

Визуальный язык и Web- среда разработки фрагментированных программ

Актуальность

- Доступность системы LuNA внешним пользователям (сложность установки окружения)
- Повышение уровня программирования на LuNA

Цели

- Разработка графического языка для записи фрагментированных алгоритмов
- Разработка WebIDE для создания и исполнения фрагментированных программ

Части проекта

1. Визуальный язык, описание алгоритмов
2. Web IDE
3. Внутреннее представление алгоритма
4. Исполнение алгоритма

1) Визуальный язык, описание алгоритмов

Задача: выбор синтаксиса и семантики

Во основе: граф информационных зависимостей

- Ветвления
- Структуры
- Массивы
- Подпрограммы
- Циклы
 - Параллельный for
 - Последовательный while

CreateFilePump

```
IRuntime runtime  
string path  
int manager  
HttpListenerResponse response  
  
returns int
```

File.Exists(path)

No

Yes

```
Stream fstream = TryOpenFile(path);
```

fstream == null

No

```
FileInfo about = new FileInfo(path);
```

```
runtime.Log.Info(  
    String.Format(  
        "Found file: {0}, Length: {1}",  
        about.Name, about.Length)  
);
```

DRAKON

Visual
Language

```
runtime.Log.Info(  
    "File not found: "  
    + path  
);
```

```
runtime.Log.Info(  
    "Could not open file: "  
    + path  
);
```

2) Web IDE

Задача: выбор средства разработки

- Выбор языка
 - JavaScript
 - TypeScript
- Выбор библиотеки
 - Выбрать и изучить средства для графических примитивов
 - Написать свою библиотеку

3) Внутреннее представление алгоритма

Задача: выбор внутреннего представления

- Синтаксис (координаты графа для сериализации)
- Семантика (для преобразования, например, в JSON)

Где хранить:

- Локально (в браузере)
- На сервере

4) Исполнение алгоритма

Задача: трансляция внутреннего представления в LuNA-программу (в JSON / в исходный код LuNA)

- Выдача LuNA-программы пользователю
- Запуск кода на локальной системе
- Отправка на сервер запуска LuNA-программ (в HPC Community Cloud)