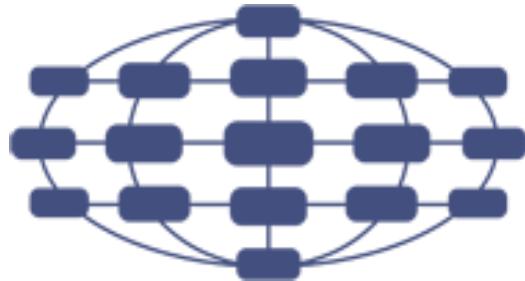


# Поддержка LuNA-6 в AZ



Руководитель: **Спирин В.А.**

Студент: **Кудюков М.Н.**

Лаборатория синтеза параллельных программ, ИВМиМГ СО РАН  
Зимняя школа по параллельному программированию, 30 января 2026 г.  
<https://ssd.sscc.ru/ru/school/2026>

# Проблема

- Концепция активных знаний позволяет автоматически применять накопленные программные модули для решения новых задач, снижая трудозатраты специалистов. Система LuNA-6 реализует эту концепцию, со временем система расширилась до автоматической генерации алгоритма в виде AZ LuNA-7, концептуально в AZ генерируется алгоритм и он же передаётся на исполнение в LuNA-6

# Цель

- Создать “мост” между AZ и LuNA-6

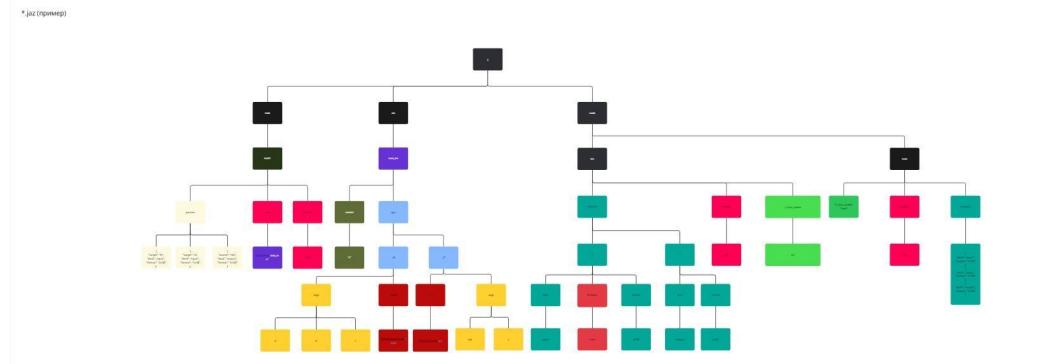
# Задачи

1. Изучение материала по AZ/LuNA-6
2. Реализовать скрипт конвертирующий файл внутреннего представления AZ (\*.jaz) в внутреннее представления LuNA-6 (\*.ja)
3. Реализовать скрипт, который создаёт заготовку для программы, используя специальный промежуточный формат фрагмента кода.
4. Доработка LuNA-6, добавление спецификатора для запуска (\*.ja) файлов

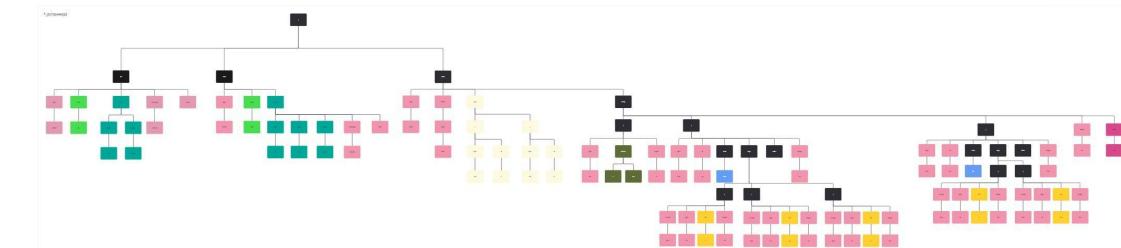
Задача  
2:

Схемы структур json файлов AZ(\*.jaz) и  
LuNA-6(\*.ja) на примере: 04\_sum\_inc(из AZ)

**AZ(\*.jaz):**



**LuNA-6(\*.ja):**

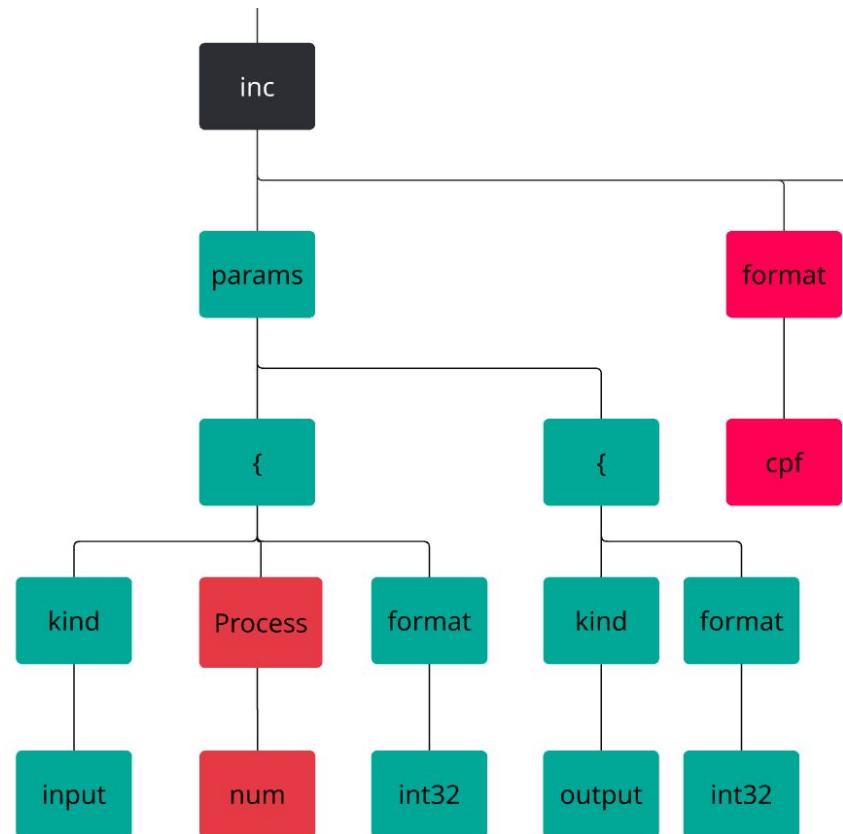


Ссылка на схему: [https://miro.com/app/board/uXjVGKdS5Qo=/?share\\_link\\_id=11746607141](https://miro.com/app/board/uXjVGKdS5Qo=/?share_link_id=11746607141) (не вечная)

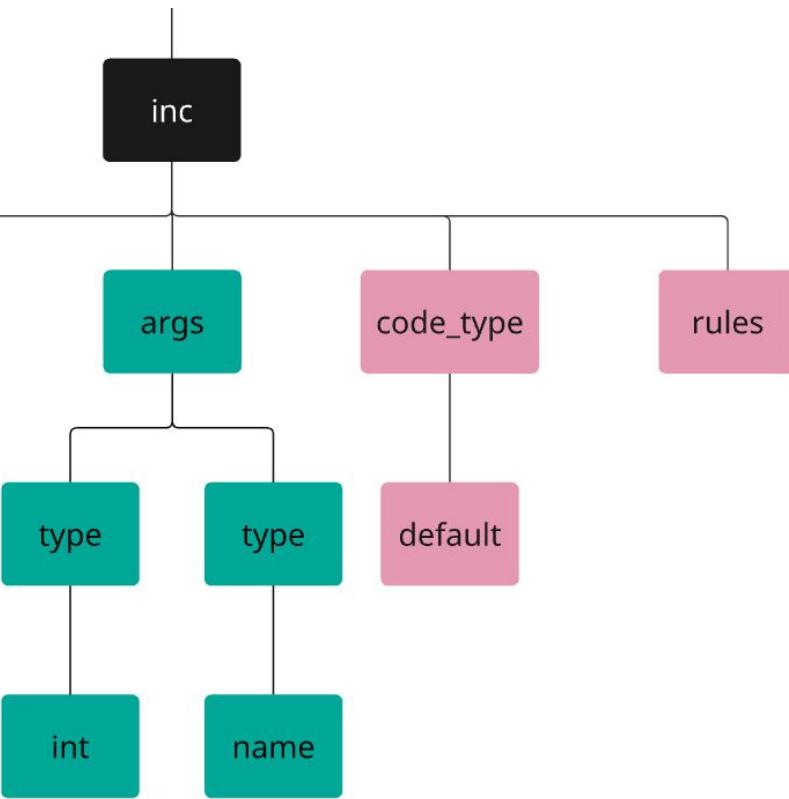
Задача  
2:

# Сравнимые части из структур AZ и LuNA-6

AZ(\*.jaz): code: inc:



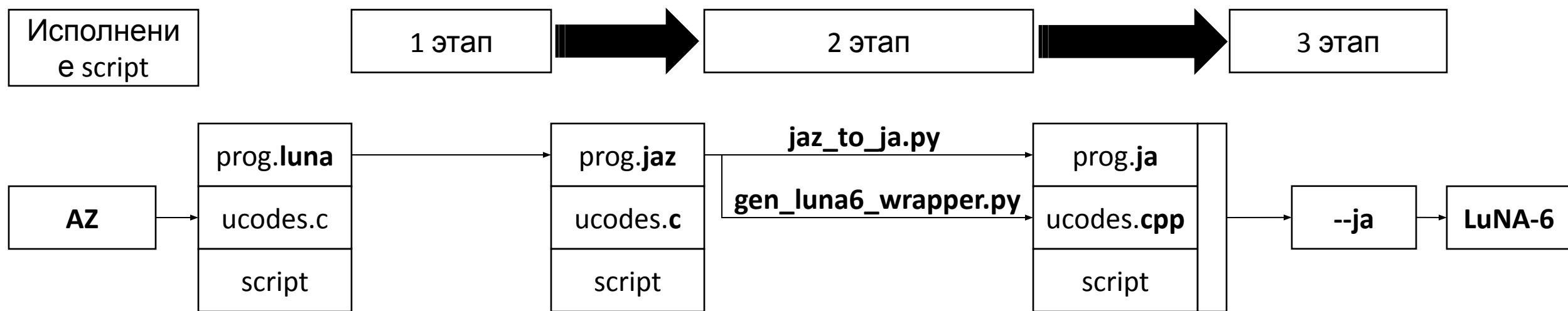
LuNA-6(\*.ja): inc:



Задача

2-4

# Пример работы скриптов задачи 2 и 3 и спецификатора 4 (схематично)



Примечание:

**jaz\_to\_ja.py** == 2 задача, скрипт конвертирующий

**gen\_luna6\_wrapper.py** == 3 задача, скрипт создающий оболочку

**--ja** == 4 задача, спецификатор в LuNA-6

Тестирование:

## Статус запущенных примеров:

- ✗ 01\_inc
- ✓ 03\_real
- ✓ 04\_sum\_inc
- ✓ 05\_sum\_inc2
- ✓ 06\_minmax
- ✗ 07\_arr
- ✗ 08\_loop
- ✗ 09\_path
- ✓ 10\_plan
- ✗ 11\_string
- ✗ 13\_corr
- ✗ 14\_web\_service
- ✗ 15\_triangle\_llm
- ✗ 16\_clif\_sh

## Причины ошибок:

- ✗ 01\_inc: \*.jaz не имеет элемента “code”
- ✗ 07\_arr: blob не реализован
- ✗ 08\_loop: не поддерживаются строки
- ✗ 09\_path: blob не реализован
- ✗ 11\_string: проблема пути(sum)
- ✗ 13\_corr: blob не реализован
- ✗ 14\_web\_service: –
- ✗ 15\_triangle\_llm: not defined p
- ✗ 16\_clif\_sh: –

# Заключение

- Сделан прототип “моста” между AZ и LuNA-6
- В следующих версиях стоит цель покрытия всех примеров и оптимизация с дальнейшим внедрением скриптов в AZ

Спасибо за  
внимание!

