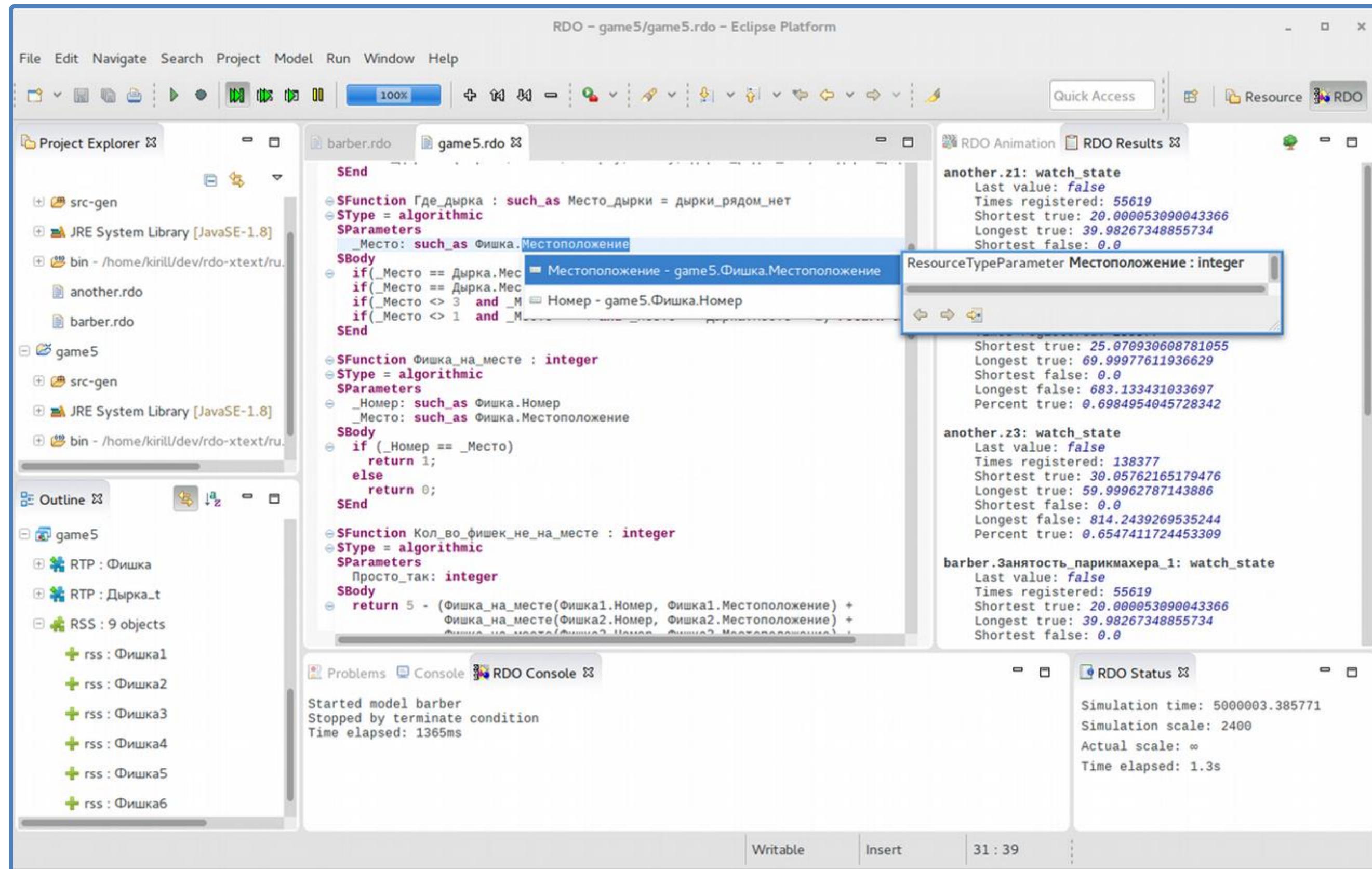


Система имитационного моделирования Rao X

Анализ существующей системы: RAO-ХТ



Преимущества по сравнению с RAO-studio

- Качественно новые возможности среды разработки
- Значительный прирост производительности
- Возможность использовать функционал языка Java

Существующие недостатки

- Интеграция с возможностями IDE Eclipse – лишь частичная
- Интеграция с возможностями языка Java – лишь частичная
- Синтаксис РДО остался прежним
- Функциональные возможности языка остались прежними

Требования к новой системе

Интеграция с Eclipse и Java

- использование произвольных классов Java
- полноценные возможности IDE:
 - подсказки и автоматическое дополнение при написании кода модели
 - автоматическое форматирование кода модели
 - подсветка синтаксиса для различных элементов модели
 - навигация по коду модели
 - рефакторинг кода модели

Новый синтаксис языка РДО

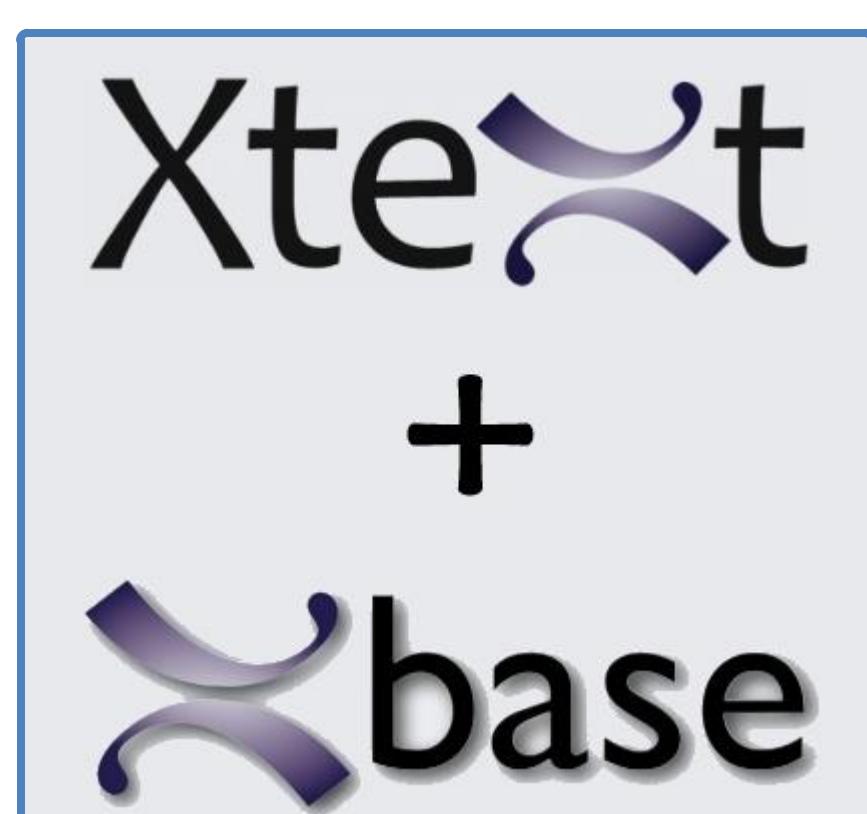
- Анализ недостатков текущего синтаксиса и разработка нового синтаксиса для существующих конструкций:
- современного и удобного
 - стилистически консистентного
 - без лишних ключевых слов

Новые возможности языка РДО

- произвольные типы во всех грамматических конструкциях модели
- произвольное число комбинаторных подборов в одном образце
- последовательности с произвольным алгоритмом генерации чисел



Использование новых технологий

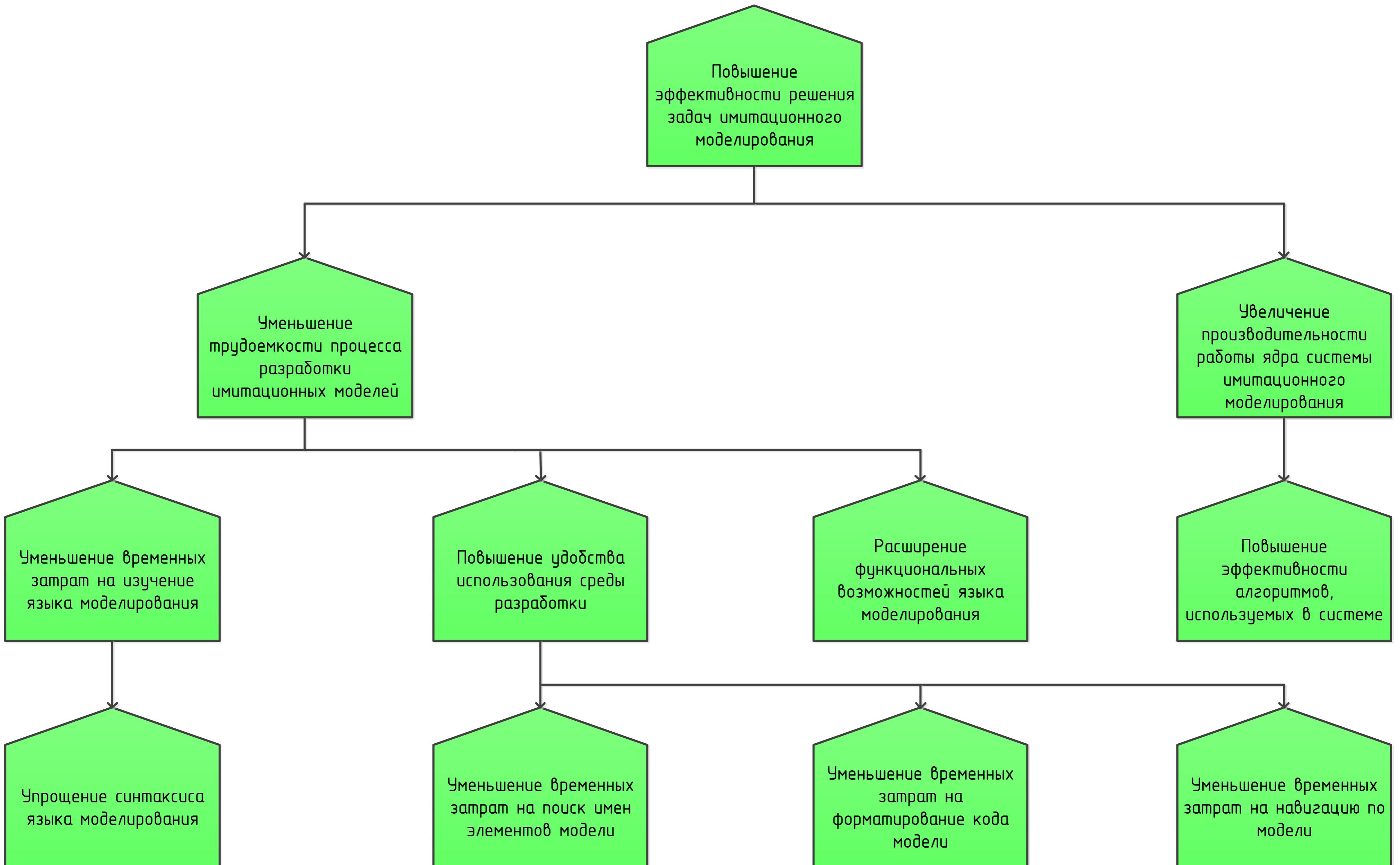


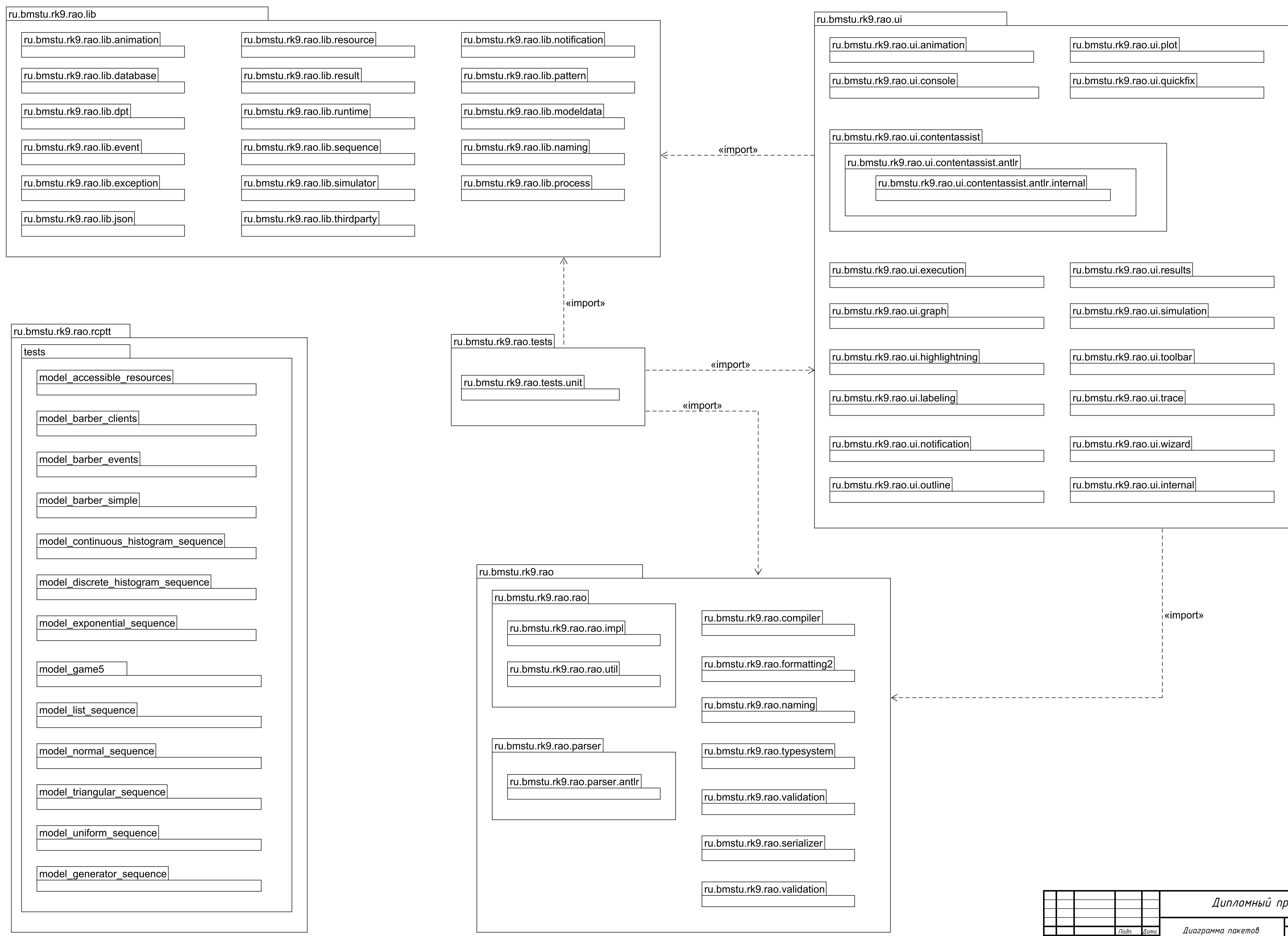
```
$Pattern Образец_поступления_звонка : event  
$Relevant_resources  
    _Служба_техподдержки: Служба_техподдержки Keep  
$Body  
    _Служба_техподдержки:  
        Convert_event  
        Образец_прихода_клиента.planning(  
            time_now + Интервал_поступления(30));  
            количество_звонков++;  
$End
```

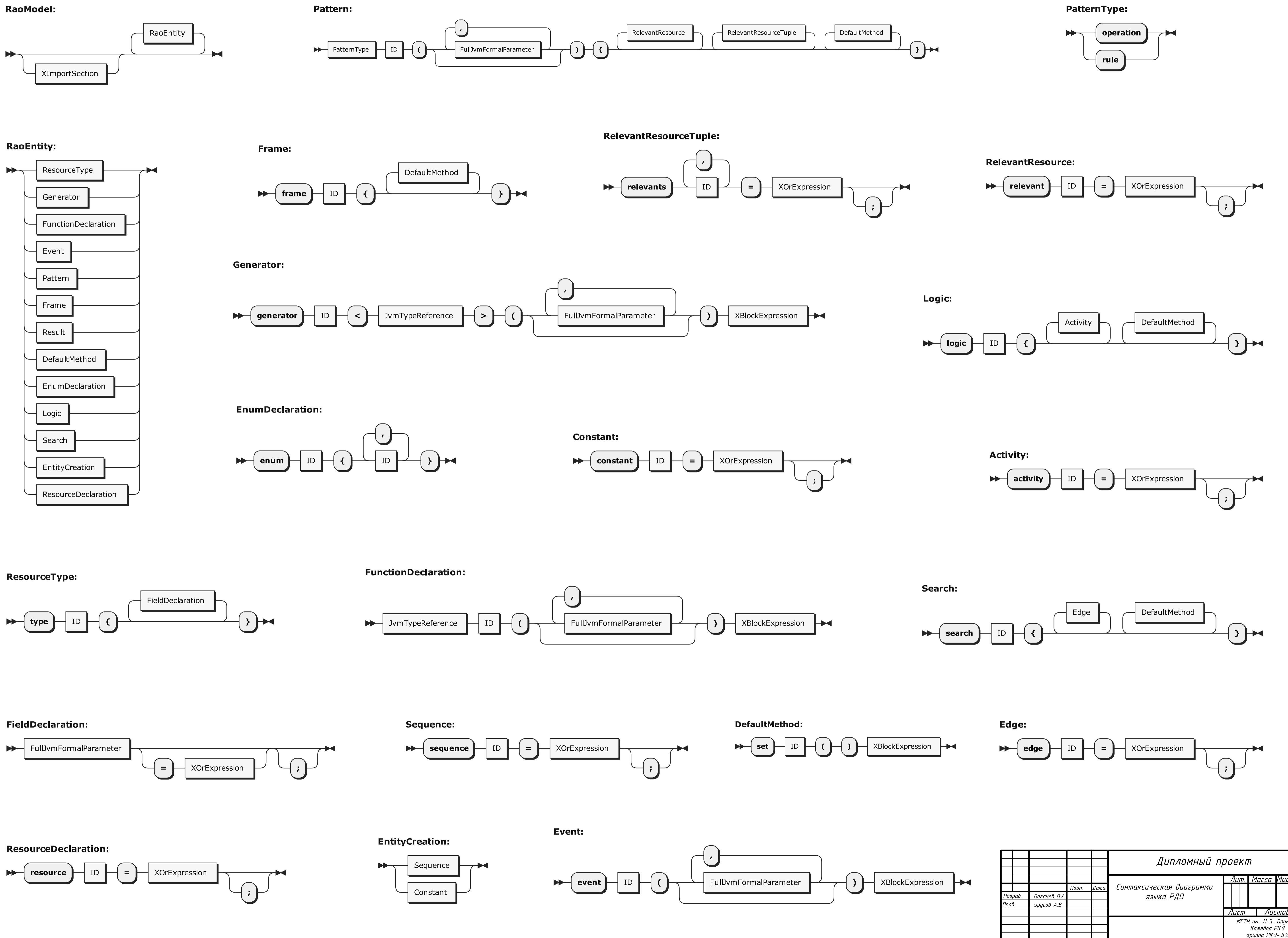
Использование механизма релевантных ресурсов даже когда ресурс не нужно подбирать

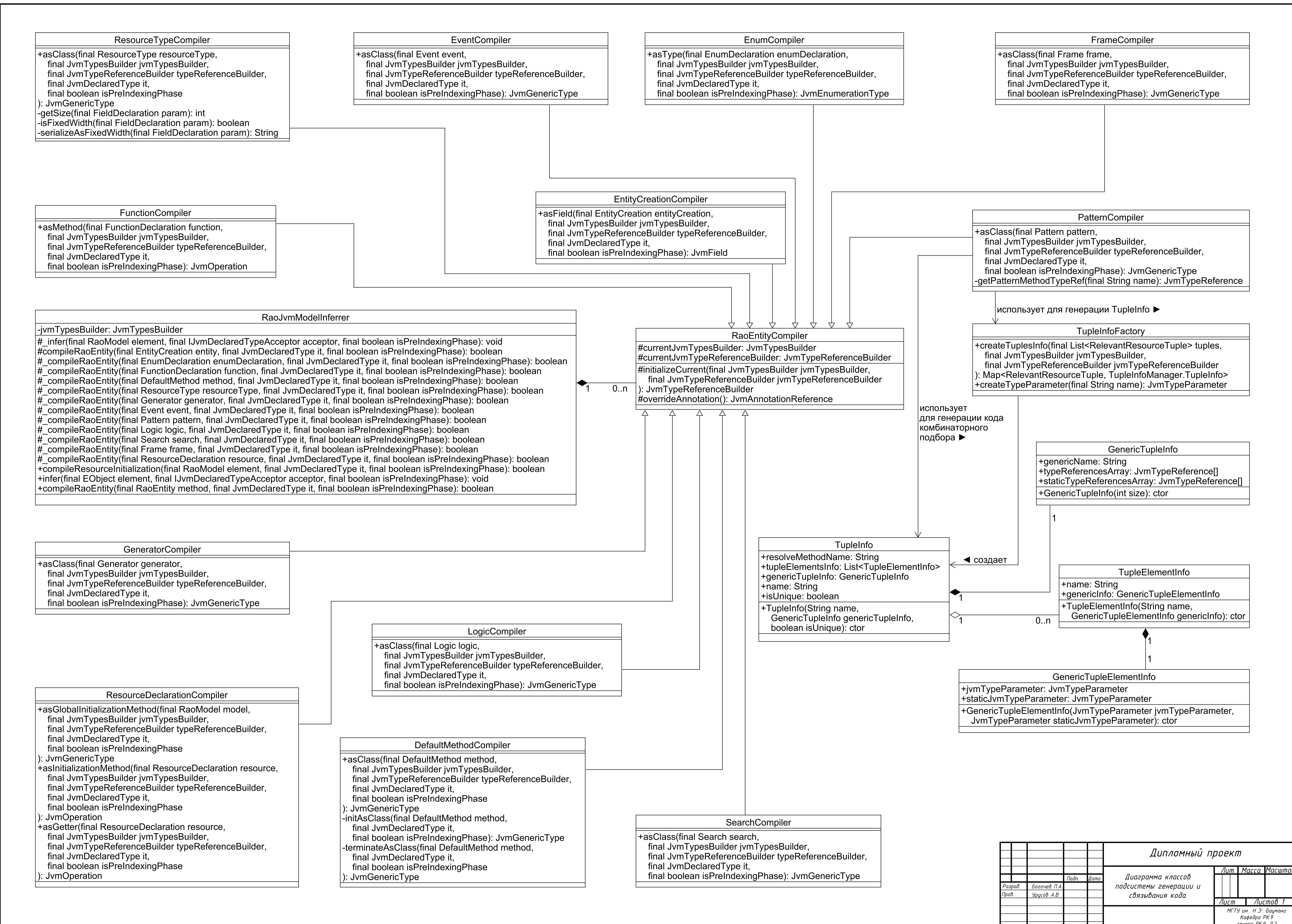
Описание тела образца отдельно для каждого релевантного ресурса

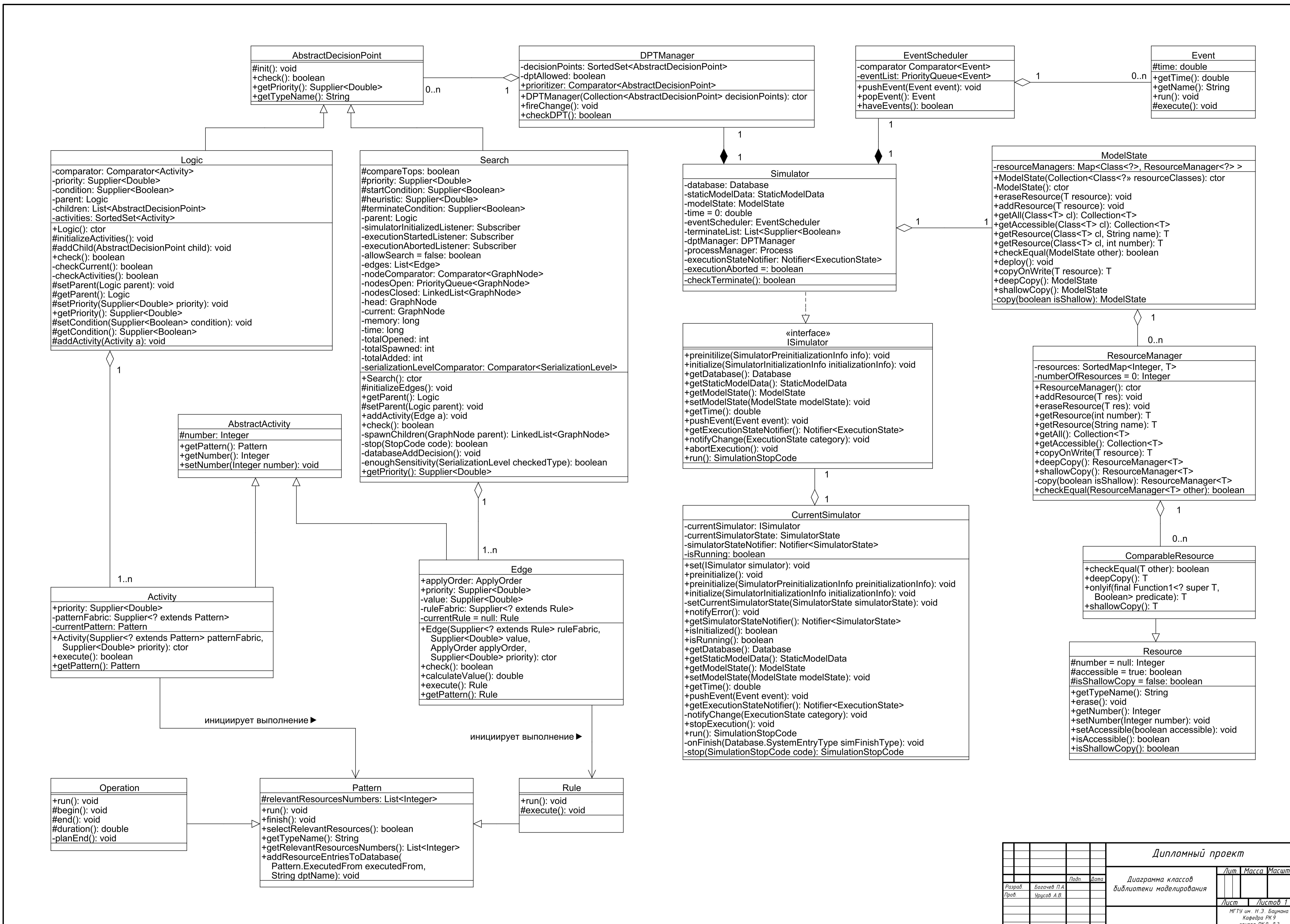
Разные стили объявления и использования переменных

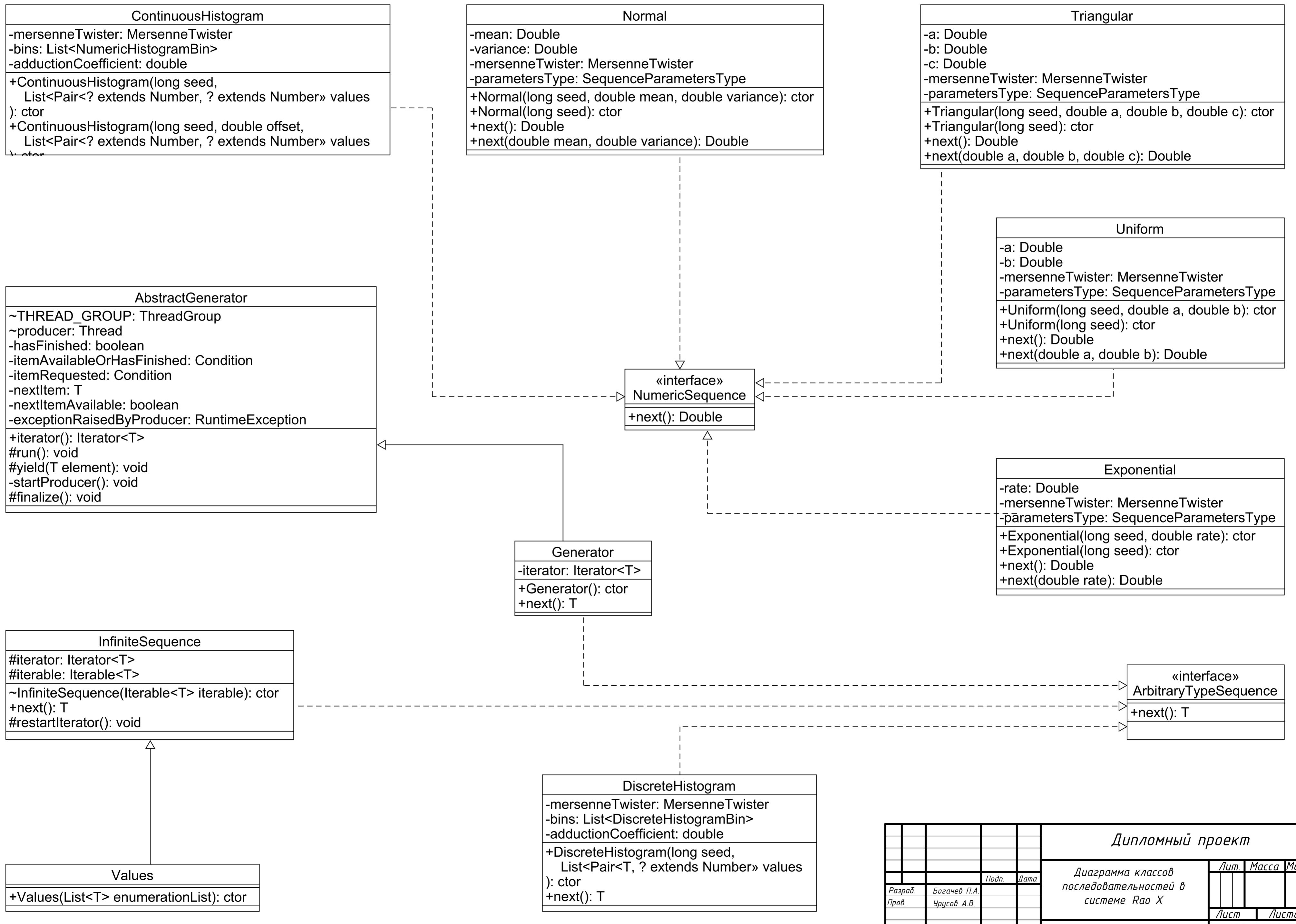




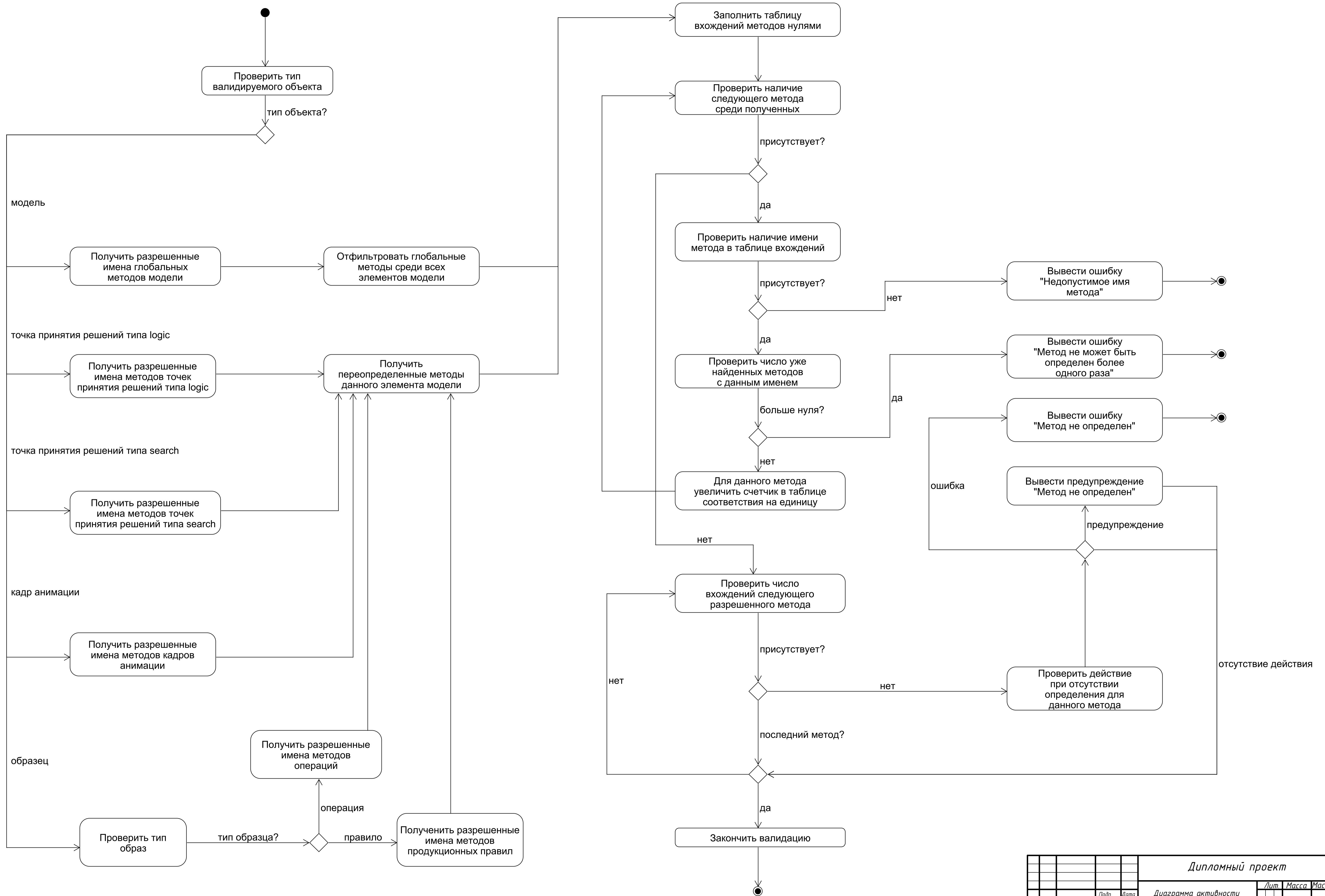


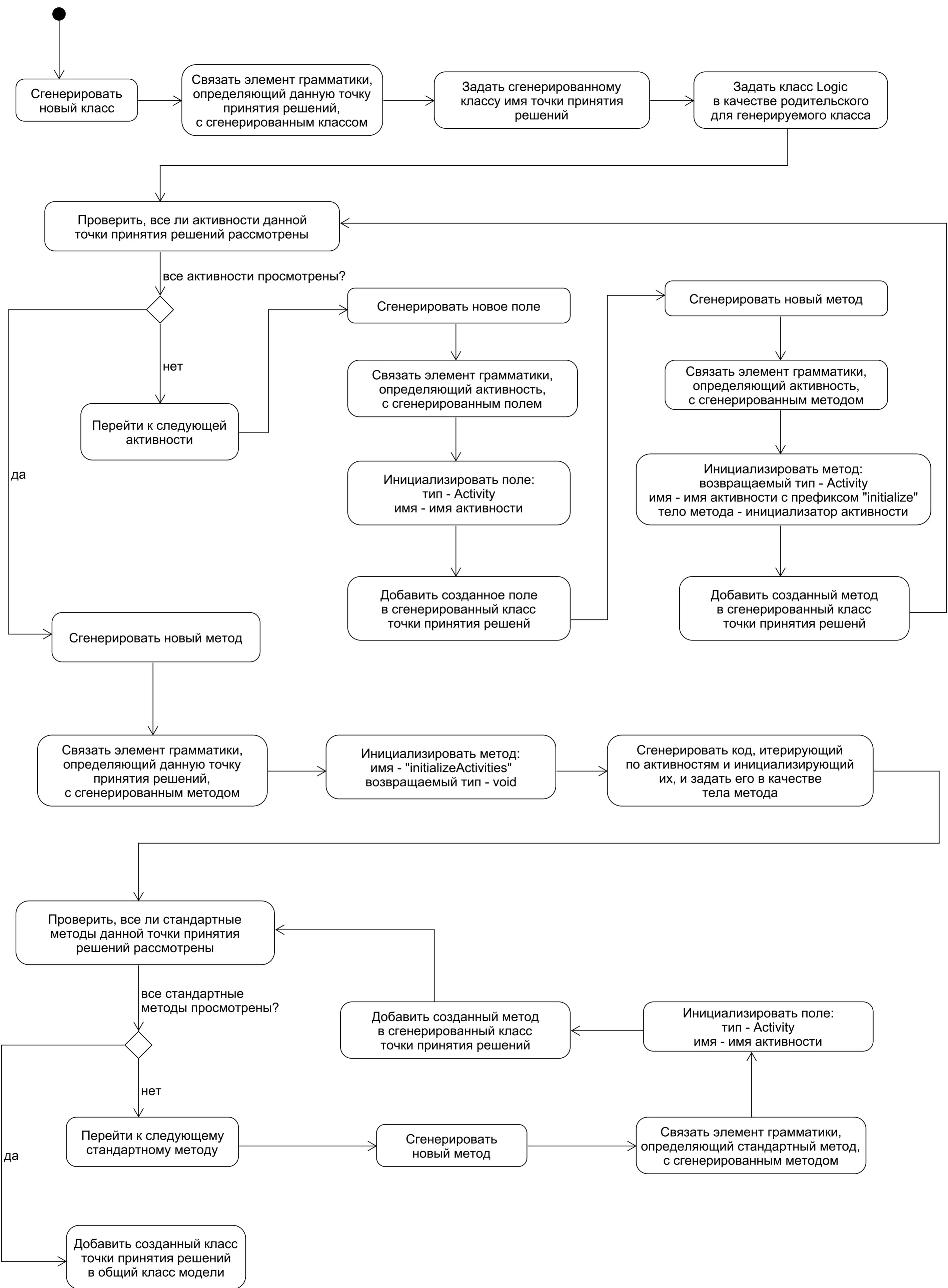


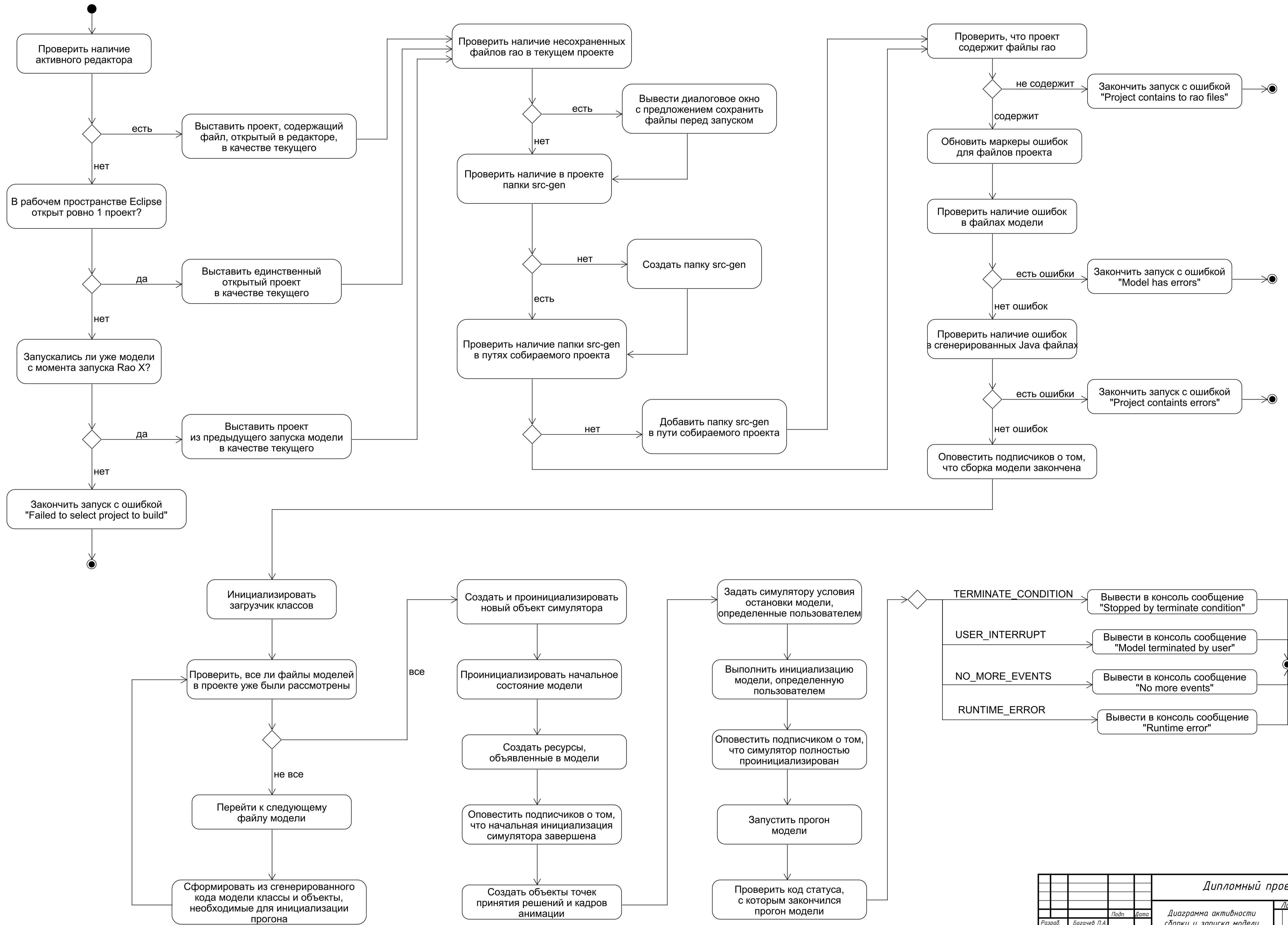




Дипломный проект

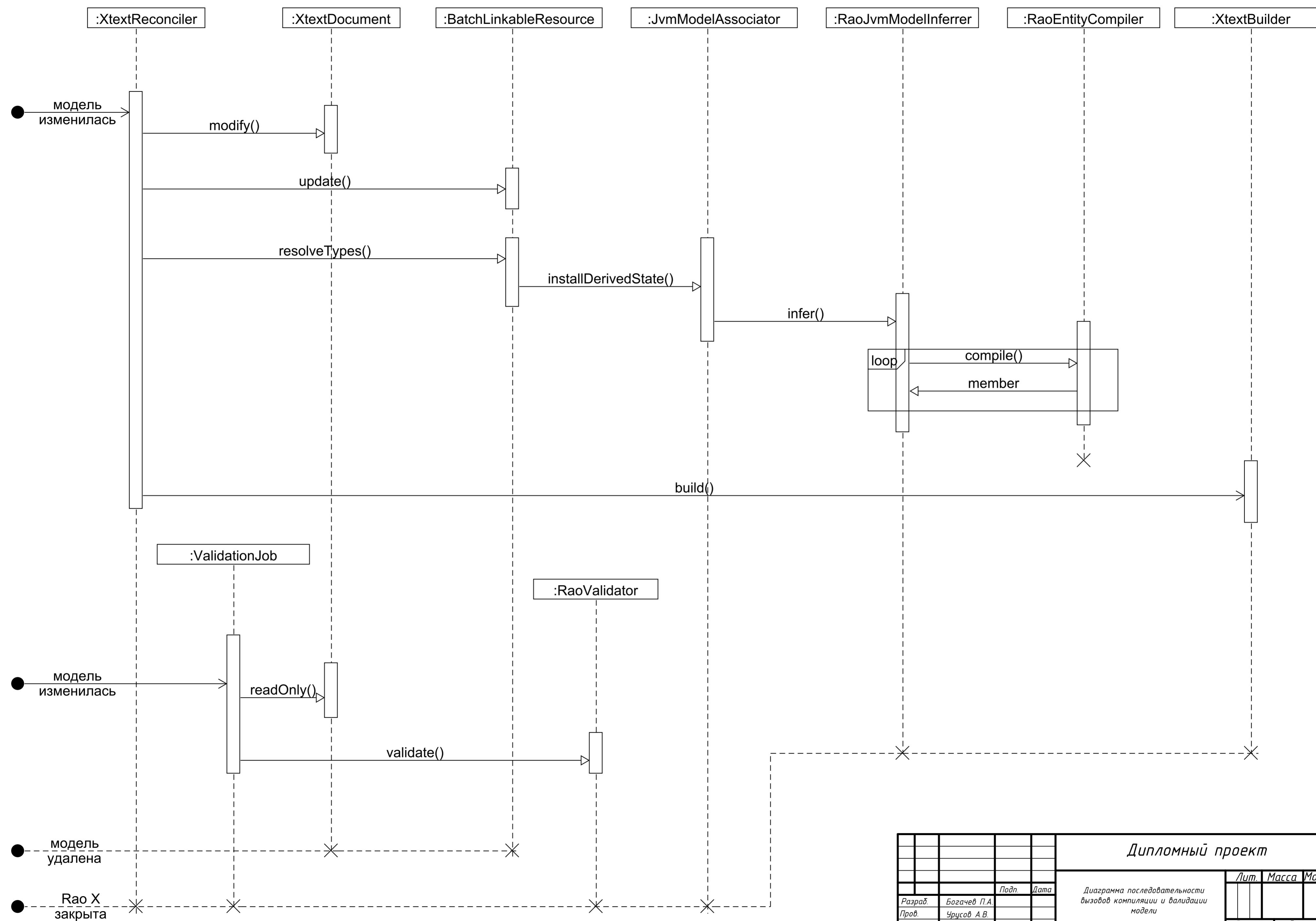






Дипломный проект					
Диаграмма активности сборки и запуска модели				Лист	Листов 1
Разраб	Богачев П.А.	Подп.	Дата		
Проф	Урусов А.В.				

МГТУ им. Н.Э. Баумана
Кафедра РК 9
группа РК 9-Д2



Дипломный проект					
			Подп.	Дата	
Разраб.	Богачев П.А.				
Пров.	Чурсов А.В.				

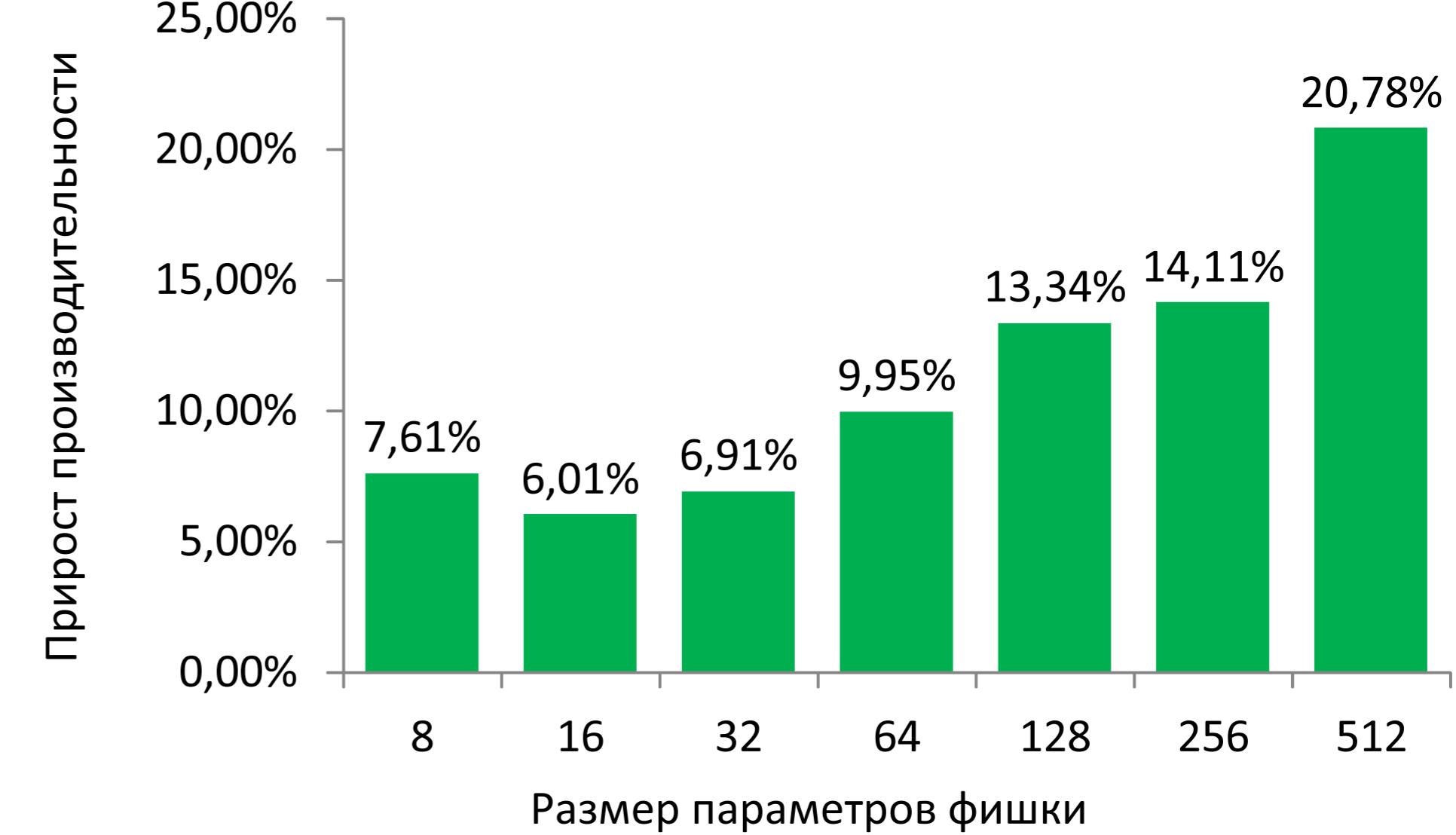
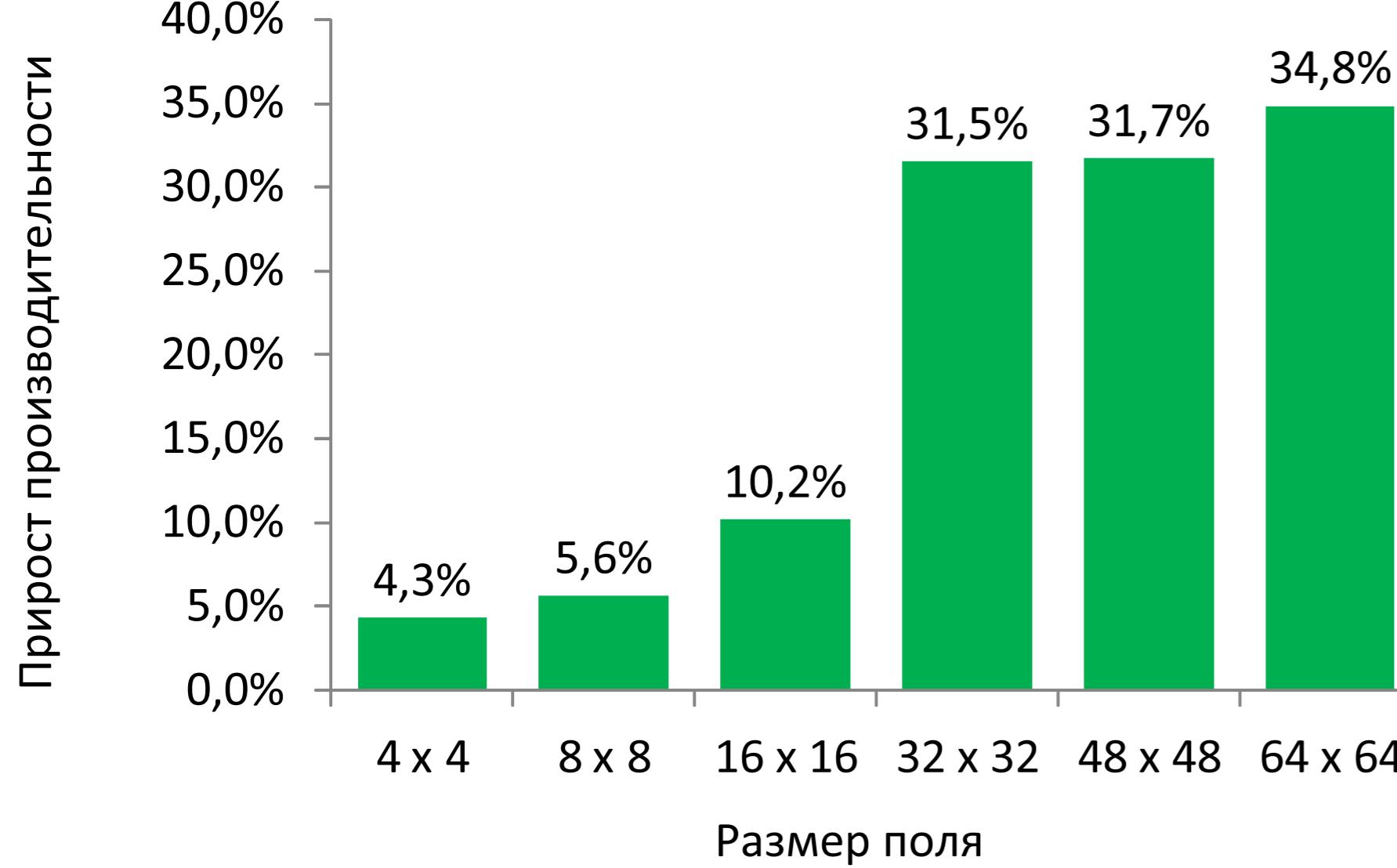
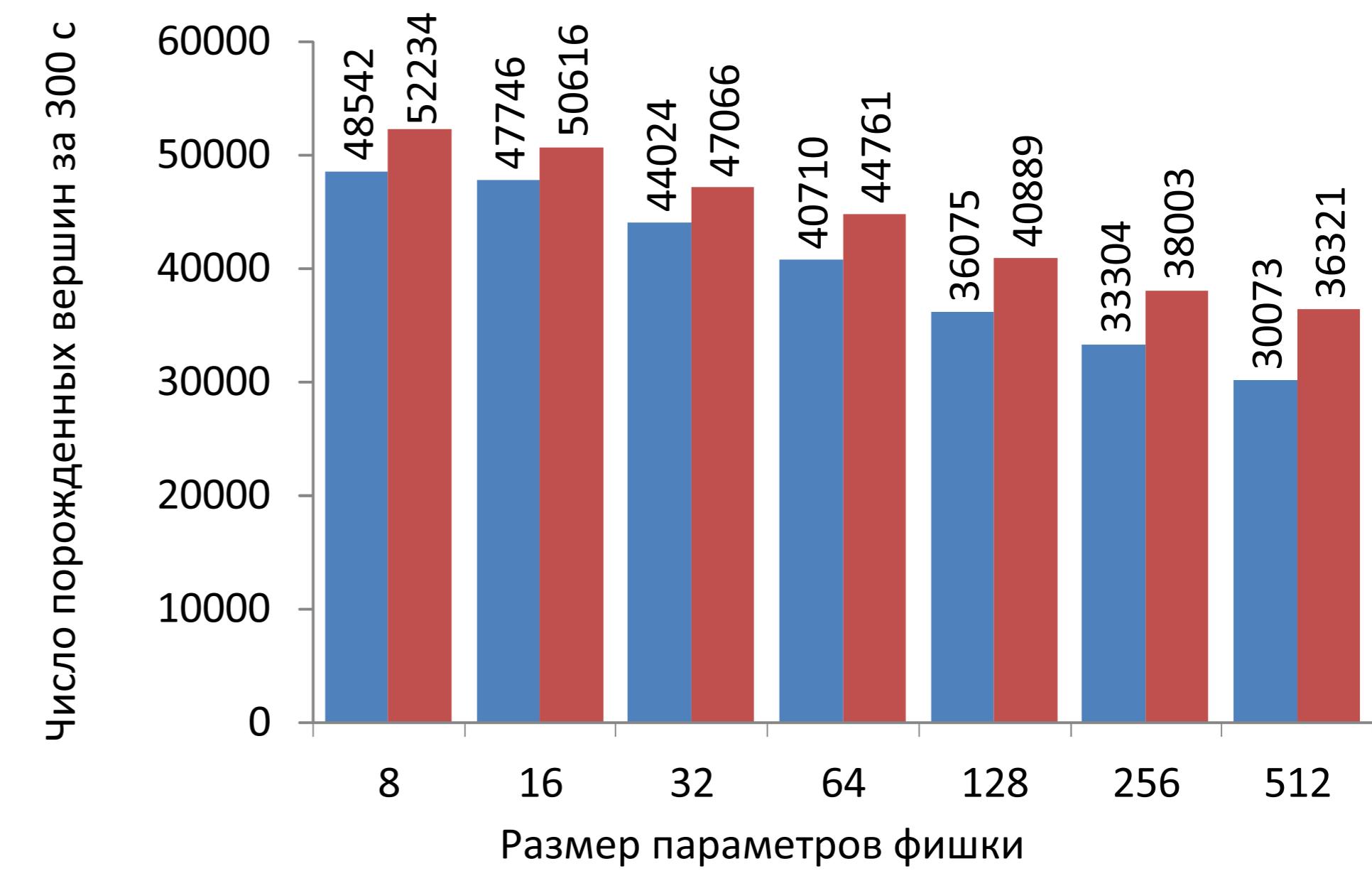
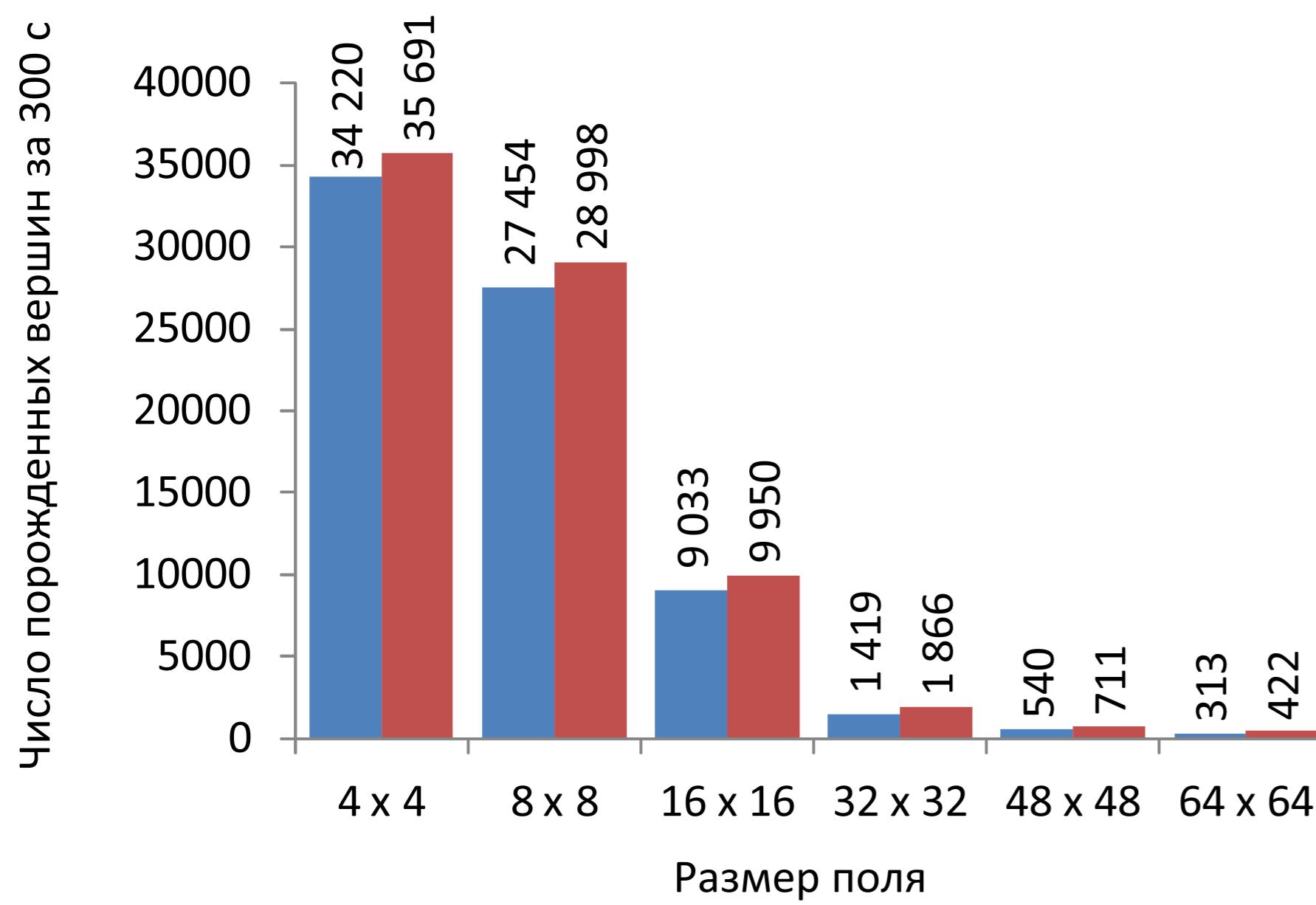
Диаграмма последовательности вызовов компиляции и валидации модели

Лист	Масса	Масштаб

Лист 1

МГТУ им. Н.Э.Баумана
Кафедра РК 9
группа РК 9-Д 2

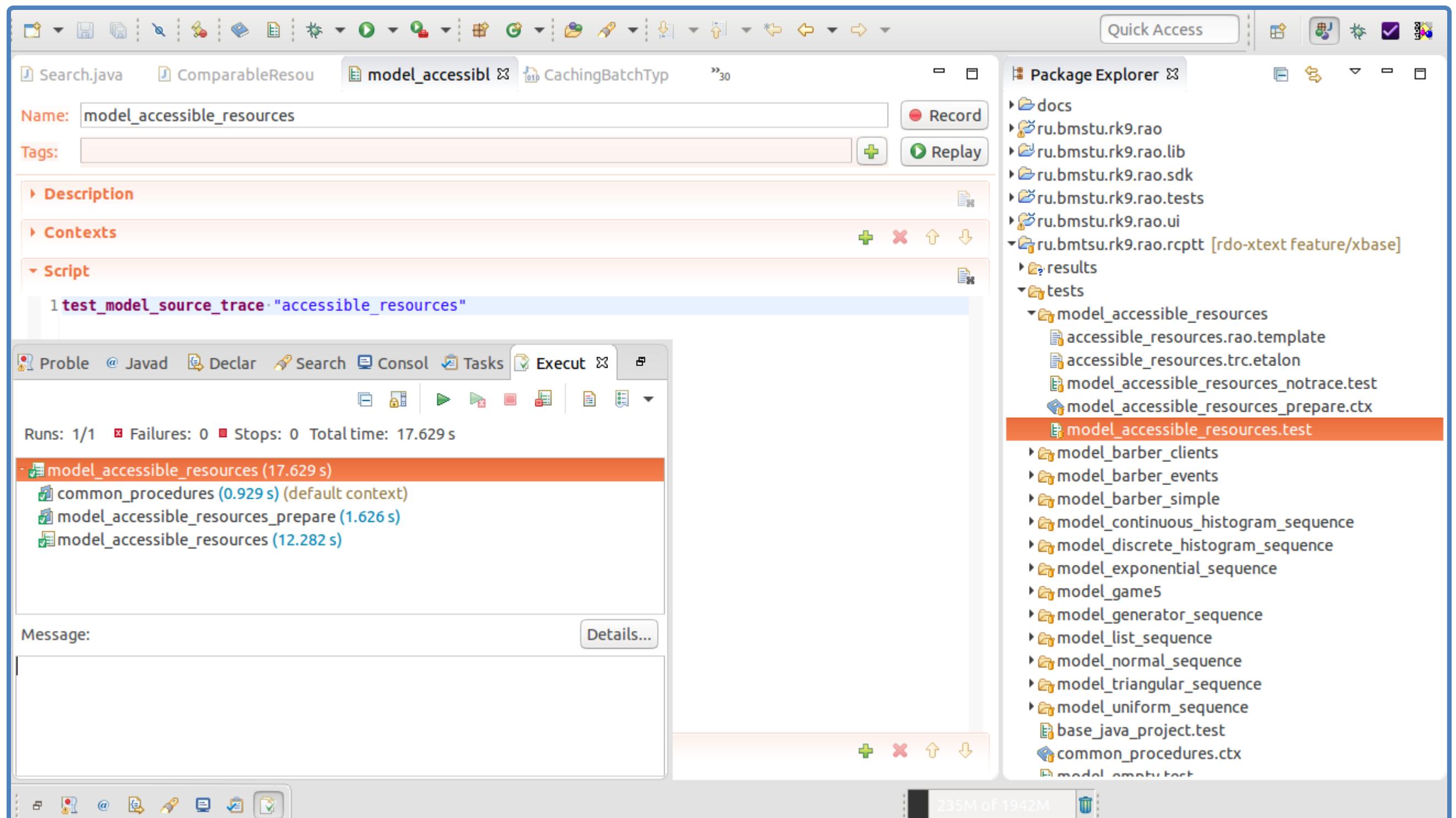
Исследование эффективности алгоритма копирования при записи



Тестирование системы

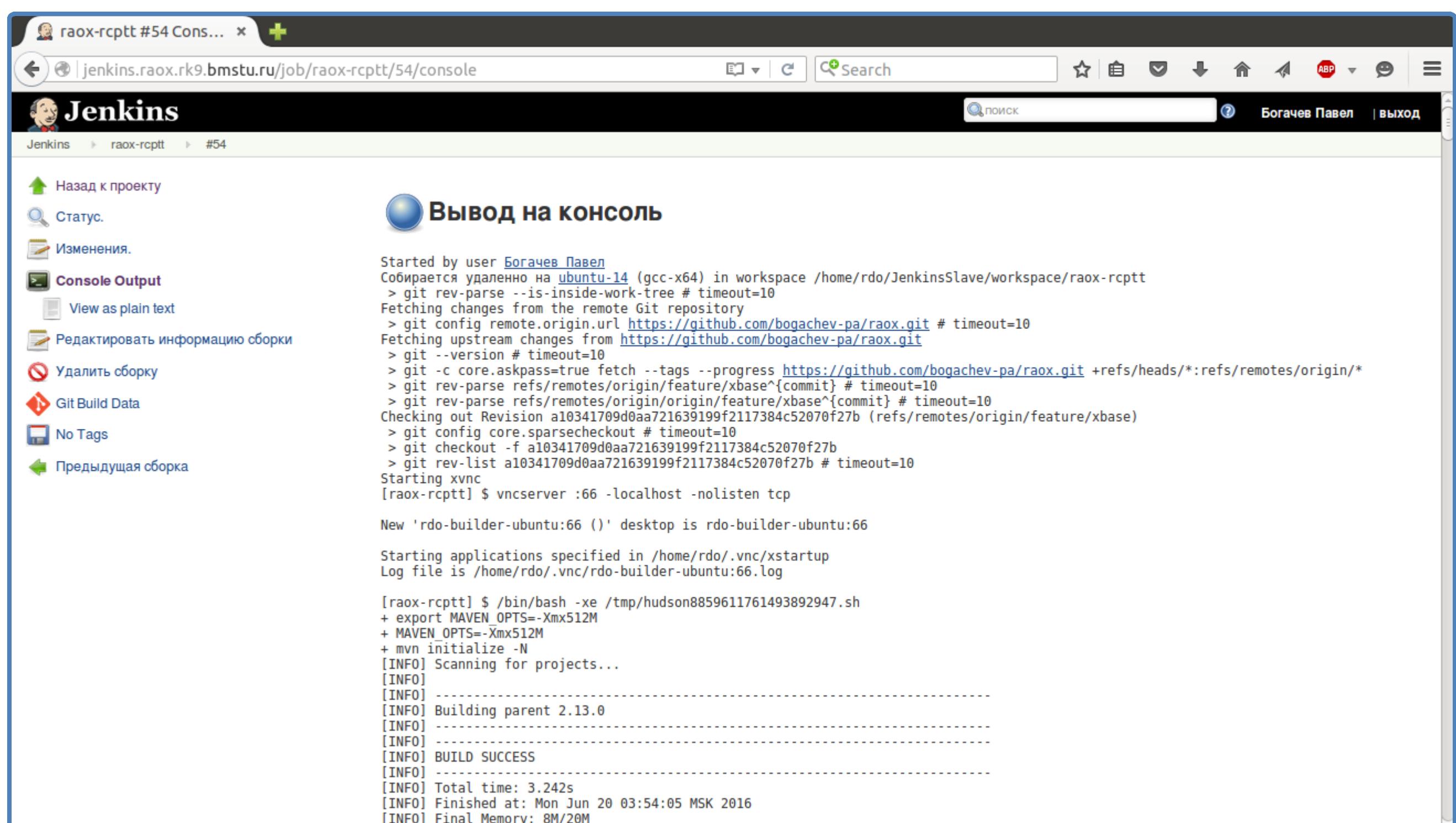
Запуск отдельных тестов в ручном режиме

- быстрый отчет
- возможность наблюдать процесс выполнения теста
- возможность быстрой отладки и доработки тестовых моделей



Автоматизированное тестирование на удаленном сервере

- автоматический запуск после обновления исходного кода в системе контроля версий
- выполнение всех существующих тестов
- подробный отчет и оповещения в случае завершения тестов с ошибкой



Функциональные возможности среды разработки моделей

Валидация ошибок

```

set duration() {
    return длительность_обслуживания.next(
        _Парикмахер.длительность_min,
        _Парикмахер.длительность_max
    );
}

set duration() {
    Error - default method cannot be set more than once
    return x + 5;
}

```

Навигация по коду модели

```

set duration() {
    return длительность_обслуживания.next(_Парикмахер.длительность_min);
}

set begin() {
    _Парикмахерская.количество_в_очереди = _Парикмахерская.количество_в_очереди - 1;
    Клиент.состояние = Состояние_клиента.Начал_стрижку;
    _Парикмахер.состояние_парикмахера = Состояние_парикмахера.Занят;
}

set end() {
    _Парикмахер.состояние_парикмахера = Состояние_парикмахера.Свободен;
    _Парикмахер.количество_обслуженных += _Парикмахер.количество_обслуженных + 1;
    Клиент.erase();
}

```

Переименование с обновлением ссылок

```

set begin() {
    _Парикмахерская.количество_в_очереди = _Парикмахерская.количество_в_очереди - 1;
    Клиент.состояние = Состояние_клиента.Начал_стрижку;
    _Парикмахер.состояние_парикмахера = Состояние_парикмахера.Занят;
} Enter new name, press Enter to refactor

set end() {
    _Парикмахер.состояние_парикмахера = Состояние_парикмахера.Свободен;
    _Парикмахер.количество_обслуженных += _Парикмахер.количество_обслуженных + 1;
    Клиент.erase();
}

```

Автоматическое форматирование

```

sequence интервал_прихода = new Exponential(123456789, 1/30.0);
sequence длительность_обслуживания = new Uniform(123456789);
sequence случайный_тип_клиента = new DiscreteHistogram<Тип_клиента>(123456789,
    #[Тип_клиента.Tип1 -> 1.0,
    Тип_клиента.Тип2 -> 5.0
    ]);

sequence интервал_прихода = new Exponential(123456789, 1/30.0);
sequence длительность_обслуживания = new Uniform(123456789);
sequence случайный_тип_клиента = new DiscreteHistogram<Тип_клиента>(123456789,
    #[
        Тип_клиента.Тип1 -> 1.0,
        Тип_клиента.Тип2 -> 5.0
    ]);

```

Подсветка синтаксиса

```

operation Образец_обслуживания_клиента() {
    relevant _Парикмахерская = парикмахерская.onlyif[
        количество_в_очереди > 0
    ];
    relevant _Клиент = Клиенты.all.filter[состояние.equals(
        Состояние_клиента.Пришел
    )].any;
    relevant _Парикмахер = Парикмахеры.all.filter[
        состояние_парикмахера.equals(Состояние_парикмахера.Свободен)
        && тип_клиента.equals(_Клиент.тип)].minBySafe[количество_обслуженных];
}

set duration() {
    return длительность_обслуживания.next(
        _Парикмахер.длительность_min,
        _Парикмахер.длительность_max
    );
}
// set duration() {

```

Автодополнение кода

```

operation Образец_обслуживания_клиента() {
    relevant _Парикмахерская = парикмахерская.onlyif[количество_в_очереди > 0];
    relevant _Клиент = Клиенты.all.filter[состояние.equals(Состояние_клиента.Пришел)].any;
    relevant _Парикмахер = Парикмахеры.all.filter[состояние_парикмахера.equals(Состояние_парикмахера.Свободен)
        && тип_клиента.equals(_Клиент.тип)].minBySafe[количество_обслуженных];
}

set duration() {
    return длительность_обслуживания.next(
        _Парикмахер.длительность_min,
        _Парикмахер.длительность_max
    );
}

set
Original declaration:
FieldDeclaration

```

Результаты работы

Система имитационного моделирования Rao X

Новый синтаксис языка РДО

```
enum СостояниеОператора {СВОБОДЕН, ЗАНЯТ}

type СлужбаТехническойПоддержки {
    СостояниеОператора состояниеОператора;
    int длинаОчереди;
    int заявокОбслужено;
}

resource службаТехническойПоддержки = СлужбаТехническойПоддержки.create(
    СостояниеОператора.СВОБОДЕН, 0, 0
);

event НоваяЗаявка() {
    службаТехническойПоддержки.длинаОчереди =
        службаТехническойПоддержки.длинаОчереди + 1;
    НоваяЗаявка.plan(currentTime + интервалПоступленияЗаявок.next());
}

operation ОбслуживаниеЗаявки() {
    relevant _службаТехническойПоддержки = службаТехническойПоддержки.onlyif[
        длинаОчереди > 0 && состояниеОператора.equals(СостояниеОператора.СВОБОДЕН)];
    set duration() {
        return времяОбслуживанияЗаявок.next();
    }

    set begin() {
        _службаТехническойПоддержки.длинаОчереди =
            _службаТехническойПоддержки.длинаОчереди - 1;
        _службаТехническойПоддержки.состояниеОператора = СостояниеОператора.ЗАНЯТ;
    }

    set end() {
        _службаТехническойПоддержки.заявокОбслужено =
            _службаТехническойПоддержки.заявокОбслужено + 1;
        _службаТехническойПоддержки.состояниеОператора = СостояниеОператора.СВОБОДЕН;
    }
}

logic Модель {
    activity callProcessing = new Activity(ОбслуживаниеЗаявки.create());
}

set init() {
    НоваяЗаявка.plan(currentTime + интервалПоступленияЗаявок.next());
}

set terminateCondition() {
    return currentTime >= 480;
}

generator ВременаОбслуживанияЗаявок<Double>(double начальноеЗначение) {
    while (true) {
        начальноеЗначение = начальноеЗначение + 2
        yield(начальноеЗначение)
    }
}

sequence интервалПоступленияЗаявок = new Exponential(123456789, 1/30.0)
sequence времяОбслуживанияЗаявок = new ВременаОбслуживанияЗаявок(20);
```

Новые возможности языка РДО

- генераторы
- гибкий комбинаторный подбор
- параметры ресурсов произвольного типа
- секция подключаемых модулей
- мощный язык выражений



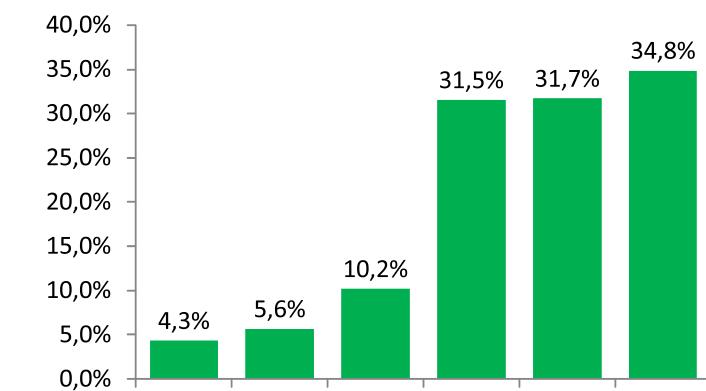
Новые возможности среды разработки

Возможности работы с кодом моделью соответствуют уровню современных сред разработки.



Более эффективные алгоритмы

Проведено исследование эффективности использования алгоритма копирования при записи при поиске на графе. Исследования показали прирост производительности до 35%.



Автоматизированное интеграционное тестирование

Разработана подсистема автоматизированного интеграционного тестирования. Подсистема запущена на удаленном сервере.

Результат тестирования последней версии системы Rao X:

Все тесты пройдены успешно.



