

FSE598 前沿计算技术

模块 1 计算思维

单元 2 工作流与可视化编程

第 2 讲 VIPL 工作流编程

本讲座的英文版内容基于教材：

Y. Chen, G. De Luca Service-Oriented Computing and System Integration: Software, IoT, Big Data, and AI as Services, 8th edition, Kendall Hunt Publishing, 2022.

<https://www.public.asu.edu/~ychen10/book/socsi.html>

学习

- ❑ VIPLE workflow及可视化编程
- ❑ VIPLE 基本活动和服务
- ❑ VIPLE 输入和输出
- ❑ VIPLE 活动和参数传递



ASU VIPLE 支持仿真和实体机器人平台

自动驾驶



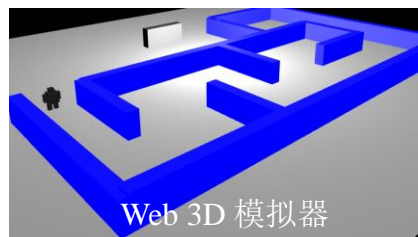
交通模拟



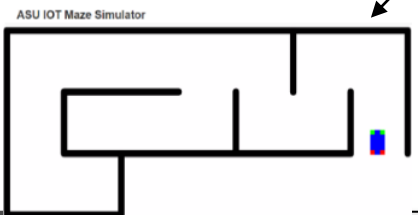
Unity
模拟器



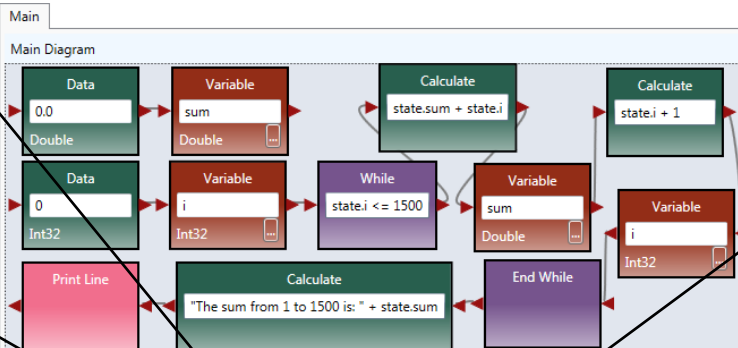
Web3D
模拟器



Web2D
模拟器



<http://venus.sod.asu.edu/VIPLE/>



飞行数据训练
与识别



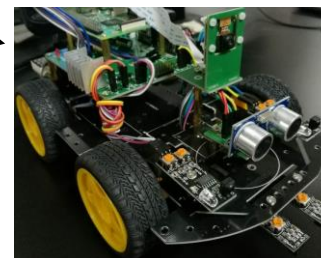
物理四旋翼

导盲犬和
人形机器人



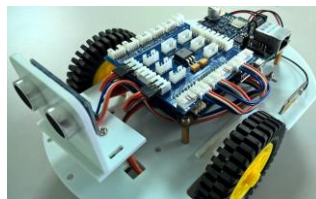
BIOLOID

智能汽车

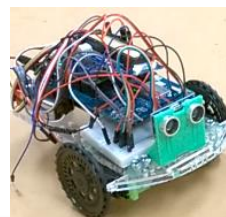


Raspberry Pi 机器人

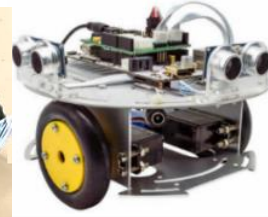
Lego
EV3



Intel Galileo 机器人



Intel Edison 机器人



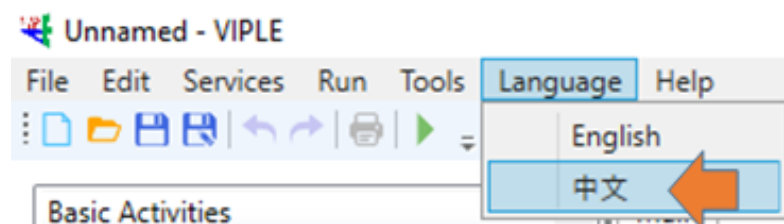
pcDuino 机器人

Computational Thinking

VIPLE 特征

- ❑ 通用控制流编程
- ❑ 面向服务的计算，支持 RESTful 和 WSDL 标准
- ❑ 并行/多线程编程，具有底层线程安全（同步）
- ❑ 事件驱动编程，具有内置和自定义事件
- ❑ 工作流和可视化编程
- ❑ 物联网和机器人编程
- ❑ 通过服务支持机器学习

VIPLE 基本活动



基本活动

自定义活动

变量

计算

数据

与并

或并

如果

开关

条件循环

退出

结束循环

注释

→ **Activity:** 用于创建一新模块。

→ **Variable:** 支持基本的类型(Int32, Double, String, Boolean, etc.)。

→ **Calculate:** 计算程序语言的一般表达式的值。

→ **Data:** 导入程序语言中的常数值。

→ **Join:** 与并将等待所有线程或输入数据的到来才能进入下一步。
所有数据可以通过成员函数"."来访问。

→ **Merge:** 或并只需等待第一个线程或输入数据的到来就能进入下一步。

→ **If:** 条件选择。允许多个条件顺序选择。

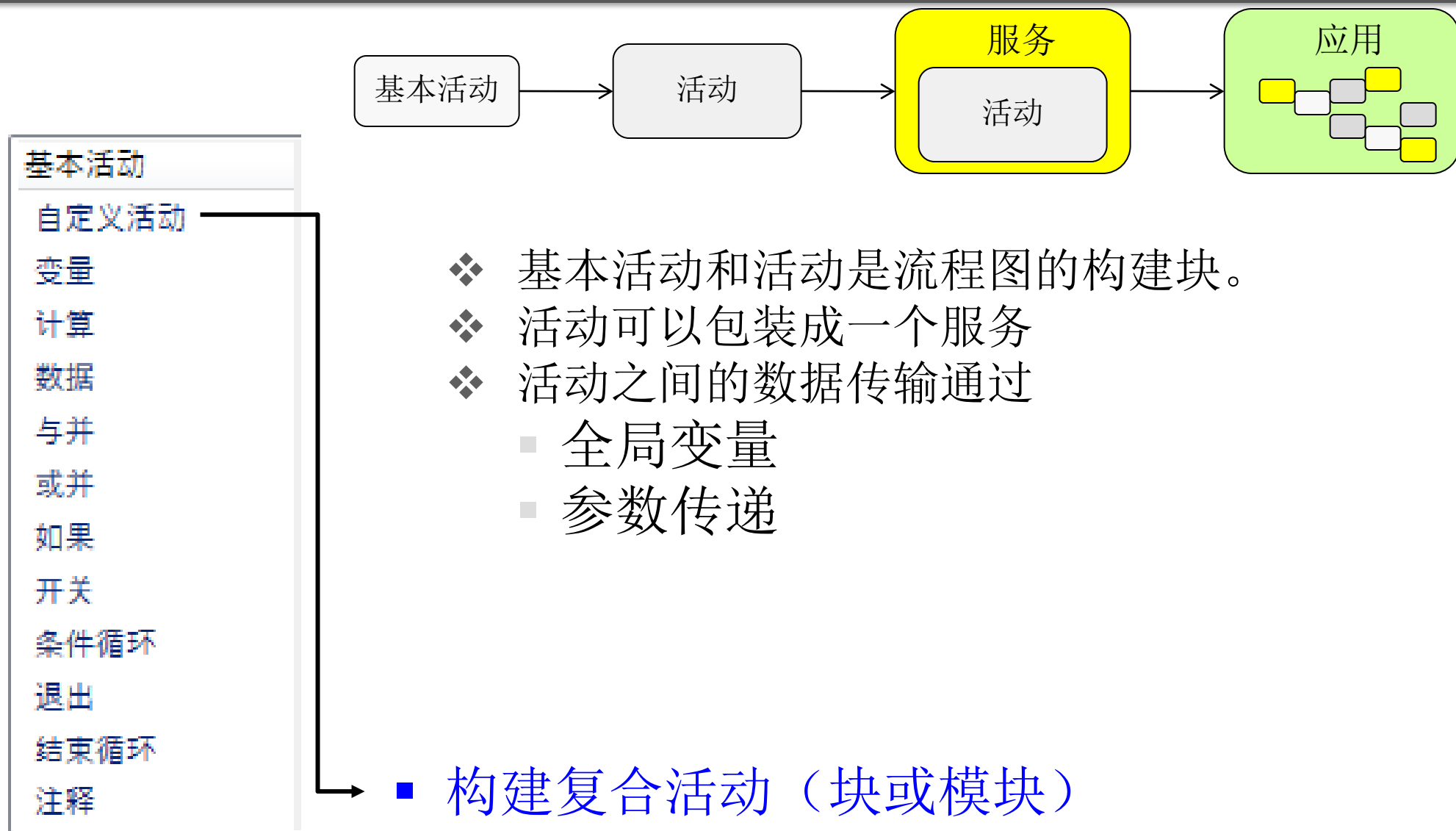
→ **Switch:** 根据输入值与所给值的匹配决定下一步的路径。

→ **While:** 循环开始点;

→ **Break:** 退出循环

→ **End While:** 循环的结束点。自动跳到循环开始点。

基本活动（继续）

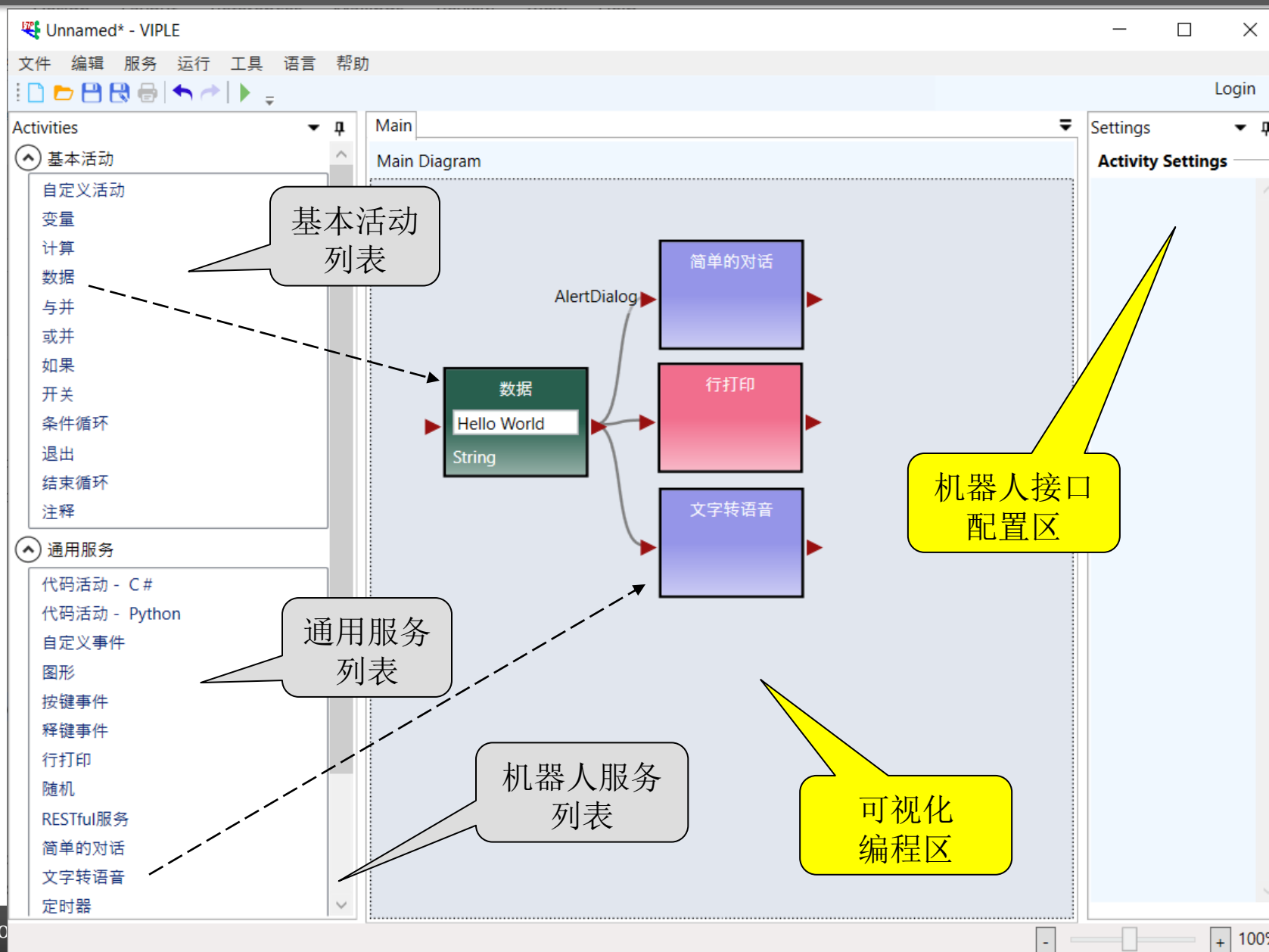


λ -代数

- VIPLE 是基于坚实的数学基础而开发的
- π -代数 定义并发计算过程
- π -代数 中的每个构造都在 VIPLE 中实现，具有相同的语义
- 可使用 π -代数 来验证每个 VIPLE 程序的正确性

π -代数 组件	计算思维图中的 π -代数 组件的图示	与图相对应的 VIPLE 活动	底层 CS 概念/计算 思维
Sequential Process $S ::= \text{new } a(A B).0$ $A ::= \tau_A.\bar{a}(x).0$ $B ::= a(x).\tau_B.\bar{b}(y).0$		活动块 变量 计算 数据	程序过程
Split $n \geq 2$ $S ::= \text{new } a1a2 \dots an(A B1 B2 \dots Bn)$ $A ::= \tau_A.\bar{a1}(x).\bar{a2}(x) \dots \bar{an}(x)$ $Bn ::= an(x).\tau_{Bn}.\bar{bn}(y)$		所有 VIPLE 活动都支持 拆分或扇出 	创建并行进程
Join $n \geq 1$ $S ::= \text{new } a1a2 \dots an(A1 A2 \dots An B)$ $An ::= \tau_{An}.\bar{an}(x)$ $B ::= a1(x1).a2(x2) \dots an(xn).\bar{b}(y)$			等待所有并行线程 并入，例如参数传 递；
Merge $n \geq 1$ $S ::= (A1 A2 \dots An B)$ $An ::= \tau_{An}.\bar{a}(x)B ::= !a(x).\bar{b}(y)$			选择并行线程中第 一个到达的结果
Choice (if) $n \geq 2$ $S ::= \text{new } a1a2 \dots an(A B1 B2 \dots Bn)$ $A ::= \text{if } bexpr1 \text{ then } \bar{a1}(x) \text{ else } \bar{a2}(x) \text{ if } bexpr2 \text{ then } \bar{a2}(x) \text{ else } \bar{a3}(x) \text{ if } \dots \text{ else if } bexpr(n-1) \text{ then } \bar{a(n-1)}(x) \text{ else } \bar{an}(x) Bn ::= an(x).\tau_{Bn}.\bar{bn}(y)$			条件语句； 嵌套语 句； 布尔运算符
Loop (While ... End While) $S ::= A B C$ $A ::= d(y).$ $B ::= a(x).\tau_B.\bar{b}(y)C ::= b(y).\bar{d}(y)$ $\text{if } bexpr \text{ then } \bar{a}(x) \text{ else } \bar{c}(y)$			迭代语句； 条件语 句；

VIPLE 编程界面



VIPL 服务列表

Activities

基本活动

- 自定义活动
- 变量
- 计算
- 数据
- 与并
- 或并
- 如果
- 开关
- 条件循环
- 退出
- 结束循环
- 注释

通用服务

- 代码活动 - C #
- 代码活动 - Python
- 自定义事件
- 图形
- 按键事件
- 释键事件
- 行打印
- 随机
- RESTful服务
- 简单的对话
- 文字转语音
- 定时器

机器人/IoT服务

- 机器人主机
- 机器人驱动器
- 机器人完整协调驱动
- 机器人消息输入
- 机器人消息
- 机器人运动
- 机器人电机
- 机器人彩色传感器
- 机器人距离传感器
- 机器人光传感器
- 机器人电机编码器
- 机器人声音传感器
- 机器人TORCS传感器
- 机器人TORCS对手传感器
- 机器人TORCS轨道传感器
- 机器人触觉传感器
- 机器人传感器流量
- 机器人TORCS命令
- 机器人交通驱动
- 机器人流量初始化
- 机器人交通计时器
- 机器人+移动-动力控制
- 机器人+转动-角度控制

乐高机器人服务

- 乐高主机EV3
- 乐高EV3彩色
- 乐高EV3驱动器
- 乐高EV3驱动器-时间控制
- 乐高EV3陀螺
- 乐高EV3舵机
- 乐高EV3电机-转角控制
- 乐高EV3电机-时间控制
- 乐高EV3按下触摸
- 乐高EV3释放触摸
- 乐高EV3超声

Quantum Basic Activities

- Controlled NOT Gate
- Hadamard Gate
- Pauli-X Gate
- Pauli-Y Gate
- Pauli-Z Gate
- Quantum Draw
- Quantum Measure

量子计算

Quantum Services

- Custom Quantum Service

VIPLE 编程：输出

基本活动

基本活动

自定义活动

变量

计算

数据

与并

或并

如果

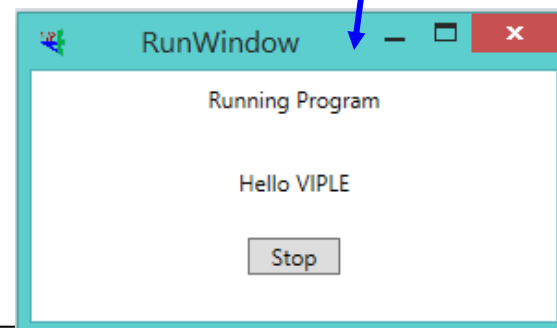
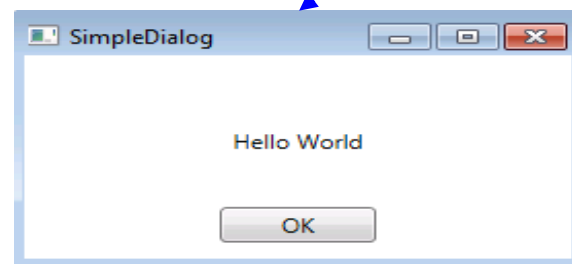
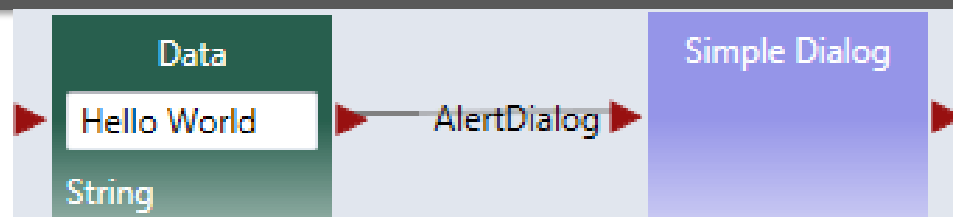
开关

条件循环

退出

结束循环

注释



通用计算和
事件服务

源代码活动

自定义事件

按键事件

释键事件

行打印

随机

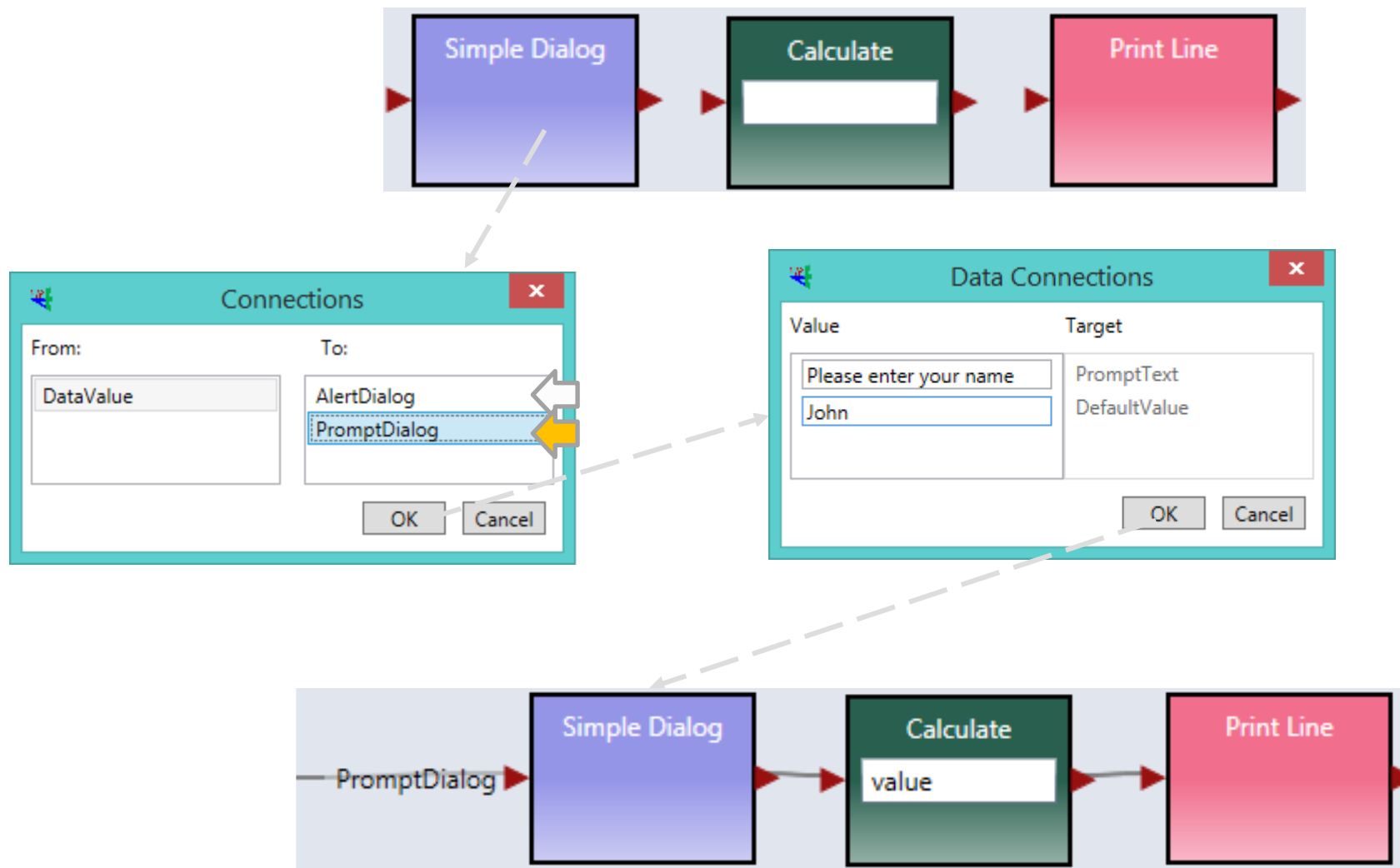
RESTful服务

简单的对话

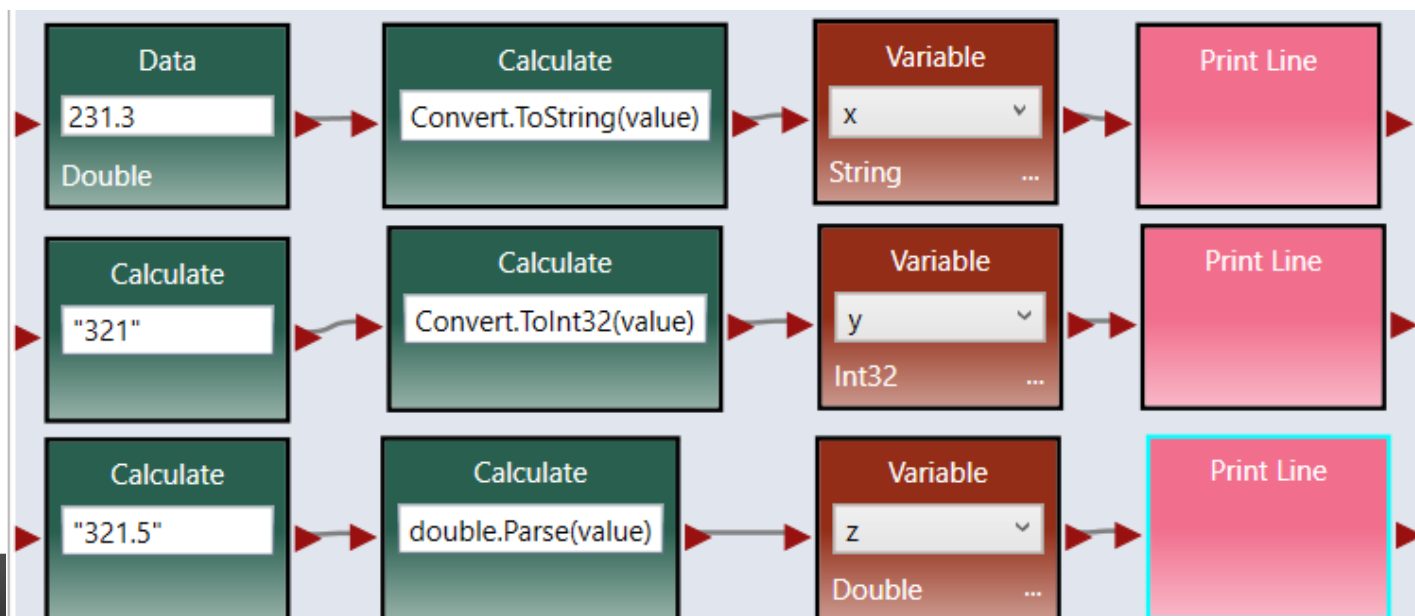
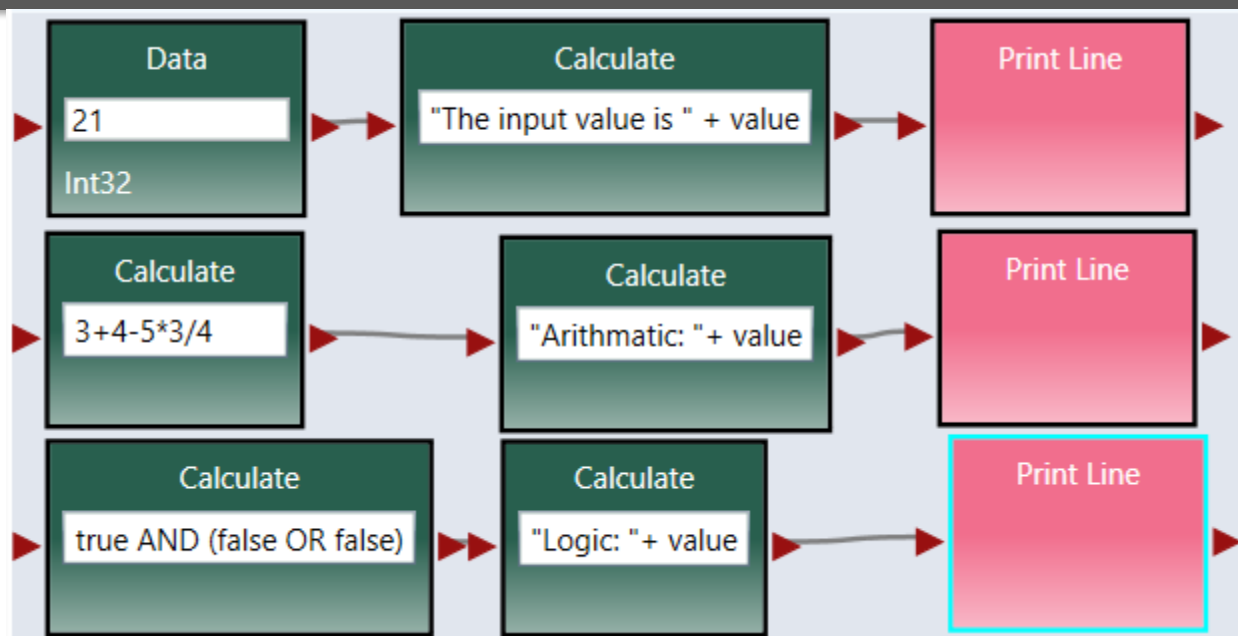
文字转语音

定时器

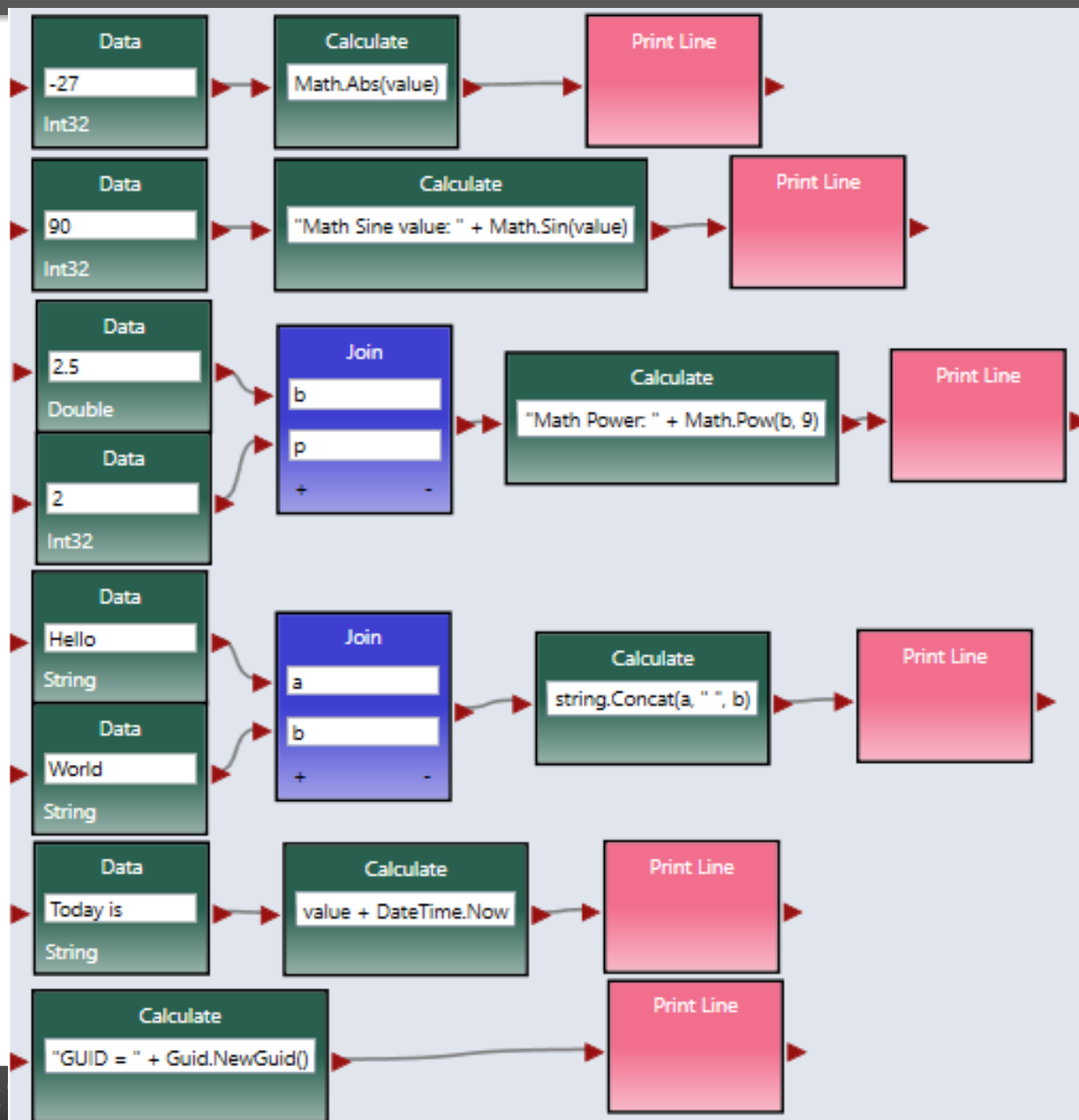
VIPLE 编程：输入



使用“计算”活动 (1)



使用“计算”活动 (2)



```
RunWindow
Running Program

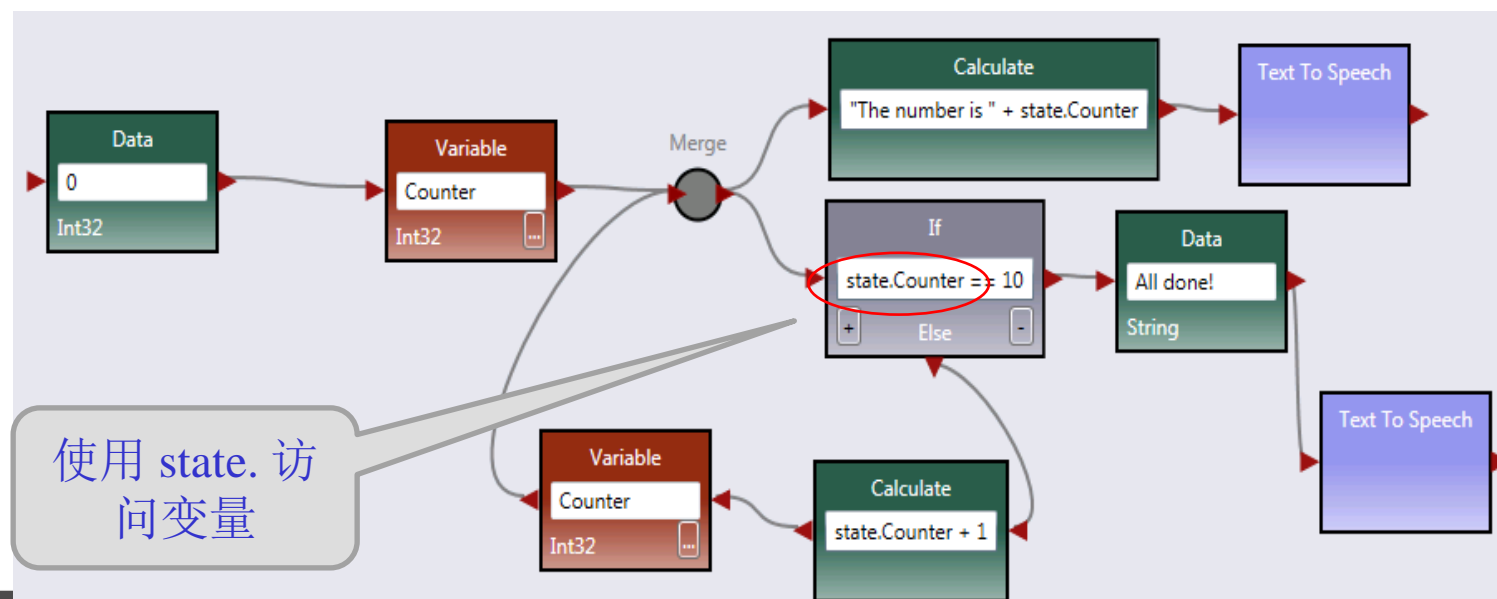
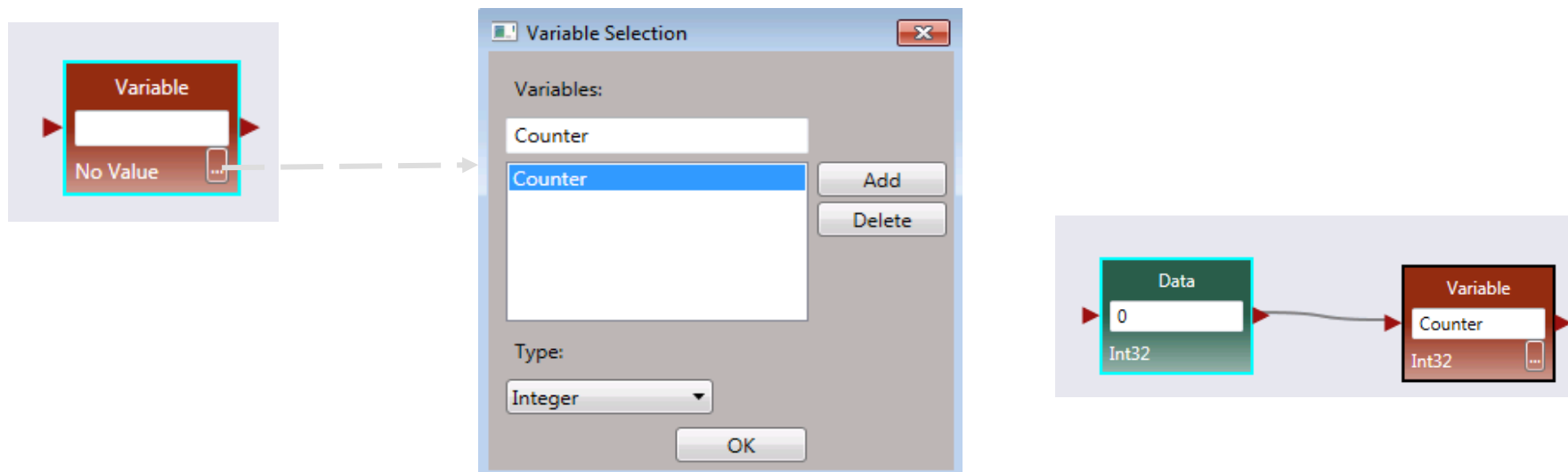
GUID = 8d9f866b-12e2-4fca-b7fa-
a71bb7006b95
The input value is 21
Today is 12/13/2018 2:11:04 AM
Math Power: 3814.697265625
27
Math Sine value: 0.893996663600558
Logic: False
Hello World
Arithmetic: 4
231.3
321
321.5
```

每次的执行顺序都不同

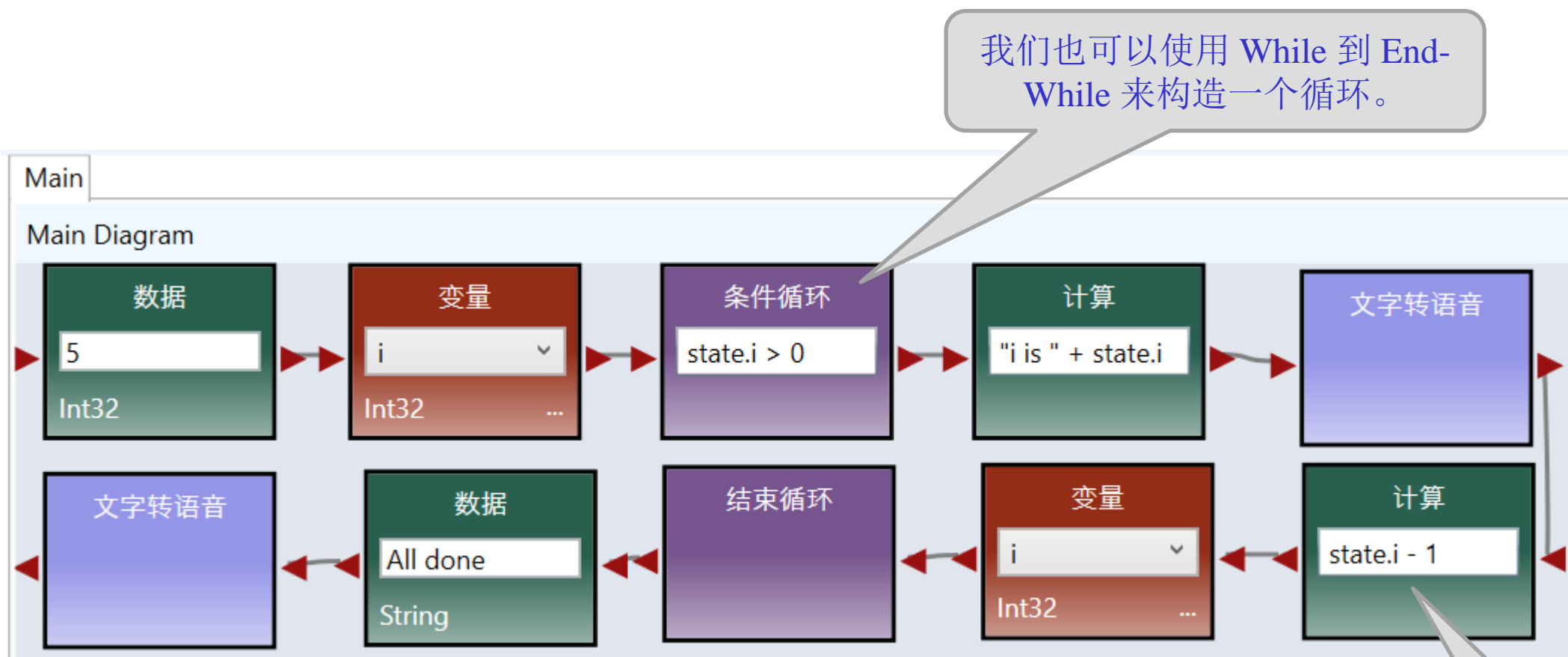
```
RunWindow
Running Program

The input value is 21
Logic: False
27
231.3
Arithmetic: 4
321
Hello World
321.5
Math Power: 3814.697265625
Math Sine value: 0.893996663600558
Today is 12/13/2018 2:12:35 AM
GUID = 990a56ea-
a2bd-4a9c-8135-6cebc0f67de3
```

VIPLE 编程：变量和循环

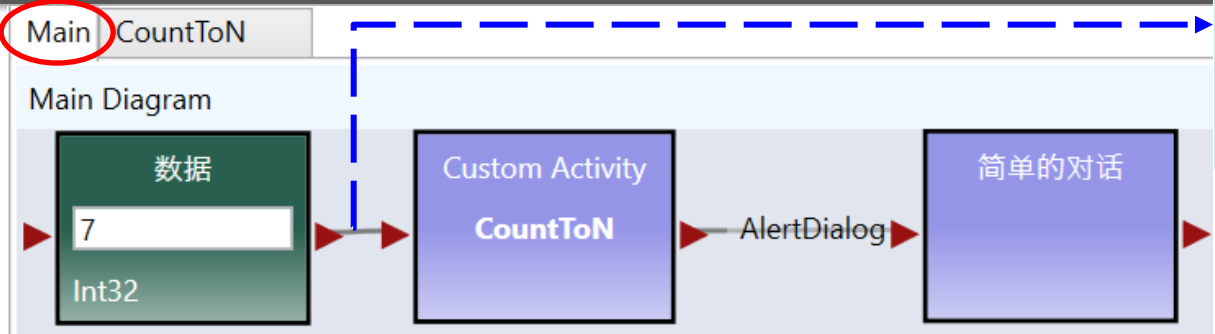


VIPLE 编程：变量和 While-循环



右击翻转块

活动和参数传递



Data Connections

Value	Target
value	Limit

OK Cancel

Variable Definition

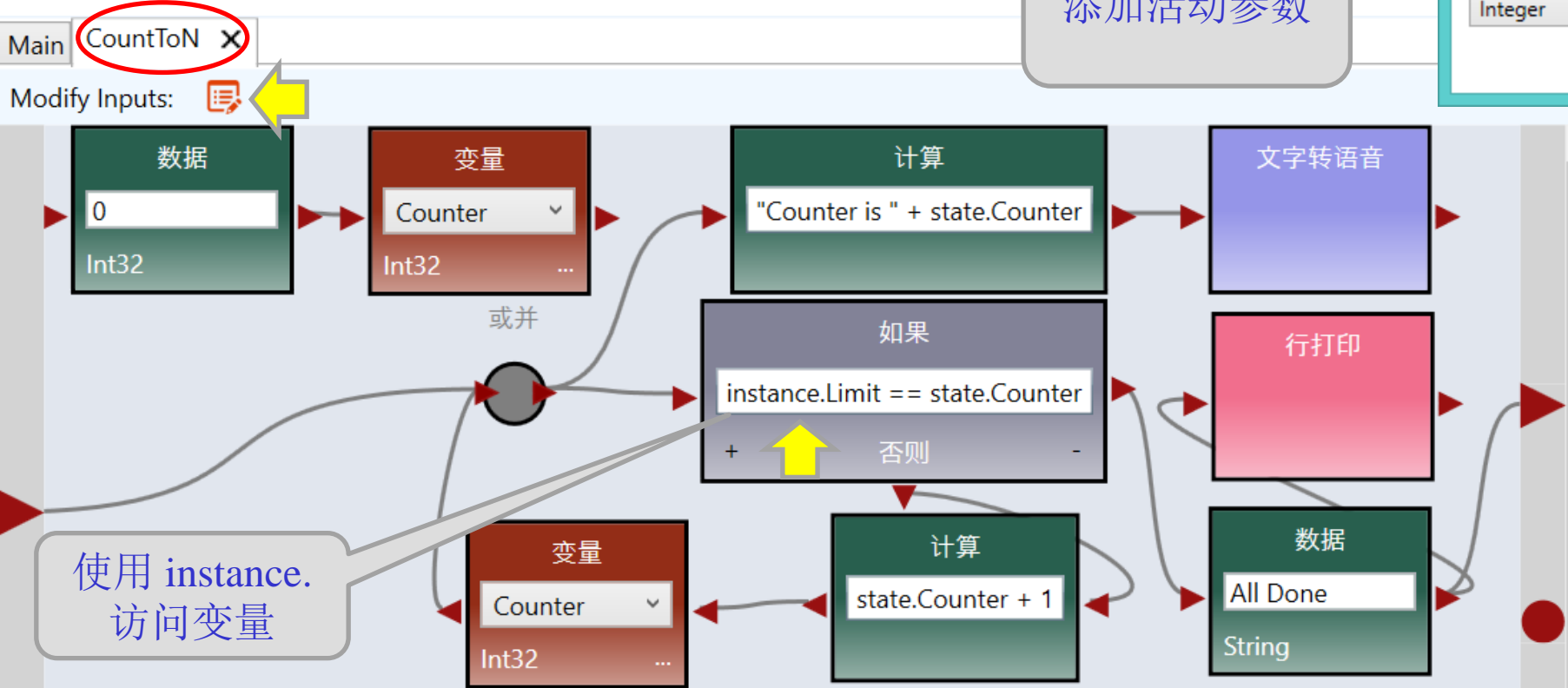
Limit

Type: Integer

OK

Add Delete

添加活动参数



使用 instance.
访问变量

带有两个参数的计数器活动

Input Variable Definition

Input Variables:

variableName

Limit

Base

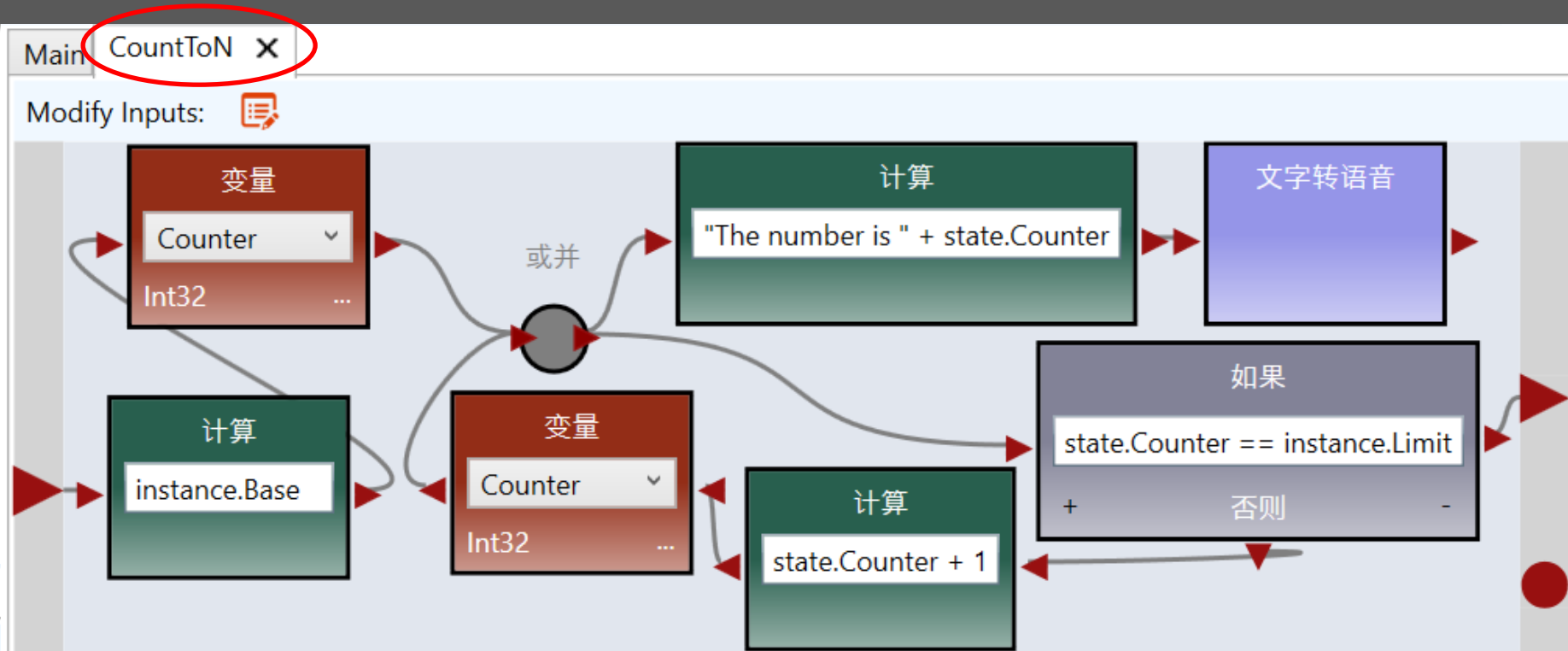
Add

Delete

Type:

Integer

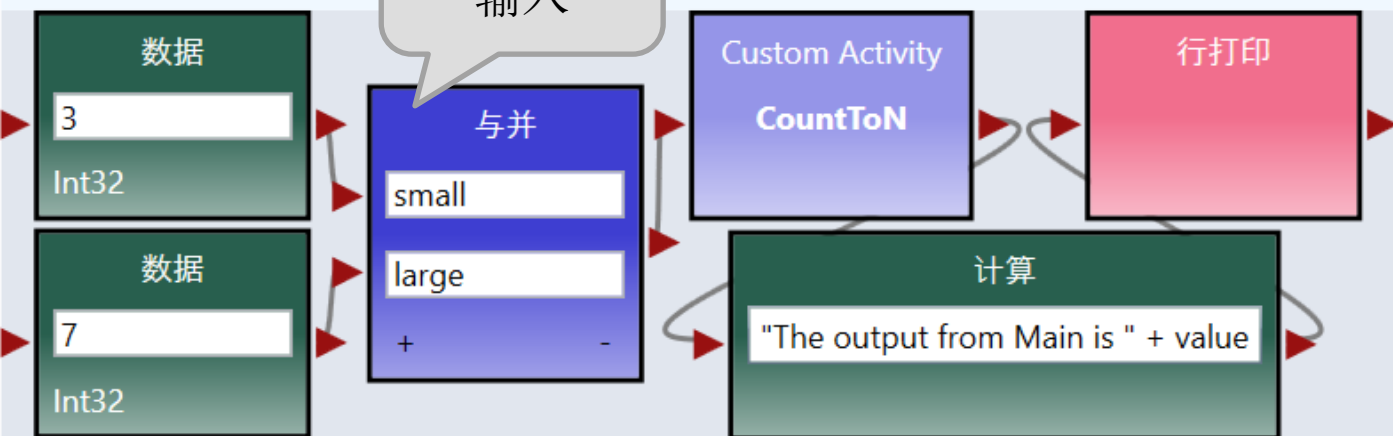
确定



Main CountToN

Main Diagram

导入两个输入



RunWindow

Running Program

The output from Main is 7

Stop

本讲小结

- ❑ VIPLE 环境
- ❑ VIPLE 编程范式和特征
- ❑ 通过活动和服务的拖放与连接进行可视化编程
- ❑ VIPLE 输入和输出
- ❑ VIPLE 活动定义和参数传递