическое и имитационное моделирование».

Методические указания предназначены для студентов направлений обучения 09.03.03 «Прикладная информатика» и 01.03.02 «Прикладная математика и информатика» и также могут быть использованы преподавателями ВУЗов, аспирантами, которые интересуются современными подходами к построению моделей сложных систем.

# Лабораторная работа 1

## Тема: Моделирование динамических систем

**Цель:** изучить возможности моделирования динамических систем в программе Anylogic.

**Пример:** Построить модель колебаний маятника Фуко. Динамическая система с непрерывным временем, называемая маятником Фуко, записывается в виде следующих дифференциальных уравнений:

$$\frac{\partial v_x}{\partial t} = 2v_y\omega + \omega^2 x - g \frac{x}{L}$$

$$\frac{\partial v_y}{\partial t} = -2v_x\omega + \omega^2 y - g \frac{y}{L}$$

$$\frac{\partial x}{\partial t} = v_x$$

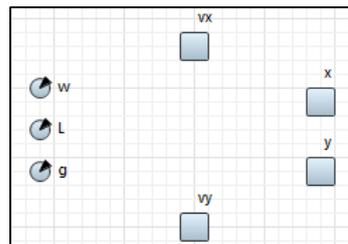
$$\frac{\partial y}{\partial t} = v_y$$

где:  $L$  – длина подвеса;  $g$  – гравитационная постоянная;  $\omega$  – относительная частота;  $t$  – время;  $x, y$  – координаты конца маятника;  $v_y, v_x$  – скорость изменения координат маятника.

### Ход работы

#### Построение модели

1. Создать новую модель: Файл -> Создать -> Модель.
2. Разместить в графическом поле объекта Main данные модели элементы - «Накопитель» (Палитра -> Системная динамика) и «Параметр» (Палитра -> Агент), как показано на рисунке:



3. Установить значения свойств для параметров (панель Свойства):

Параметр	Начальное значение/Значение по умолчанию	Тип
w	0.05	double
L	10	double
g	7	double

4. Установить для накопителей следующие свойства (панель Свойства):

Накопитель	Начальное значение/Значение по умолчанию	Тип
x	1	double
y	1	double
vx	0	double

vy	0	double
----	---	--------

и ввести дифференциальные уравнения описывающие колебания маятника Фуко (панель Свойства).

5. Настроить связи между параметрами и накопителями согласно заданным дифференциальным уравнениям с помощью инструмента «Связь» (Палитра -> Системная динамика).

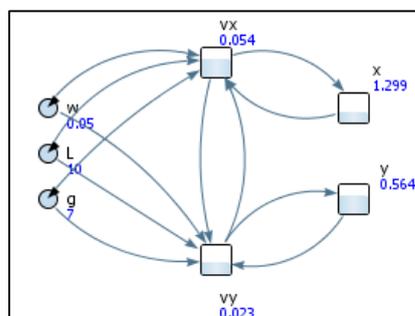
### Запуск модели

6. Настроить эксперимент (панель Свойства для объекта Simulation):

Свойство	Значение
Реальное время со скоростью	1
Остановить	Нет

7. Выполнить компиляцию модели (Модель -> Построить) и запустить (Модель -> Запустить).

8. Проверить работу модели, фрагмент работы которой показан на рисунке.



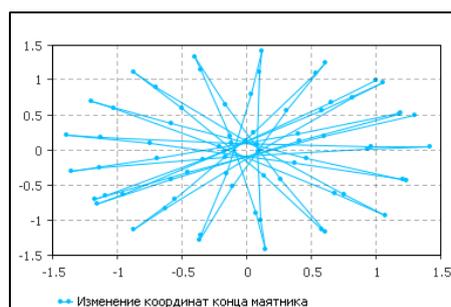
### Размещение графика

9. Разместить график в графическом поле объекта Main (палитра Свойства -> График) и настроить его свойства (в разделе «Данные»):

Свойство	Значение
Заголовок	Изменение координат конца маятника
Значение по оси X	x
Значение по оси Y	y

10. Выполнить компиляцию модели (Модель -> Построить) и запустить (Модель -> Запустить).

11. Проверить построение графика, отображающего изменение координат конца маятника.



## Практические задания к лабораторной работе 1

1. Выполнить исследование поведения модели динамической системы, описывающей колебания маятника Фуко для двух вариантов значений исходных данных. Сравнить результаты и объяснить изменения поведения маятника.
2. Подготовить ответы на контрольные вопросы.
3. Составить отчет по лабораторной работе.

### Варианты заданий

№ варианта	L	g	w
1	10	7	0.05
	20	7	0.1
2	30	7	0.15
	40	7	0.20
3	50	7	0.25
	10	8	0.05
4	20	8	0.1
	30	8	0.15
5	40	8	0.20
	50	8	0.25
6	10	9	0.05
	20	9	0.1
7	30	9	0.15
	40	9	0.20
8	50	9	0.25
	10	10	0.05
9	20	10	0.1
	30	10	0.15
10	40	10	0.2
	50	10	0.25

### Контрольные вопросы

1. Приведите примеры моделей динамических систем.
2. Что означает элемент параметр в программе Anylogic и для чего используется?
3. Что означает элемент накопитель в программе Anylogic и для чего используется?

## Лабораторная работа 2

### Тема: Дискретно-событийная модель

**Цель:** изучить возможности моделирования дискретно-событийных систем в Anylogic.

**Пример:** Построить модель для исследования процесса незатухающих гармонических колебаний. Колебательный процесс описывается уравнением гармонических колебаний следующего вида:

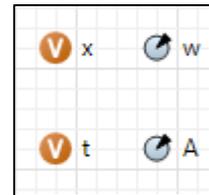
$$x[n] = A \sin \varphi$$

$$\varphi = \omega t[n] + \varphi_0$$

где:  $\varphi$  – фаза колебаний;  $\varphi_0$  – начальная фаза;  $\omega$  – частота колебаний;  $t[n]$  – дискретное время;  $A$  – амплитуда колебаний;  $x$  – текущее значение величины колебательного процесса.

### Ход работы

1. Создать новую модель: Файл -> Создать -> Модель.
2. Разместить в графическом поле объекта Main данные модели - элементы «Переменная» и «Параметр», как показано на рисунке (Палитра -> Агент). Будем считать, что начальная фаза колебаний  $\varphi_0$  равна нулю.
3. Установить значения следующих свойств для переменных и параметров (панель Свойства):



Параметр/переменная	Начальное значение/Значение по умолчанию	Тип
A	1	double
$\omega$	0.25	double
x	0	double
t	0	double

4. В графическом поле класса Main разместить элемент «Событие».
5. Настроить свойства элемента «Событие»:

Свойство	Значение
Имя	event
Тип события	По таймауту
Режим	Циклический
Время первого срабатывания	time()
Период	second()
Действие	getX()

Функция *time()* возвращает текущее модельное время, а функция *second()* возвращает значение модельного времени в секундах.

Свойство «Действие» используется для задания программного кода, который будет выполняться при выполнении события.

6. Написать код функции *getX()*, обращение к которой выполняется при выполнении события (панель Свойства для объекта Main в разделе «Дополнительный код»):

```
public void getX() {
    t=time();
    x=A*Math.sin(w*t);
}
```

Math – статический класс математических функций.

### Запуск модели

7. Настроить эксперимент (панель Свойства для объекта Simulation):
8. Выполнить компиляцию модели (Модель -> Построить) и запустить (Модель -> Запустить).
9. Проверить работу модели, фрагмент работы которой показан на рисунке.



### Размещение графика

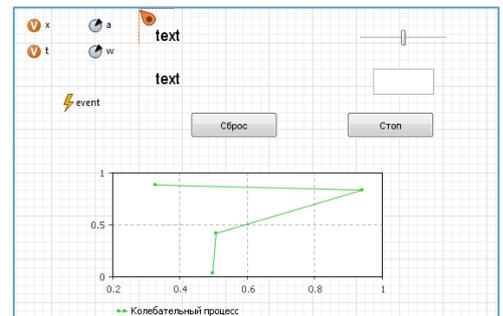
10. Разместить график в графическом поле объекта Main (палитра Свойства -> Временной график) и настроить его свойства (в разделе «Данные»):

Свойство	Значение
Заголовок	Колебательный процесс
Значение по оси X	t
Значение по оси Y	x
Отображать до	500
Период	0.25

11. Выполнить компиляцию модели (Модель -> Построить) и запустить (Модель -> Запустить).

### Настройка презентации модели

12. Убрать отображение переменных ( $x$  и  $t$ ) и параметров ( $w$  и  $A$ ) в презентации (в панели Свойства установить Видимость - Нет);
13. Разместить в графическом поле объекта Main элементы управления «Текстовое поле» для изменения амплитуды ( $A$ ) и «Бегунок» - для частоты ( $w$ ), два элемента «Текст» для отображения комментариев и два элемента «Кнопка». Настроить следующие свойства элементов.



Элемент	Свойство	Значение
Текстовое поле	Имя	editboxA
	Связать с	активен, A
	Минимальное значение	1
Бегунок	Имя	sliderF
	Связать с	Активен, w
	Минимальное значение	0,25
	Максимальное значение	1
Текст	Имя	textF
	Шрифт	18 пунктов
Текст	Имя	textA
	Шрифт	18 пунктов
Кнопка	Метка	Сброс
Кнопка	Метка	Стоп

14. Для элемента «Событие» (event) добавить код (в панели Свойства раздел «Действие»):

```
getX();
String bufer;
bufer="Угловая частота 1/c=";
textF.setText(bufer+w);
bufer="Амплитуда колебаний M=";
textA.setText(bufer+A);
```

15. Для кнопки «Сброс» добавить код:

```
w=0.25;
A=1;
sliderF.setValue(w);
editboxA.setText(""+A);
```

16. Кнопка «Стоп» должна работать как переключатель: первое нажатие - остановка генерации события модели, повторное нажатие запуск генерации. Процесс повторяется циклически. Надпись на кнопке менять по смыслу ее использования: «Стоп» или «Пуск». Добавить соответствующий программный код для кнопки «Стоп» с использованием следующих методов.

Тип метода	Название метода	Описание
String	getText()	возвращает текст, отображаемый в текстовом поле или метке в текущий момент времени
boolean	pauseSimulation()	приостанавливает запущенную модель
boolean	runSimulation()	запускает приостановленную модель

17. Выполнить компиляцию модели (Модель -> Построить) и запустить (Модель -> Запустить).

#### *Добавление областей просмотра*

18. Разместить в графическом поле объекта Main элемент «Область просмотра» (вкладка Презентация):

Свойство	Значение
Имя	viewAreaSinus
Заголовок	Незатухающие колебания

19. Выделить график и «вырезать» его в буфер обмена. Используя кнопку панели инструментов (  Области просмотра), выбрать область просмотра «Незатухающие колебания» и поместить в нее график из буфера обмена.

20. Разместить в графическом поле объекта Main еще один элемент «Область просмотра» (вкладка Презентация):

Свойство	Значение
Имя	viewAreaMain
Заголовок	Модель колебаний

21. Вырезать в буфер обмена все элементы модели кроме графика, перейти в область просмотра и вставить из буфера выделенные элементы.

22. Выполнить запуск модели, в окне выполнения модели для перехода в области просмотра используется кнопка панели  - Показать области.
23. Разместить командные кнопки для переключения между областями просмотра. Кнопка «График» - для просмотра графика, кнопка «Назад» - для возврата ко всей презентации.
24. Для каждой кнопки определить области просмотра (добавить программный код в панели Свойства раздел «Действие»), используя метод `ИмяОбластиПросмотра.navigateTo()`.

### Практические задания к лабораторной работе 2

1. Построить модель затухающих гармонических колебаний, для которых колебательный процесс описывается уравнениями вида:

$$y = Ae^{-\delta \cdot t[n]} \sin \varphi$$

$$\varphi = \omega_{\text{зат}} t [n] + \varphi_0$$

$$\omega_{\text{зат}} = \sqrt{\omega_0^2 - \delta^2}$$

где  $e$  – основание натурального логарифма,  $\delta$  – коэффициент затухания. Дискретное время вычислить по формуле:  $t[n] = T_0 \cdot n$ , где  $n = 0, \dots, \infty$ ,  $T_0$  – период дискретизации модели.

Для работы с моделью добавить график, области просмотра и настроить презентацию модели.

2. Подготовить ответы на контрольные вопросы.
3. Составить отчет по лабораторной работе.

#### Варианты заданий

№ варианта	$T_0$	$a$	$\omega$	$\varphi_0$	$\delta$
1	0.001	1	0.2	0	0.01
2	0.001	0.9	0.25	0.05	0.02
3	0.001	0.8	0.3	0.1	0.03
4	0.001	0.7	0.35	0.15	0.04
5	0.001	0.6	0.4	0.2	0.05
6	0.01	1	0.2	0	0.06
7	0.01	0.9	0.25	0.05	0.07
8	0.01	0.8	0.3	0.1	0.08
9	0.01	0.7	0.35	0.15	0.09
10	0.01	0.6	0.4	0.2	0.1

#### Контрольные вопросы

1. Приведите примеры дискретно-событийных моделей.
2. Что означает элемент переменная в программе Anylogic и для чего используется?
3. Для чего используются области просмотра в программе Anylogic?

## Лабораторная работа 3

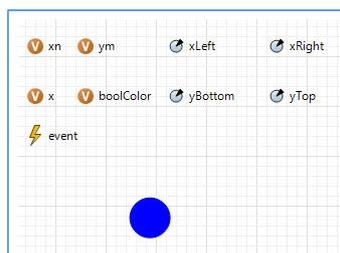
### Тема: Использование анимации при моделировании в программе Anylogic

**Цель:** изучить возможности создания анимированных фигур при моделировании в программе Anylogic.

**Пример:** Разработать модель, которая демонстрирует движение сферического объекта синего цвета по заданной траектории. Траектория объекта изменяется по синусоидальному закону. Когда траектория объекта приближается к значению амплитуды траектории с погрешностью равной 0,01 окрасить его в красный цвет, при удалении восстановить синий цвет. Параметры траектории: амплитуда = 3 метра, угловая частота = 0.5 рад/с, модельное время = 500с.

#### Ход работы

1. Создать новую модель: Файл -> Создать -> Модель.
2. Используя контекстное меню узла модели, создать активный класс Ball (Создать -> Тип агента -> Ball).
3. Разместить в графическом поле объекта Ball следующие элементы, как показано на рисунке.



4. Настроить свойства всех размещенных элементов.

Элемент	Характеристика элемента	Тип характеристики	Свойство
xn	физическая координата, положение шара по оси X	переменная	
x	расчетное значение траектории	переменная	
ym	физическая координата, положение шара по оси Y	переменная	
boolColor	логическая переменная	переменная	Начальное значение false
xLeft	левая координата физической области вывода по оси X	параметр	Значение по умолчанию = 0
yBottom	левая координата физической области вывода по оси	параметр	Значение по умолчанию = -3
xRight	правая координата физической области вывода по оси X	параметр	Значение по умолчанию = 500

уTop	правая координата физической области вывода по оси Y	параметр	Значение по умолчанию = +3
------	--	----------	----------------------------

Для размещения закрашенного круга использовать палитру «Презентация» и элемент «Овал».

Элемент	Тип характеристики	Свойство
oval	овал	Цвет заливки – blue, Цвет линии - blue
event	событие	Тип события: По таймауту, Режим: Циклический Время первого срабатывания: time() Период: second()/1000

5. Добавить программный код для элемента event (раздел Действие:), чтобы определить движение круга по синусоиде и значение boolColor – сигнала для смены цвета круга:

```
double A=3; //амплитуда
double w=0.5; //частота рад/с
x=A*Math.sin(w*time());
getX1();getY1();
//погрешности и изменение цвета
double eps;
eps = A-Math.abs(x);
if(eps<0.01){boolColor = true;} else
{boolColor=false;}
```

6. Изменить свойства круга, чтобы он двигался и менял цвет.

Свойство	Значение
X	xп
Y	ym
Цвет заливки	boolColor ? new Color(255,0,0): new Color(0,0,255)

7. Разместить в графическом поле объекта **Main** элемент «Прямоугольник» (вкладка Презентация) и настроить его свойства:

Свойство	Значение
Имя	rectView
Цвет заливки	silver
Цвет линии	black
X	60
Y	80
Ширина	690
Высота	260

8. Добавить программный код для пересчета расчетных координат в координаты физической области вывода - прямоугольник. Программный код добавить в код настройки активного класса Ball:

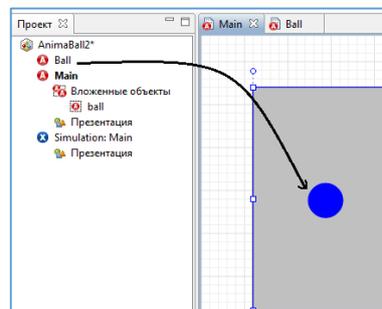
```
public void getX1()
{
//вычисление максимальных значений по оси X
double nLeft,nRight;
nLeft=get_Main().rectView.getX();
nRight=get_Main().rectView.getX()+get_Main().rectView.getWidth();
//вычисление физической координаты по оси X
xn=(time()-xLeft)/(xRight-xLeft)*(nRight-nLeft)+nLeft;
}
```

```

public void getY1()
{
//вычисление максимальных значений по оси Y
double mBottom,mTop;
mBottom=get_Main().rectView.getY()+get_Main().rectView.getHeight();
mTop=get_Main().rectView.getY();
//вычисление физической координаты по оси Y
ym=(x-yBottom)/(yTop-yBottom)*(mTop-mBottom)+mBottom;}

```

9. Сделать объект Ball вложенным в объекте Main: для этого нужно открыть дерево проекта, затем выбрать объект Main двойным щелчком мыши и используя технологию Drag & Drop, перетащить объект Ball в поле объекта Main, так как это показано на рисунке.



10. Настроить параметры эксперимента Simulation:

Свойство	Значение
Остановить	В заданное время
Конечное время	500

## Лабораторная работа 4

### Тема: Использование стейтчарта и диаграммы процесса при моделировании в программе Anylogic

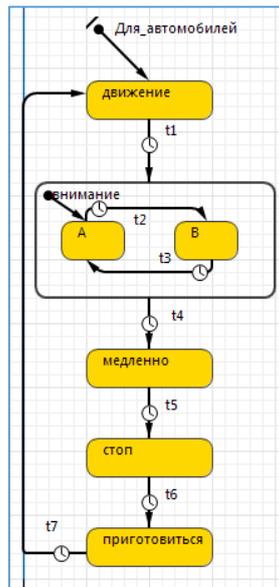
**Цель:** изучить возможности использования стейтчартов и процессной диаграммы при моделировании в программе Anylogic.

**Пример:** Разработать модель, регулируемого пешеходного перехода с двумя светофорами. Светофор, разрешающий или запрещающий движение автотранспорта, и светофор, разрешающий или запрещающий движение пешеходов. Создать графическую анимацию работы светофоров и движения автомобилей и пешеходов.

#### Ход работы

#### Создание стейтчарта, моделирующего состояния светофора для автотранспорта

1. Создать новую модель: Файл -> Создать -> Модель.
2. Разместить в графическом поле объекта Main следующую диаграмму состояний и настроить свойства ее элементов.



Элемент диаграммы	Характеристика элемента
<b>Блоки состояний</b>	
движение	движение автомобилям разрешено (зеленый свет)
внимание	внимание (мигающий зеленый): включает в себя пару переключающихся элементарных состояний: в одном из них зеленый горит (состояние А), в другом – нет (со-состояние В)
медленно	приготовиться к остановке (желтый свет)
стоп	запрет движения (красный свет)
приготовиться	приготовиться к движению (красный и желтый свет горят одновременно)
<b>Стрелки перехода</b>	
переход: имя: t1	в состоянии движение светофор находится 10 с.
переход: имя: t2	между состояниями А и В переходы срабатывают через 1 с.
переход: имя: t3	
переход: имя: t4	7 с. зеленый сигнал мигает
переход: имя: t5	в состоянии медленно 4 с. горит желтый
переход: имя: t6	в течение 10 с. движение запрещено
переход: имя: t7	в течение 10 с. движение запрещено

Для построения гиперсостояния в блоке «внимание» сначала создайте обычное состояние, увеличьте его (растянув мышью) и поместите внутрь два других состояний.

3. Запустить модель и проверить работу стейтчарта.

#### *Создание светофора для автотранспорта*

4. Разместить в графическом поле объекта Main следующие параметры: «красный», «желтый», «зеленый» - тип: boolean.

5. Настроить свойства блоков состояний диаграммы, чтобы менялись соответствующие значения параметров:

Например:

Блок состояния	Свойство
движение	Действие при входе: зеленый = true Действие при выходе: зеленый = false

Для гиперсостояния «внимание» у состояния А эти поля нужно оставить пустыми – когда светофор находится в этом состоянии, он не горит.

6. Запустить модель и проверить работу параметров.

### Создание презентации и ее анимация

7. Используя элементы палитры Презентация: прямоугольник и три овала постройте в графическом поле объекта Main изображение светофора.

8. Установите динамическое значение цвета верхнего сигнала светофора: если параметр красный истинна, то цвет должен быть red (красный), в противном случае его цвет нужно установить gray (серый) (Свойство: Цвет заливки: красный? red: gray). Аналогично для остальных сигналов (овалов) светофора.

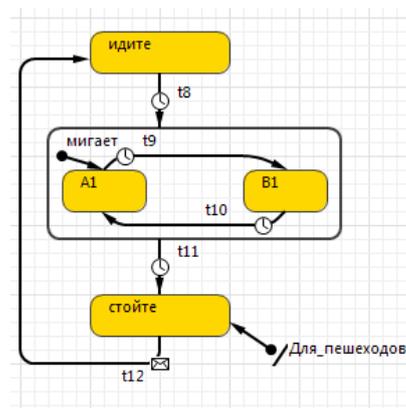
9. Запустить модель и проверить работу светофора для автомобилей.

### Создание стейтчарта, моделирующего состояния светофора для пешеходов

10. Разместить в графическом поле объекта Main следующую диаграмму состояний.

11. Синхронизировать стейтчарты, посылая специальные разрешающие сигналы из одного стейтчарта в другой:

а. в стейтчарте Для\_пешеходов переход t12 будет срабатывать когда получен сигнал ПЕШЕХОДЫ, который будет генерироваться в стейтчарте Для\_автомобилей при переходе t5 в состояние «стоп»:



Стрелка перехода	Свойство
<b>Стейтчарт «Для_автомобилей»</b>	
t5	Действие: Для_пешеходов.fireEvent("ПЕШЕХОДЫ")
<b>Стейтчарт «Для_пешеходов»</b>	
t12	Происходит: при получении сообщения Тип сообщения: String Осуществить переход: при получении заданного сообщения Сообщение: "ПЕШЕХОДЫ"

метод fireEvent( ) – вызывается для передачи сообщения в диаграмму состояния.

б. в стейтчарте Для\_автомобилей переход t6 будет срабатывать когда получен сигнал АВТОМОБИЛИ, который генерируется в стейтчарте Для\_пешеходов при переходе t11 в состояние «стойте»: настроить свойства указанных переходов аналогично.

12. Запустить модель и проверить работу стейтчарта.

### Создание светофора для пешеходов

13. Добавить в модель два параметра: «стоять» и «идти» - тип: boolean.

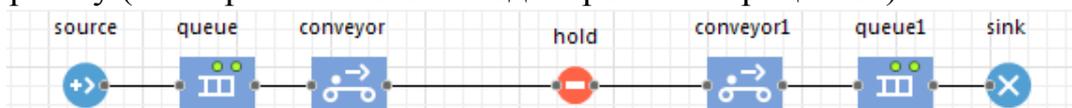
14. Настроить свойства блоков состояний диаграммы, чтобы менялись соответствующие значения параметров.
15. Запустить модель и проверить работу параметров.

*Создание презентации и ее анимация*

16. Используя элементы палитры Презентация: прямоугольник и два овала постройте в графическом поле объекта Main изображение светофора.
17. Настройте динамическое изменение цвета сигналов (овалов) светофора в зависимости от значений параметров.
18. Запустить модель и проверить работу светофора для пешеходов.

*Создание диаграммы процесса, моделирующей движение автотранспорта*

19. Разместить в графическом поле объекта Main следующую потоковую диаграмму (палитра Библиотека моделирования процессов):



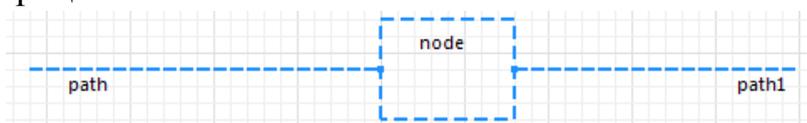
и настроить свойства ее элементов.

Элемент	Характеристика элемента	Свойства элемента
source	генерирует заявки определенного типа	Прибывают согласно: Времени между прибытиями Время между прибытиями: exponential( 0.1 ) сек
queue queue1	моделирует очередь заявок, ожидающих приема объектами, следующими далее в диаграмме процесса	Вместимость: 10
conveyor conveyor1	конвейер, который перемещает заявки по пути заданной длины с заданной скоростью (одинаковой для всех заявок)	Длина задается: согласно пути Скорость: 55м/с
hold	блокирует/разблокирует поток заявок на определенном участке диаграммы	Режим: ручную
sink	уничтожает поступившие заявки	-

20. Запустить модель и проверить работу диаграммы.

*Создание анимации*

21. Разместить в графическом поле объекта Main с помощью инструментария меню Разметка пространства (палитра Библиотека моделирования процессов) следующее графическое представление работы диаграммы процесса:



22. Создать нового агента Car с анимацией в виде автомобиля.
23. Связать элементы диаграммы source, hold и sink с новым агентом Car (свойства Новый агент: или Тип агента:).
24. Связать элементы диаграммы queue и conveyor с элементом разметки пространства path и элементы диаграммы queue1 и conveyor1 с элементом разметки пространства path1 (свойство Место агентов:).
25. В диаграмме состояний для автомобилей добавить блокировку и разблокировку элемента диаграммы процесса hold (методы block() и unblock()).
26. Запустить модель и проверить работу анимации диаграммы.
27. Создать диаграмму процесса и анимацию для движения пешеходов аналогично.

### Практические задания к лабораторным работам 3,4

1. Разработать модель с использованием инструментов различных палитр программы Anylogic согласно варианту своего задания.
2. Подготовить ответы на контрольные вопросы.
3. Составить отчет по лабораторной работе.

#### Варианты заданий

№ варианта	Задание
1	<p>а) Создайте модель, в презентации поместите прямоугольник, цвет которого будет задаваться состоянием стейтчарта (красный, синий, черный). Переключение между состояниями – автоматическое, через каждые 2 секунды.</p> <p>б) Создайте модель секундомера, в презентации поместите кнопку (СБРОС) и текст, показывающий секунды. При нажатии на кнопку секундомер должен сбрасывать показания и останавливаться.</p>
2	<p>а) Создайте модель со стейтchartом с четырьмя состояниями и одной кнопкой. Переключение между состояниями должно происходить только при нажатии этой кнопки. Информация о состоянии стейтчарта должна выводиться в текстовом поле.</p> <p>Создайте модель с реплицированными объектами ball (10 шт). В классе Ball создайте событие, которое на 5 секунде работы модели удалит объект №8 объектом №3.</p>
3	<p>а) Создайте модель светофора с двумя красными сигналами, моргающими поочередно.</p> <p>б) Создайте модель с реплицированными объектами ball (10 шт). В классе Ball создайте параметр N, который должен показывать порядковый номер объекта и кнопку, при нажатии которой, параметр N объекта №3 станет равным 111.</p>

4	<p>а) Создайте модель, на презентации поместите скругленный квадрат, цвет заливки которого, каждую секунду должен принимать случайные значения.</p> <p>б) Создайте модель счетчика, в презентации поместите слайдер и текст, показывающий число импульсов. Слайдер должен регулировать частоту следования импульсов.</p>
5	<p>а) Создайте модель, на презентации поместите зеленый овал, каждую секунду цвет овала должен становиться бледнее, до тех пор, пока не станет белым.</p> <p>Создайте модель клавиатуры калькулятора: в презентацию поместите текст и кнопки (от 0 до 9), при нажатии которых к тексту справа будут прибавляться нажатые цифры, примерно так, как это выглядит при вводе числа на калькуляторе.</p>
6	<p>а) Создайте модель, на презентации поместите синий треугольник, каждую секунду цвет треугольника должен становиться темнее, до тех пор, пока не станет черным.</p> <p>б) Создайте модель с реплицированными объектами ball (10 шт). В классе Ball создайте параметр P, который случайным образом примет значение – истина или ложь. В классе Main создайте кнопку, при нажатии которой, параметр P во всех объектах поменяет свое значение на противоположное.</p>
7	<p>а) Создайте модель, в презентацию поместите овал, размеры которого будут изменяться с помощью 2-х слайдеров, соответственно по горизонтали и вертикали.</p> <p>б) Создайте модель с реплицированными объектами ball (10 шт). В классе Ball создайте параметр P, который случайным образом примет значение – истина или ложь. В классе Main создайте кнопку, при нажатии которой, подсчитается число объектов с истинным и с ложным значением параметра P.</p>
8	<p>а) Создайте модель, в презентации поместите кнопку, текст на кнопке должен выводить сообщение о том, сколько раз нажата кнопка.</p> <p>б) Создайте модель, на презентации поместите квадрат 150x150 пикселей, овал (размером 3 пикселя) и 2 слайдера (горизонтальный и вертикальный, расположенные вдоль сторон квадрата). Положение овала должно динамически задаваться внутри квадрата слайдерами.</p>
9	<p>а) Создайте модель секундомера, в презентации поместите кнопку (старт/стоп) и текст, показывающий секунды с десятичными долями. При нажатии на кнопку секундомер должен останавливаться, при повторном нажатии, продолжать отсчет.</p> <p>б) Создайте модель для поиска максимума функции <math>y = f(x)</math>, описываемой уравнением: <math>y = x^{\sin(x)}</math></p>
10	<p>а) Создайте модель таймера обратного отсчета, в презентации поместите кнопку ПУСК; текст, показывающий оставшиеся секунды и поле для установки времени. При достижении нуля показания таймера должны</p>

	<p>останавливаться.</p> <p>б) Создайте модель секундной стрелки: в презентацию поместите линию, которая должна вращаться как секундная стрелка часов.</p>
--	---

### *Контрольные вопросы*

1. Дайте классификацию переменных в Java.
2. Как выполняется вызов методов и доступ к свойствам класса Java?
3. Какими способами реализуется условная конструкция в языке Java?
4. Дайте определение стейтчарту и диаграмме процесса.
5. Объясните для чего используется каждый элемент меню палитры Диаграмма состояний.
6. Объясните назначение таких элементов меню палитры Библиотека моделирования процессов, как Source, Sink, Delay, Queue, Conveyor, Hold.

## Лабораторная работа 5

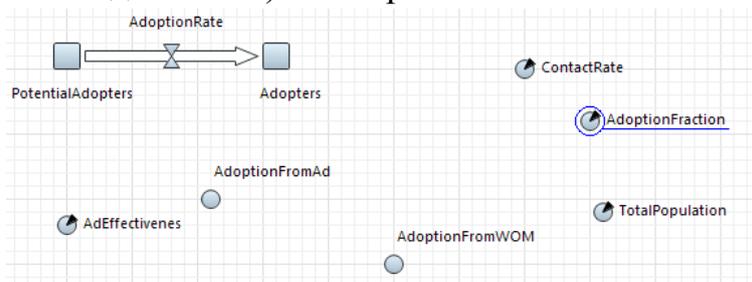
### Тема: Системно-динамическая модель

**Цель:** изучить возможности создания системно-динамической модели в программе Anylogic.

**Пример.** Разработать модель системной динамики исследования рынка новых продуктов (или распространения среди населения инноваций и новых продуктов) по Франку Бассу. Изначально существует определенное количество потенциальных потребителей продукции, которые под воздействием рекламы или под влиянием общения с потребителями, которые уже купили продукт, становятся владельцами продукта. Эффективность рекламы пропорциональна числу людей, на которых она действует, т.е. числу потенциальных покупателей. В свою очередь, эффективность влияния общения зависит от числа людей, уже купивших продукт.

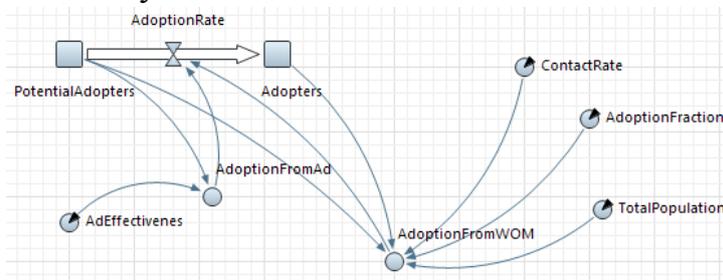
### *Ход работы*

1. Создать новую модель: Файл -> Создать -> Модель.
2. Разместить в графическом поле объекта Main следующие элементы (палитра Системная динамика) и настроить их свойства.



Элемент	Характеристика элемента	Свойства элемента
Накопители		
PotentialAdopters	потенциальные потребители продукции	Начальное значение: 100000
Adopters	потребители, которые уже купили продукт	Начальное значение: 0
Потоки		
AdoptionRate	поток, который моделирует интенсивность процесса приобретения продукта	
Переменные		
AdoptionFromAd	число потребителей продукта, которые его приобрели под влиянием рекламы	
AdoptionFromWOM	число потребителей продукта, которые его приобрели под влиянием общения с потребителями, которые уже купили продукт	
Параметры		
TotalPopulation	численность населения	Значение по умолчанию: 100000
ContactRate	число контактов	Значение по умолчанию: 100
AdEffectiveness	эффективность рекламы (или вероятность приобретения продукта под воздействием рекламы)	Значение по умолчанию: 0.011
AdoptionFraction	сила убеждения (или доля контактов, которые приводят к продаже продуктов )	Значение по умолчанию: 0.015

3. Создать связи между элементами модели:



4. Модель реализации продукции описывается системой уравнений (здать их в свойствах соответствующих элементов модели):

$$\frac{d(\text{PotentialAdopters})}{dt} = -\text{AdoptionRate}$$

$$\frac{d(\text{Adopters})}{dt} = \text{AdoptionRate}$$

$$\text{AdoptionFromAd} = \text{PotentialAdopters} * \text{AdEffectiveness}$$

$$\text{AdoptionFromWOM} = \text{Adopters} * \text{ContactRate} * \text{AdoptionFraction} * \frac{\text{PotentialAdopters}}{\text{TotalPopulation}}$$

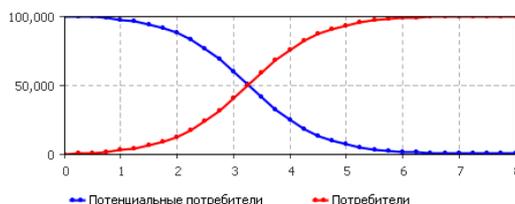
$$\text{AdoptionRate} = \text{AdoptionFromAd} + \text{AdoptionFromWOM}$$

5. Настроить эксперимент модели Simulation: модель должна останавливаться в заданное время, при достижении модельного времени 8 единиц.

6. Запустить модель и проверить ее работу.

### Сбор статистики

7. Разместить в графическом поле объекта Main временной график (палитра Статистика) для построения диаграммы изменения количества потребителей продукта (накопитель Adopters) и количества потенциальных потребителей продукта (накопитель PotentialAdopters), настроить его свойства, чтобы он имел следующий вид:



8. Разместить в графическом поле объекта Main временной график (палитра Статистика) для построения диаграммы изменения интенсивности продаж (поток AdoptionRate), настроить его свойства, чтобы он имел следующий вид:



9. Запустить модель и проверить построение графиков.

### Практические задания к лабораторной работе 5

1. Разработать модель системной динамики исследования рынка новых продуктов (или распространения среди населения инноваций и новых продуктов) по Франку Бассу для двух вариантов значений исходных данных. Сравнить результаты и объяснить изменения в поведении модели.
2. Подготовить ответы на контрольные вопросы.
3. Составить отчет по лабораторной работе.

#### Варианты заданий

№ варианта	Эффективность рекламы	Сила убеждения	Длительность процесса
1	0.010	0.01	7
	0.020	0.05	7
2	0.010	0.02	8
	0.020	0.04	8
3	0.010	0.03	9
	0.015	0.03	9
4	0.005	0.04	10
	0.010	0.01	10
5	0.005	0.05	10
	0.015	0.02	10
6	0.005	0.01	11

	0.015	0.01	11
7	0.015	0.03	9
	0.005	0.04	9
8	0.015	0.04	8
	0.005	0.03	8
9	0.010	0.05	7
	0.015	0.02	7
10	0.010	0.02	10
	0.005	0.05	10

### *Контрольные вопросы*

1. Объясните концепцию построения моделей системной динамики.
2. Объясните для чего используется каждый элемент меню палитры Системная динамика.
3. Как построить уравнение для накопителя модели системной динамики?

## **Лабораторная работа 6**

### **Тема: Агентная модель**

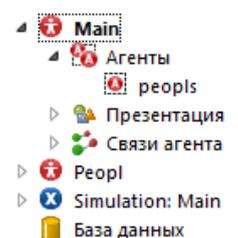
**Цель:** изучить возможности агентного моделирования в программе Anylogic.

**Пример:** Разработать многоагентную модель исследования рынка новых продуктов (или распространения среди населения инноваций и новых продуктов) по Франку Бассу.

### *Ход работы*

#### *Создание модели*

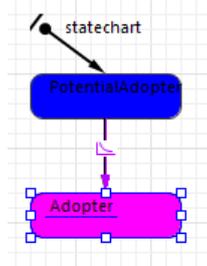
1. Создать новую модель: Файл -> Создать -> Модель.
2. Разместить в графическом поле объекта Main элемент Агент (палитра Агент) при работе с Мастером создания агентов:
  - шаг 1 – выбрать Популяция агентов
  - шаг 2 – создать новый тип агента
  - шаг 3 – имя нового типа задать Peopl
  - шаг 4 – выбрать анимацию агента: 2D, пиктограмма человека
  - шаг 5 – добавить параметр AdEffectiveness (обозначает эффективность рекламы), тип double, использовать стохастическое выражение: 0.011 (означает, что в единицу модельного времени 1.1% потребителей приходит к решению о приобретении продукта)
  - шаг 6 – задать размер популяции 500 агентов
  - шаг 7 – установить следующие настройки среды Main - тип пространства: Непрерывное, Размер 600\*350, установить галку Применить случайное расположение, Тип сети: Нет сети/Задается пользователем



3. Проверить наличие в объекте Main - агентов reorls и в структуре модели - объекта Peopl, который будет описывать поведение агентов.
4. В графическом поле объекта Main выделить фигуру анимации популяции агентов и в панели Свойств выбрать опцию Рисовать агента со сдвигом от данной точки.
5. Запустить модель и проверить наличие ошибок.

### *Моделирование покупок под влиянием рекламы*

6. В графическом поле объекта Peopl разместить диаграмму состояний.



7. Настроить свойства элементов диаграммы следующим образом.

Элемент	Характеристика элемента	Свойства элемента
PotentialAdopter	потенциальные потребители продукции	Цвет заливки: blue Действие при входе: shapeBody.setFillColor(blue);
Adopter	потребители, которые уже купили продукт	Цвет заливки: magenta Действие при входе: shapeBody.setFillColor(magenta);
transition	переход от потенциальных потребителей к потребителям, которые купили продукт под влиянием рекламы	Происходит: с заданной интенсивностью Интенсивность: AdEffectiveness

shapeBody - является именем фигуры анимации человека, которую выбрали в мастере создания агентов (найдите в панели Проекты - тип агента Peopl - раздел Презентация - внутри группы person - фигура shapeBody).

8. Настроить эксперимент модели Simulation: модель должна останавливаться в заданное время, при достижении модельного времени 8 единиц.
9. Запустить модель и проверить ее работу.

### *Сбор статистики*

10. В графическом поле объекта Main выделить вложенный объект reorls. В панели Свойства для объекта reorls в разделе Статистика выбрать Добавить функцию сбора статистики.

11. Установить имя функции PotentialAdopters, Тип: кол-во, Условие: `item.statechart.isStateActive(item.PotentialAdopter)`

Условие выбирает тех агентов, которые являются потенциальными потребителями (состояние PotentialAdopter).

item – это агент (элемент реплицированного объекта people).

12. Аналогично добавить для объекта people еще одну функцию Adopters, которая будет подсчитывать количество потребителей, купивших продукт (состояние Adopter).

13. Разместить в графическом поле объекта Main временной график и настройте его свойства, чтобы он отображал динамику изменения потенциальных потребителей (вызвать функцию people.PotentialAdopters()) и потребителей, которые купили товар (вызвать функцию people.Adopters()).

14. Запустить модель и проверить ее работу.

#### *Моделирование покупок под влиянием общения*

15. В графическом поле объекта People добавить два параметра:

Параметр	Характеристика	Свойства
ContactRate	среднегодовое количество встреч человека	Значение по умолчанию: 100 Тип: int
AdoptionFraction	сила убеждения человека, показывающая сколько людей он может уговорить на покупку	Значение по умолчанию: 0.015 Тип: double

16. В стейтчарт в состоянии Adopter добавить внутренний переход и установить следующие его свойства:

Переход	Характеристика	Свойства
transition1	переход посылает сообщение «Buy!» случайно выбранному человеку	Интенсивность: ContactRate Действие: send("Buy!", RANDOM);

17. Добавить еще один переход из состояния PotentialAdopter в состояние Adopter и установить следующие его свойства:

Переход	Характеристика	Свойства
transition2	переход от потенциальных потребителей к потребителям, которые купили продукт, при получении сообщения «Buy!»	Происходит: при получении сообщения Тип сообщения: String При получении заданного сообщения: "Buy!"

18. Изменить свойства класса People: в панели Свойства установить Действие на шаге: statechart.receiveMessage("Buy!");

19. Запустить модель и проверить ее работу

## Практические задания к лабораторной работе 6

1. Разработать агентную модель исследования рынка новых продуктов (или распространения среди населения инноваций и новых продуктов) по Франку Бассу для двух вариантов значений исходных данных. Сравнить результаты и объяснить изменения в поведении модели.

### *Варианты заданий*

№ варианта	Эффективность рекламы	Сила убеждения	Длительность процесса
1	0.010	0.01	7
	0.020	0.05	7
2	0.010	0.02	8
	0.020	0.04	8
3	0.010	0.03	9
	0.015	0.03	9
4	0.005	0.04	10
	0.010	0.01	10
5	0.005	0.05	10
	0.015	0.02	10
6	0.005	0.01	11
	0.015	0.01	11
7	0.015	0.03	9
	0.005	0.04	9
8	0.015	0.04	8
	0.005	0.03	8
9	0.010	0.05	7
	0.015	0.02	7
10	0.010	0.02	10
	0.005	0.05	10

2. Подготовить ответы на контрольные вопросы.

3. Составить отчет по лабораторной работе.

### *Контрольные вопросы*

1. Объясните возможности агентного моделирования.

2. Что такое агент модели?

3. Объясните для чего используется каждый элемент раздела Компоненты агента в меню палитры Агент.

## СПИСОК ЛИТЕРАТУРЫ

1. Маликов, Р.Ф. Практикум по имитационному моделированию сложных систем в среде AnyLogic 6: учеб. пособие / Р. Ф. Маликов. – Уфа: Изд-во БГПУ, 2013. – 296с.
2. Куприяшкин, А.Г. Основы моделирования систем: учеб. пособие / А.Г. Куприяшкин; Норильский индустр. ин-т. – Норильск: НИИ, 2015. – 135 с.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**  
К ЛАБОРАТОРНЫМ РАБОТАМ ПО ДИСЦИПЛИНЕ  
**«МАТЕМАТИЧЕСКОЕ И ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ»**  
ДЛЯ СТУДЕНТОВ НАПРАВЛЕНИЙ ПОДГОТОВКИ  
09.03.03 «ПРИКЛАДНАЯ ИНФОРМАТИКА»,  
01.03.02 «ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА»  
(очная форма обучения)  
Часть I

**Составитель:**  
**В.В. Юданова**

**Технический редактор Л.В. Николаева**

Подписано в печать 23.11.2017. Формат 60x84/16.  
Бумага тип. №2. Гарнитура «Таймс». Печать офсетная.  
Печ. л. 1,75. Тираж 50 экз. Заказ 384.  
Издательство ТИ (ф) СВФУ, 678960, г. Нерюнгри, ул. Кравченко, 16.

---

Отпечатано в ТИ (ф) ФГАОУ ВО «СВФУ»  
г. Нерюнгри