BrainIT, mayo

2019, Bosque

Hola,

La charla de mayo 2019 del ciclo BrainIT iba a ser, así está anunciado, sobre La esfera mental. Sin embargo, finalmente, será sobre Bosque.

Inicialmente, por tanto, trataba de la nube, de un sistema operativo IoT, de un PaaS Cloudfoundry. Sin embargo, finalmente, será sobre un nuevo, y reciente, lenguaje de programación.

La charla compone de tres partes:

a) una explicación introductoria al lenguaje.

b) Seguida de una sesión boilerplate LiveShare Visual Studio Code.

c) Seguido de un debate/coloquio: Bosque, thumb up or thumb down.

Parte 1 Introducción

https://www.microsoft.com/en-

us/research/uploads/prod/2019/04/beyond_structured_report_v2.pdf

Programación estructurada. Tipos de datos abstractos. Los 70.

hacía inteligente a esa máquina en particular. Con la aparición de Estructuras y Tipos florecieron los compiladores y los IDEs ya que los desarrolladores codificaban para el universo no para una máquina."

"La programación era en cada máquina, por separado. El código

La arquitectura de Von Neumann.

Programación funcional: no hay variables. Es sistema de funciones puras: inmutables. Las máquinas tienen memoria y cpu. Crecimiento exponencial en cpus, también en memoria si es distribuida. CentralPU, GraphicPU, TensorPU. Problema evolutivo: hilos simultáneos disputándose la memoria. Corrupción y demás: estructural tipada vs funcional.

Idea de que ese paradigma tiene una serie de errores.

Si se le quitan los errores queda Bosque.

The BOSQUE specification, parser, type checker, reference interpreter,

and IDE support are open-sourced and available at https://github.com/ Microsoft/BosqueLanguage.

Tema colateral: SemVer y SIMD, o de cómo enumerar la evolución y de cómo compilar mejor. Errores que subsana Bosque:

- Falta de pureza: lo puro es lo que no muta. Solución: representación pura de la información.
- Estrategias inductivas e invariantes: bucles, recursividad y contratos. Solución: fin de los bucles y poca recursividad. • Variables no inicializadas y árboles. Solución:
- determinización de las semánticas del lenguaje; órdenes de evaluación.
- Accesores y actualizadores (los actualizadores llegaron con la programación imperativa) de individuos nodos en el árbol.

Solución: operadores algebraicos de datos masivos.

• Igualdad referencial, paso por valor o por referencia, órdenes de evaluación: dependiente de la arquitectura de Von Neumann. Solución: ¿Se puede eliminar la igualdad referencial, y con ello, el resto?

Presentación del sistema de tipo

- Sistema de tipo
- Sistema de tipo nominal
- Sistema de tipo estructurado Combinación de sistemas de tipo

Presentación expresiones

- Expresiones Heredadas
- Accesores de variables global/const/local Operadores binarios y comparadores
- Accesores de datos estructurados - Reformateado de datos: copiar, fusionar, actualizar.

constructores atómicos como los de c#

- Nuevas
- Combinando los operadores spread (coge de mi objeto lo que quieras) and rest (mete los parámetros no tipados en un array args) obtenemos Call arguments. - A parte de las Tuples, records y lambda functions, unos nuevos
- Operaciones algebraicas de datos masivos: operaciones algebraicas con tipos, tuplas, registros, y tipos nominales y
- operaciones como proyección, actualización múltiple y fusión. - Gestión del Null o undefined: None. Coalescencia-nula y operador elvis. - Tuberías de colección: filter, map, reduce, etc. Pipelining
- like LINQ.

Presentación órdenes - Match, if. Operador de asignación estructurado. Assert &

var!, that can be updated.

check. Pre/post condición e invariantes. - Block SSA Dos novedades: allowing multiple assignments to a variable and

supporting statement expressions, to support functional style programming in a block-scoped

Parte 2 Sesión de código simultáneo LiveShare

language. Tipos de variables: var, that are fixed and those,

Coffee-shop finder application.

Puntos calientes:

Parte 3 Debate coloquio: Bosque thumb up

• Resistencia/facilidad de los programadores senior a mutar un paradigma ya crecido.

or down

• Posición de los junior en la cresta objetiva evolutiva a la hora de aprender axiomas y fundamentos de programación,

momento de ruptura/continuidad. • ... Propuesta: Bosque Wired Edition. Presentación demo de micro-

servicio type-wired llamado SmartPumper CGV: introducción a la

Twitter Facebook Sé el primero en decir que te gusta. mayo 9, 2019 👗 Encargado del contenido online de la nonata editorial e-

Share this:

programación enlazada.

Editar Artesanía

← Ruedo ibérico, trimestre 1 Ruedo Ethernet, trimestre 1 2019 → de 2019

Introduce aquí tu comentario...

Responder

Buscar ...

Archivos • <u>junio 2019</u> • <u>mayo 2019</u>

Sitio web ofrecido por WordPress.com.

Personalizar Editar Estadísticas ••••