

Основы Java для AnyLogic

Эта презентация является
частью стандартной
программы обучения
AnyLogic



Общие замечания

- Не требуется знать принципы объектно-ориентированного программирования
- Достаточно знать типы данных Java и понимать синтаксис выражений
- Обратите внимание на следующее:
 - Java учитывает регистр: переменная `MyVar` отлична от `myVar`!
 - Имена не могут содержать пробелов: имя “`My Var`” недопустимо!
 - Каждое действие должно заканчиваться “;”: `MyVar = 150;`
 - Каждая функция должна иметь скобки: `time()`, `add(a)`
 - Следует помнить о целочисленном делении: `3/2 = 1`, а не 1.5
 - Булевские переменные могут принимать только значения `true` и `false`, использовать значения 1 и 0 нельзя
 - Точка “.” позволяет перейти “внутрь” объекта: `agent.event.restart()`
 - Элементы массивов имеют индексы от 0 до N-1

Типы

- **Примитивные типы**

- `double` – вещественное число: 1.43, 3.6E18, -14.0
- `int` – целое число: 12, 16384, -5000
- `boolean` – булево значение: `true`, `false`

- **Составные типы – классы**

- `String` – текстовая строка: “AnyLogic”, “Привет!”
- `ArrayList`, `LinkedList` – коллекции объектов
- `HyperArray` – многомерный массив вещественных чисел в системно-динамических моделях
- ... и многие другие. См. справочник классов AnyLogic и Java

Выражения

- **Арифметические операции**

- **+**, **-**, *****, **/**, **%** (остаток от деления)
- При целочисленном делении дробная часть отбрасывается, например: **3 / 2** равняется **1**, **2 / 3** равняется **0**
- Операции типа умножение имеют приоритет над операциями типа сложение
- Оператор **+** также конкатенирует строки типа **String**

- **Операции сравнения**

- **>**, **>=**, **<**, **<=**, **==**, **!=**

- **Логические операции**

- **&&** (и), **||** (или), **!** (не)

- **Условный оператор**

- условие **?** рез-т-если-true **:** рез-т-если-false

- **Присваивание и сокращения**

- **=**, **+=**, **-=**, ***=**, **/=**, **%=**, **++**, **--**
- Пример: **a+=b** равносильно **a=a+b**



Операции одинакового приоритета выполняются слева направо

Для управления приоритетом используйте скобки

Примеры

- $5 \% 2 \equiv ?$
- $5 / 2 \equiv ?$
- $5. / 2 \equiv 5 / 2. \equiv ?$
- $(\text{double})5 / 2 \equiv ?$
- $a += b; \equiv ?$
- $a++; \equiv ?$

- $\text{"Any"} + \text{"Logic"} \equiv ?$
- Пусть $x = 14.3$, тогда:
 $\text{"x = "} + x \equiv ?$
- $\text{""} \equiv ?$
- $\text{""} + x \equiv ?$
- $y = x > 0 ? x : 0$
 $\equiv ?$
- $x == 5 \equiv ?$
- $x = 5 \equiv ?$

Вызов методов и доступ к полям

- **Вызов методов**

- Чтобы вызвать метод, укажите имя метода и после имени поставьте скобки. Если необходимо, в скобках через запятую укажите параметры. Примеры:

```
x = time();  
moveTo( getX(), getY() + 100 );  
traceln( "Population is increasing" );
```

- **Доступ к полям и методам**

- Чтобы обратиться к полю или методу элемента модели (стейтчарта, события, анимации), укажите имя этого элемента, добавьте точку '.', а затем укажите имя поля или метода. Примеры:

```
statechart.fireEvent( "go" );  
sum = sum + agents.get(i).x;
```

Комментарии в Java коде

- Есть два вида комментариев:

`/* text */`

Обычный комментарий: игнорируется весь текст, находящийся между ASCII символами `/*` и `*/` (как в C и C++).

```
/**
 * The class represents AnyLogic 3D animation. It contains the canvas object.
 *
 * @author Daniil Chunosov
 * @version 5.0
 */

public class Animation3DPanel extends javax.swing.JPanel ...
```

`// text`

Строчный комментарий: игнорируется весь текст от ASCII символов `//` и до конца строки (как в C++)

```
// Prepare Engine for simulation:
engine.start( root );
engine.runFast(); // fast mode – no animation
```

Реплицированные объекты

 objects [..]

 people [..]

- Реплицированный объект представляется массивом переменного размера. Элементы массива имеют индексы от 0 до N-1
 - Получение размера массива:
`people.size()`
 - Получение i-го элемента массива:
`people.get(i)`
 - Добавление нового объекта:
`add_people();`
 - Удаление объекта:
`remove_people(person);`

Встроенные функции

- Системные функции
 - `time()`, `getOwner()`, `pause()`, `isActive(...)`, и т.д.
- Математические функции
 - Базовые: `sqrt`, `sin`, `cos`, `tan`, `exp`, `log`, `round`, `zidz`, `xidz`, и т.д.
- Специальные функции
 - Случайные числа: `uniform`, `exponential`, `bernoulli`, `beta`, и т.д.
 - Для работы со временем: `delay`
- И другие ...
 - См. классы `Utilities`, `Presentable`, `ActiveObject` и `Agent` в справочнике классов AnyLogic

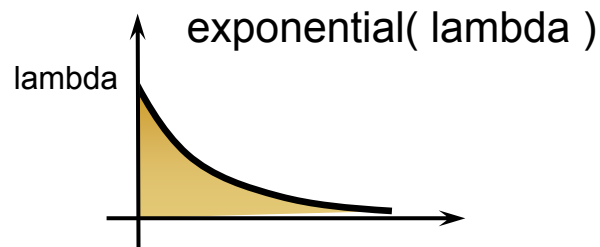
Вероятностные распределения

`uniform(min, max)`

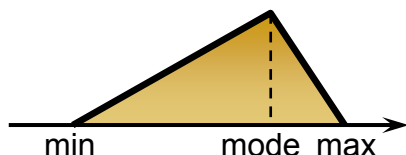


Описывает случайную величину, для которой вероятность попадания в любой интервал между `min` и `max` прямо пропорциональна длине этого интервала.

Используется для моделирования временных интервалов между случайными событиями. Важное свойство: отсутствие последействия.

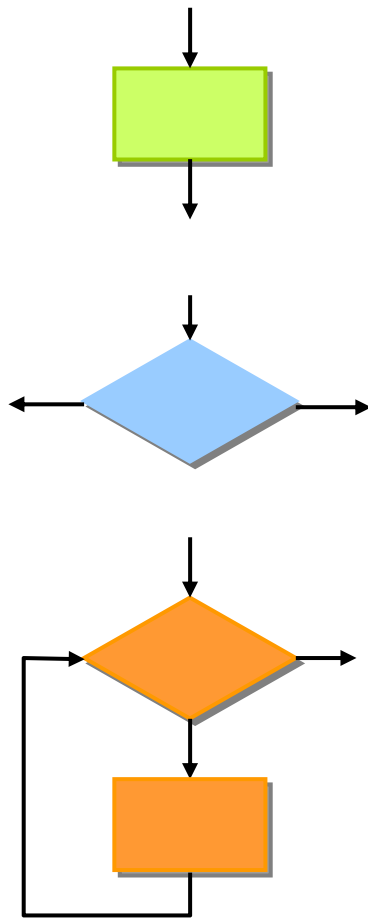


`triangular(min, mode, max)`



Обычно используется в условиях недостатка данных о длительности какого-либо процесса.

Основные конструкции языка



- **Присваивание или вызов метода:**

```
y = f(x1,x2) + 5*z;  
event.restart( uniform( 10, 20 ) );
```

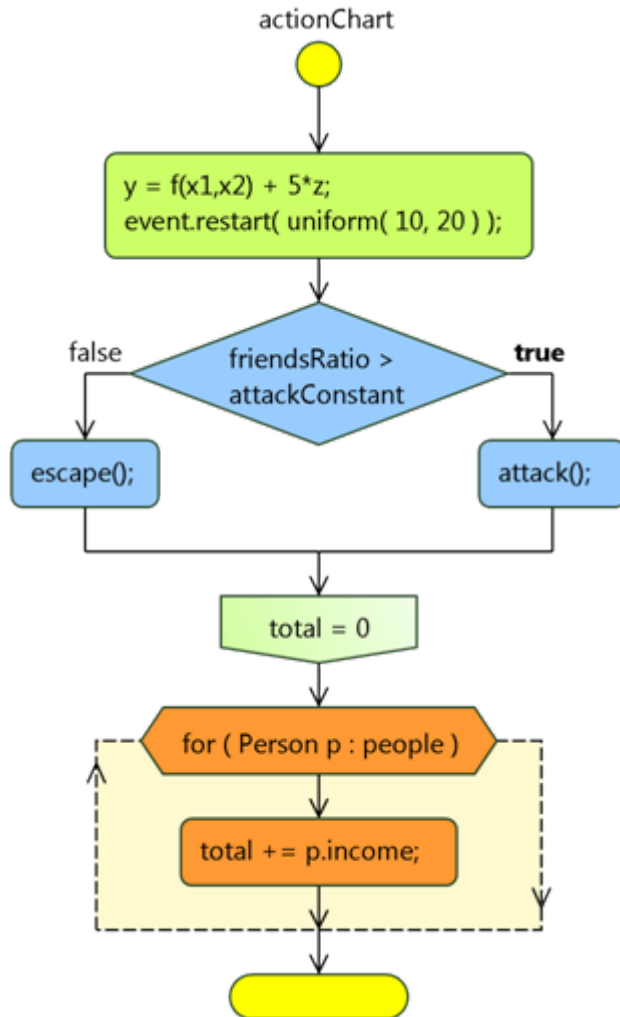
- **Ветвление по условию:**

```
if ( friendsRatio > attackConstant )  
    attack();  
else  
    escape();
```

- **Цикл:**

```
double total = 0;  
for ( Person p : people )  
    total += p.income;  
for( int i=0; i<100; i++ )  
    send( msg, RANDOM );
```

Основные конструкции языка



- Присваивание или вызов метода:

`y = f(x1,x2) + 5*z;`

`event.restart(uniform(10, 20));`

- Ветвление по условию:

`if (friendsRatio > attackConstant)`

`attack();`

`else`

`escape();`

- Цикл:

`double total = 0;`

`for (Person p : people)`

`total += p.income;`

`for(int i=0; i<100; i++)`

`send(msg, RANDOM);`

Коллекции

- Коллекции – это predefined классы, реализующие базовые структуры данных
- Одна и та же операция для разных классов занимает разное время

| | ArrayList Vector | LinkedList | HashSet HashMap | SortedSet SortedMap |
|----------------------|---------------------|------------|--------------------|------------------------|
| Получение размера | Const | Const | Const | Const |
| Добавление элемента | Const | Const | Const | Log |
| Удаление элемента | Linear | Linear | Const | Log |
| Удаление по индексу | Linear | Linear | - | - |
| Получение по индексу | Const | Linear | - | - |
| Поиск элемента | Linear | Linear | Const | Log |
| Поиск мин/макс | Linear | Linear | Linear | Const |

Источники информации

- Справочник классов AnyLogic
 - Раздел [API Reference](#) справочной системы AnyLogic
- Спецификация Java API
 - <http://java.sun.com>
- Лучшая книга по Java:
 - Bruce Eckel. Thinking in Java.
Доступна по адресу: <http://www.mindview.net/Books/TIJ>