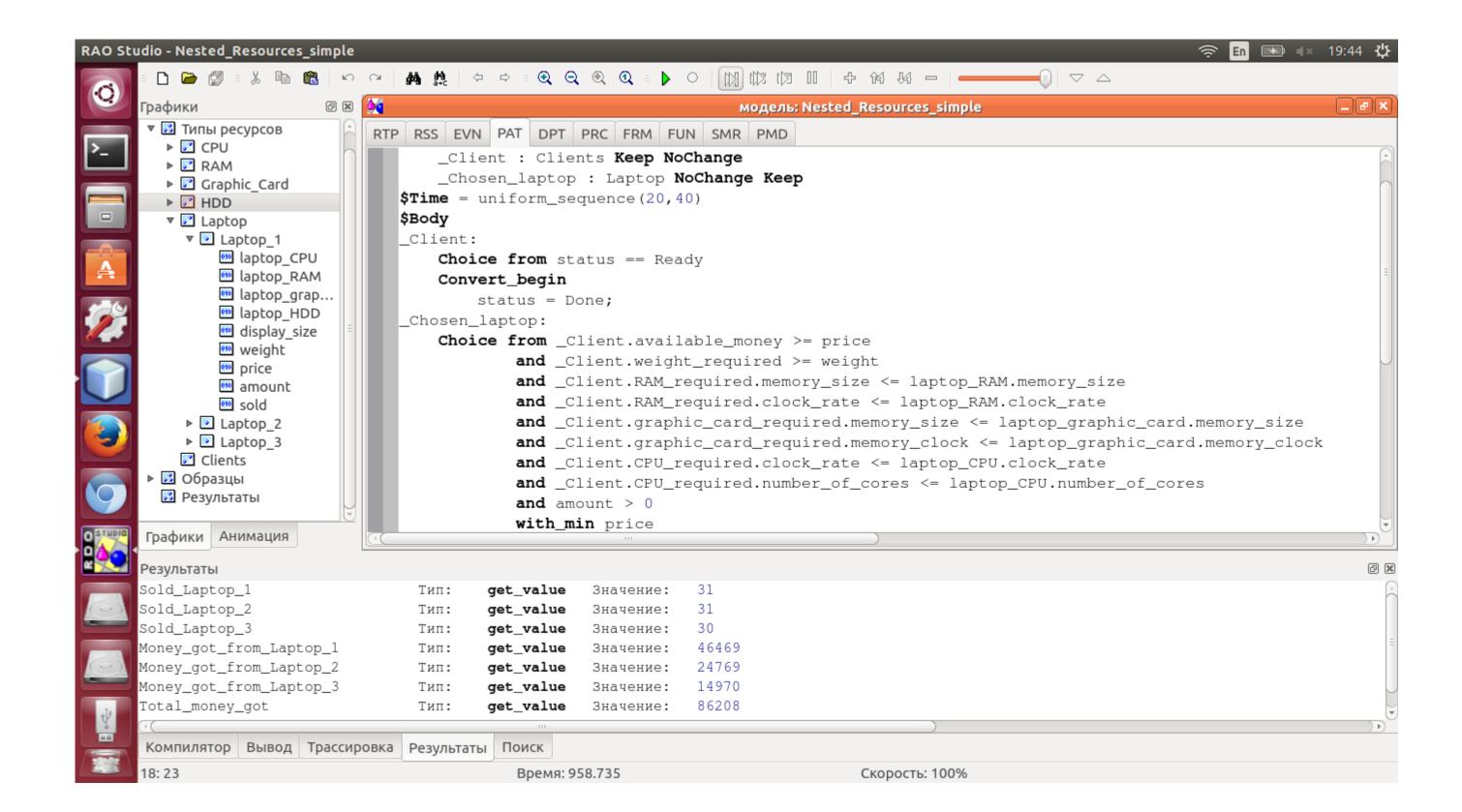
Реализация ключевых концепций

- Разработаны и внедрены конструкции создания вложенных ресурсов
- Разработаны и внедрены конструкции работы с параметрами вложенных ресурсов
- Реализована возможность работы с ресурсами неограниченного уровня вложенности

Автоматическое тестирование

- Разработана модель с вложенными ресурсами для проведения тестов
- В модели используется подбор релевантных ресурсов по параметрам различных уровней вложенности
- В модели успешно собираются различные показатели с использованием вложенных ресурсов
- Модель внедрена в систему автоматического тестирования

«Скриншот консольного вывода системы автоматического тестирования, где видно, что модель с вложенными ресурсами, проходит тесты>



Документация

 Документация по языку РДО обновлена в соответствии с нововведениями: добавлена информации о синтаксисе описания и создания вложенных ресурсов

```
🤙 🕶 🖈 ช 🔯 🔯 📦 🔍 🔍
        Вызов метода трассировки имеет следующий формат:
          <uмя_ресурса>.<признак_трассировки>();
        имя_ресурса
           Имя ресурса - это имя одного из уже описанных ресурсов, для которого вызывается метод.
        признак_трассировки
           Признак трассировки - это один из допустимых признаков трассировки (подробнее смотри <u>описание трассировки</u>).
    При использовании цепочных ссылок значение по умолчанию принимается равным ближайшему из указанных в цепочке. Поэтому в ниже рассмотренном примере последний параметр
    ресурса "Ресурс_2" при такой записи получит начальное значение "Занят", установленное для параметра "Параметр_9", а не "Свободен", установленное для параметра "Параметр_5", а
     параметр "Параметр_8" получит значение "Свободен".
 Примеры
     $Resources
             Ресурс_1 = Tun_1(0, 5.25, 100, 0.0, Занят, 20, 10., Свободен, Погрузка, Занят);
             Pecypc_2 = Tип_1(*, 5.25, *, 0.0, *, 10, 10., *, *, *);
             Pecypc_2.trace();
             Pecypc_3 = Tun_1(*, 5.25, *, 0.0, *, 10, 10., *, *, *);
             Pecypc_3.no_trace();
             Pecypc_4 = Tun_2(0, 5.25);
             Pecypc_5 = Tип_3(1, 0.0, Tип_2(0, 0.0), *);
             Pecypc_6 = Tип_4(
                     Тип_3(0, 1.0, Тип_2(0, 1.0), 5),
                     Тип_2(0, 0.0)
     $End
Содержание

    Базовые конструкции языка

        Арифметические и логические выражения
        Глобальные переменные и стандартные функции

    Процедурное программирование

       Ресурсы
        Типы данных Соотствие типов
```