Г2. PSAD, Система трансляции псевдокода в диаграмму деятельности

Золин Иван[1], Кузнецов Даниил[1]

[1]группа 5030102/00201

1 Назначение и область применения

Назначение

Проект предназначен для визуализации алгоритмов, написанных на псевдокоде, в наглядные диаграммы деятельности.

Область применения

1. Образование и обучение. Система может быть использована в учебных целях для визуализации материалов по алгоритмам и программированию.
2. Программирование и разработка ПО. Разработчики ПО могут применять эту систему для визуализации алгоритмов перед реализацией, что помогает им лучше понять структуру и логику кода.

2 Цели и задачи

1. Предложить процесс трансляции псевдокода в диаграмму деятельности.
2. Определить основные сущности, которые будут представлены на диаграмме деятельности. Согласовать выбранные сущности с командой DiaDel.
3. Согласовать с командой Псевдокода грамматику псевдокода.
4. Написать программу на языке DiaDel с описанием отображения сущностей диаграммы в фигуры.
5. Сделать транслятор AST псевдокода в экземпляр семантической модели.
6. Проверить работоспособность всей системы(PSAD+DiaDel) на реальных примерах.

3 Программа и методика испытаний.

1. Тестирование транслятора псевдокода в AST.
   - **Задача**: Проверить, что транслятор корректно преобразует псевдокод в абстрактное синтаксическое дерево (AST), отражая все сущности диаграммы.
   - **Методика**:
     (a) Создать набор тестовых алгоритмов на псевдокоде с различными сущностями.

(b) Запустить транслятор и проверить на каждом примере результаты.

(c) Убедиться, что недочётов нет. При наличии исправить.

2. Тестирование парсера AST в объекты семантической модели.

   - **Задача**: Убедиться, что парсер правильно интерпретирует AST и преобразует его в соответствующие объекты семантической модели,
   - **Методика**:
     (a) Создать набор тестовых AST, полученных из разлчиных алгоритмов на псевдокоде.
     (b) Запустить парсер на каждом AST и проверить, что каждый узел соотвествует своему объекту семантической модели.
     (c) Убедиться, что недочётов нет. При наличии исправить.

3. Тестирование системы с использованием языка DiaDel

   - **Задача**: Убедиться, что система успешно визуализирует псевдокод в диаграмму деятельности, используя систему PSAD и язык DiaDel.
   - **Методика**:
     (a) Создать набор тестовых алгоритмов на псевдокоде с различными сущностями.
     (b) Запустить систему и проверить, что каждый псевдокод корректно отображается в диаграмму.
     (c) Добавить несколько новых сущностей из псевдокода.
     (d) Протестировать на новом наборе алгоритмов.
     (e) Убедиться, что недочётов нет. При наличии исправить.

4. Тестирование системы на реальных примерах использования

   - **Задача**: Проверить работоспособность системы на реальных практических задачах, которые могут встретиться пользователям.
   - **Методика**:
     (a) Собрать набор реальных алгоритмов на псевдокоде из учебных пособий, книг, онлайн-ресурсов и практических задач.
     (b) Протестировать систему, удостоверившись, что получаются корректные диаграммы деятельности.

4 Сущности диаграммы деятельности

1. Точка входа (начальное состояние) - чёрный круг
2. Точка выхода (конечное состояние) - чёрный круг с белым кольцом

3. Узел условия - белый ромб
4. Узел слияния - белый ромб
5. Действие - прямоугольник с круглыми углами
6. Переход (ребро действия) - стрелка с надписью (в квадратных скобках)
7. Комментарий - значок файла
8. Аннотация - пунктирная линия с кругом на конце

5 Пример трансляции псевдокода в диаграмму деятельности

5.1 DiaDel программа

```
start = BC360
action = CR4
decision = WRH4
merge = WRH4
=> = A2
comment = WF5
—> = DA2
end = BWC360
```

5.2 Примеры

5.2.1 Пример 1

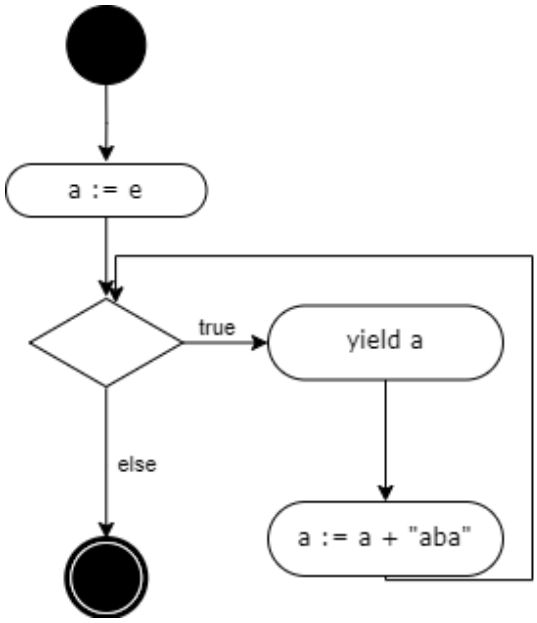
1. Псевдокод

```
a := e
while true do
    yield a
    a := a + "aba"
end while
```

2. Экземпляр семантической модели

```
start{id=1} => action{id=2, text="a :=e"}
action{id=2, text="a :=e"} => decision{id=3}
decision{id=3} =>{true} action{id=4, text="yield a"}
action{id=4, text="yield a"} => action{id=5, text =
"a := a + 'aba'"}
action{id=5, text = "a := a + 'aba'"} => decision{id=3}
decision{id=3} =>{else} end{id=6}
```

3. Диаграмма деятельности



5.2.2 Пример 2

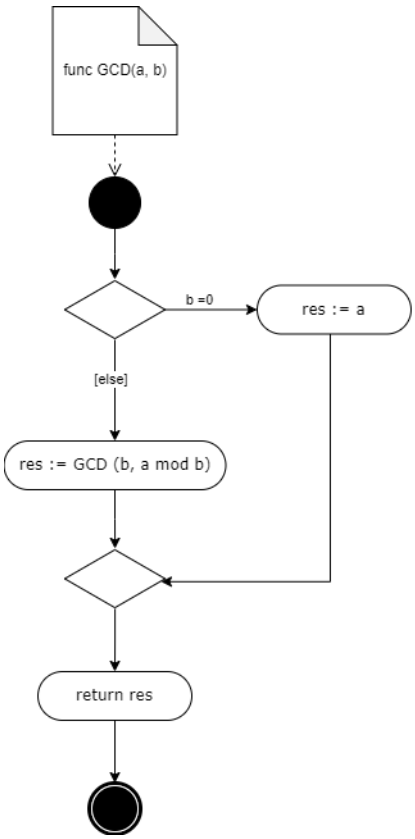
1. Псевдокод

```
func GCD(a, b)
  if b = 0 then
    res := a
  else
    res := GCD(b, a mod b)
  return res
```

2. Экземпляр семантической модели

```
comment{id=1, text="func GCD(a, b)"} —> start{id=2}
start{id=2} => decision{id=3}
decision{id=3} =>{b=0} action{id=4, text="res := a"}
decision{id=3} => {else} action{id=5, text="res := GCD
(b, a mod b)"}
action{id=4, text="res := a"} => merge{id=6}
action{id=5, text="res := GCD (b, a mod b)"} => merge
{id=6}
merge{id=6} => action{id=7, text="return res"}
action{id=7, text="return res"} => end{id=8}
```

3. Диаграмма деятельности



5.2.3 Пример 3

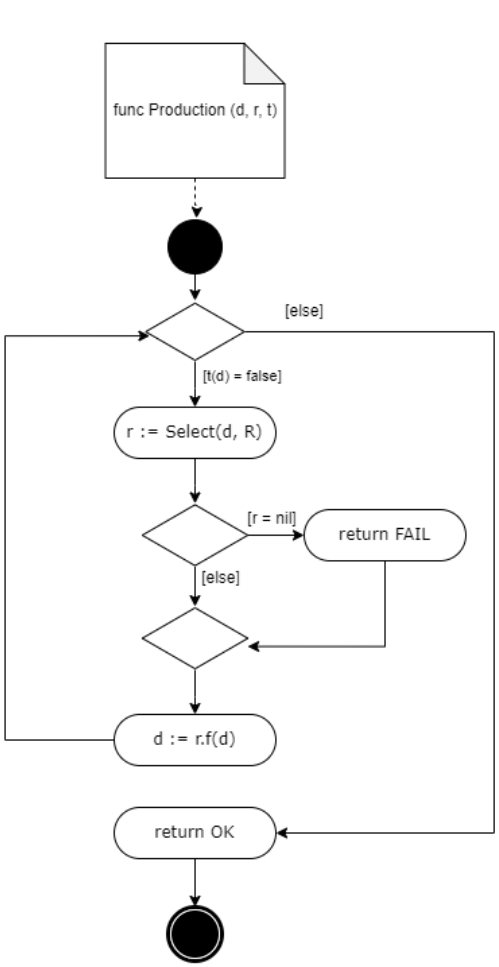
1. Псевдокод

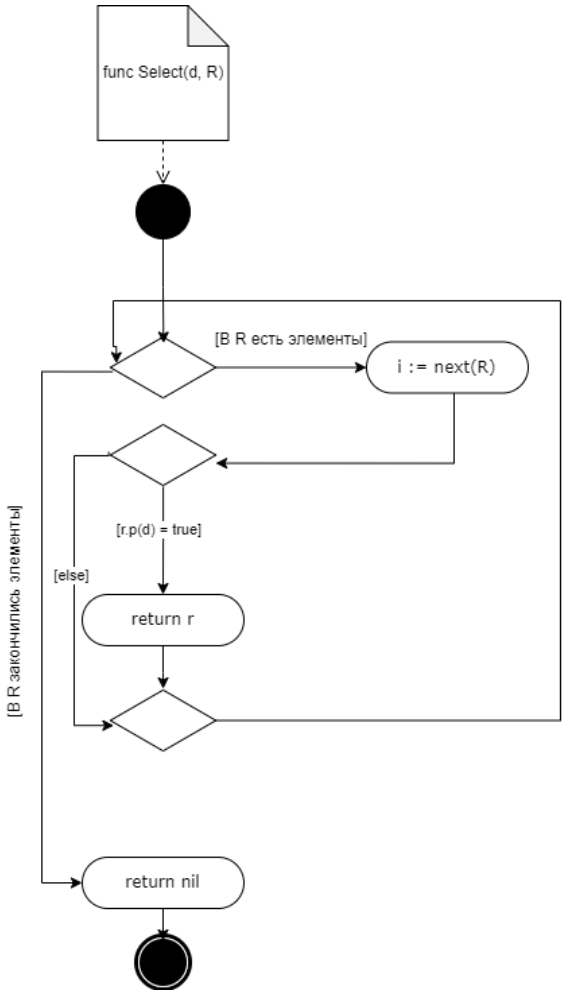
```
func Select (d, R)
    for i in R do
        if r.p (d) = true then
            return r
        end if
    end for
    return nil
end func

func Production (d, R, t)
    while t(d) = false
        r := Select(d, R)
        if r = nil then
            return FAIL
        end if
        d := r.f(d)
    end while
    return OK
end func
```

2. Экземпляр семантической модели

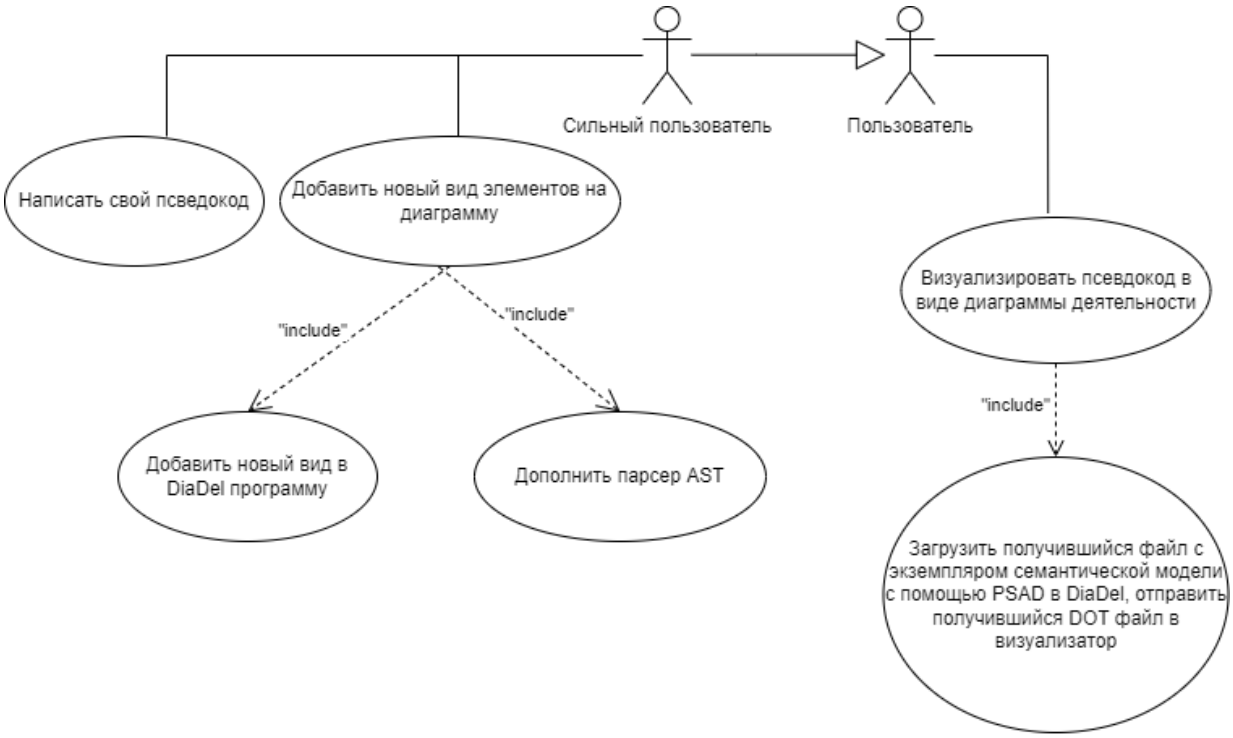
```
comment{id=1, text="func Select (d, R)"} —> start{id=2}
start{id=2} => decision{id=3}
decision{id=3} =–>{There are elements in R} action{id=4,
decision{id=3} =>{R has run out of elements} action{id=5
action{id=4, text="i := next(R)"} => decision{id=6}
decision{id=6} =>{r.p(d) = true} action{id=7,
text="return r"}
decision{id=6} =>{else} merge{id=8}
action{id=7, text="return r"} => merge{id=8}
merge{id=8} => decision{id=3}
action{id=5, text="return nil"} => end{id=9}

comment{id=10, text="func Production (d, R, t)"} —>
start{id=11}
start{id=11} => decision{id=12}
decision{id=12} =>{b(d) = false} action{id=13, text="r
:= Select(d, R)"}
decision{id=12} =>{else} action{id=14, text="return OK"}
action{id=13, text="r := Select(d, R)"} =>
decision{id=15}
decision{id=15} =>{r = nil} action{id=16,
text="return FAIL"}
decision{id=15} =>{else} merge{id=17}
action{id=16, text="return FAIL"} => merge{id=17}
merge{id=17} => action{id=18, text="d := r.f(d)"}
action{id=18, text="d := r.f(d)"} => decision{id=12}
action{id=14, text="return OK"} => end{id=19}
```

3. Диаграмма деятельности

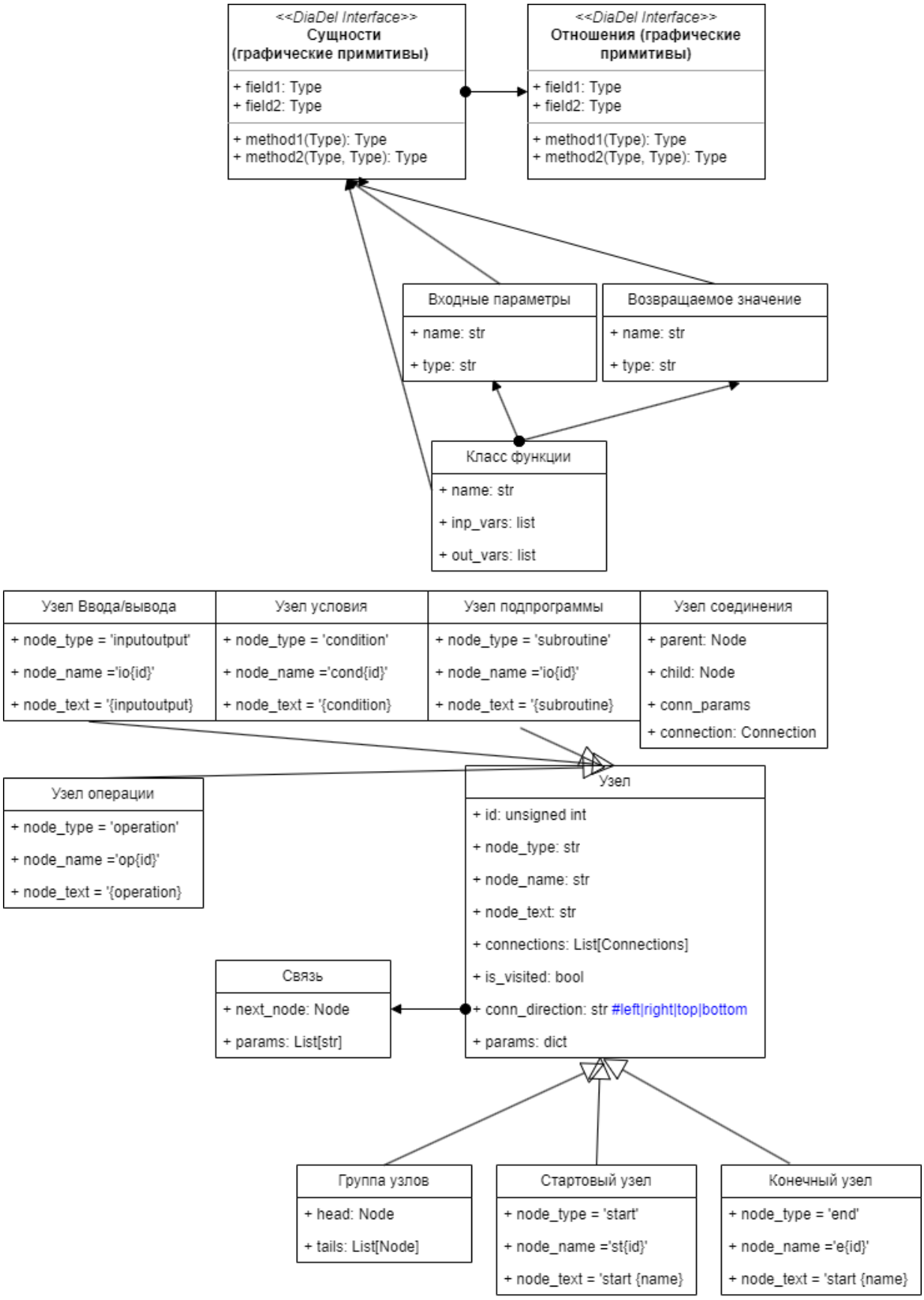




6 Диаграмма использования



7 Метамодель



8 Понедельный план-график

01.03:

- Разобрать постановку задачи
- Написать одностранчиное описание
- Написать диаграмму использования UML
- Создать метамодель
- Согласовать описание, диаграммы использования, метамодели и планов с Новиковым Ф.А.

15.03:

- Согласовать синтаксис псевдокода.
- Определить основные сущности, которые будут представлены на диаграмме деятельности.
- Изучить инструмент Воротникова.
- Начать разрабатывать транслятор абстрактное синтаксическое дерево (AST) в экземпляр семантической модели.

22.03:

- Написать инструмент транслирования псевдокода в AST, используя грамматику Псевдокода
- Согласовать написанную программу на яызке DiaDel с описанием отображения сущностей диаграммы в фигуры.

29.03:

- Написать транслятор AST в экземпляр семантической модели.

05.04:

- Протестировать всю систему (PSAD + DiaDel)
- Проверить отрисовку диаграмм с помощью языка DiaDel

12.04:

- Оформить финальный отчёт
- Защитить проект

Github репозиторий: `https://github.com/IMZolin/PSAD`