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
                                        Press 'F2' for focus
    return x + 5;
}
```

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();
}
```

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() {
```

Rao - clients/clients.rao - Eclipse Platform

```
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 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();
    }
}
```

```
RK 5037.04417229909 clients.clients.парикмахерская = {0}
RK 5037.04417229909 clients.clients.Клиенты[166] = {Тип2, Начал_стрижку}
RK 5037.04417229909 clients.clients.парикмахер_2 = {Занят, 68, 25, 70, Тип2}
EI 5054.261604716086 clients.clients.Событие_прихода_клиента[167]
RC 5054.261604716086 clients.clients.Клиенты[167] = {Тип2, Пришел}
RK 5054.261604716086 clients.clients.парикмахерская = {1}
ES 5054.261604716086 Simulation finished: terminate condition
```

```
Simulation time: 5054,261605
Simulation scale: 3600
Actual scale: ∞
Time elapsed: 0,2s
```

Навигация по коду модели

```
set duration() {
    return длительность_обслуживания.next(_Парикмахер.длительность_min,
}
    Uniform clients.длительность_обслуживания
    Original declaration:
        Sequence длительность_обслуживания
                                        Press 'F2' for focus
set begin()
    _Парикм                                    личество_
    _Клиент
    _Парикм                                    .Занят;
}
```

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[к
    set
    }
    set
    }
    set
    }
}
```

```
int Парикмахеры.getКоличество_обслуженных()
Original declaration:
    FieldDeclaration
                                        Press 'F2' for focus
```

```
количество_обслуженных : int - Парикмахеры.getКоличе
количество_обслуженных = value : void - Парикмахеры.se
                                        Ctrl+Space to show shortest proposals
```

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.Tun1 -> 1.0,
Тип_клиента.Tun2 -> 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.Tun1 -> 1.0,
        Тип_клиента.Tun2 -> 5.0
    ]
);
```