

Программное обеспечение автоматизации компоновки модулей ЭВА

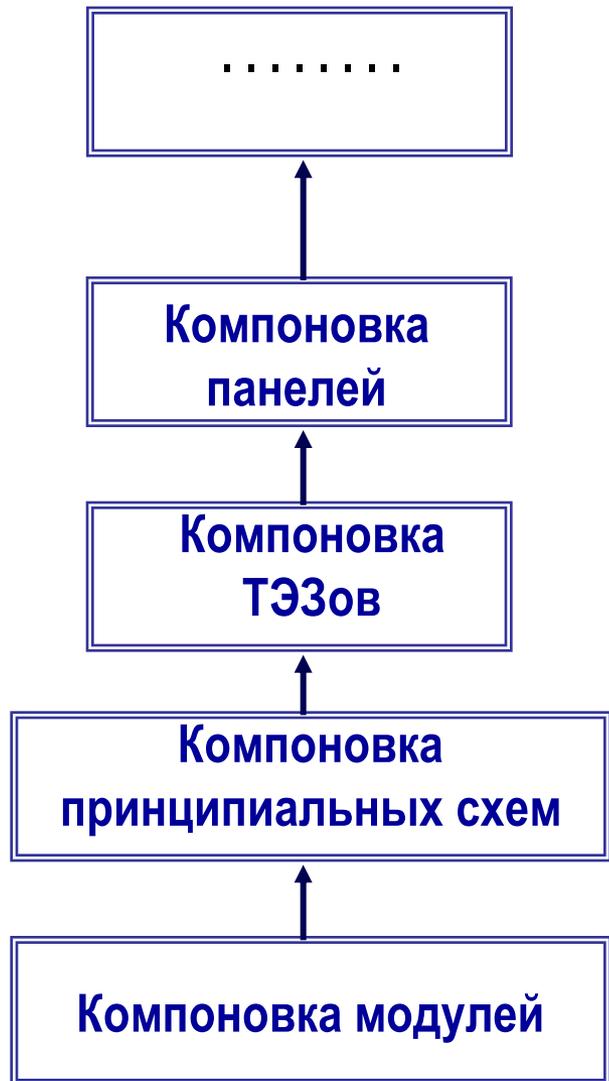
студент: Ларютин А.В.

научный руководитель: доцент, к.т.н. Камышная Э.Н.

Цель работы: Разработка программного обеспечения, решающего задачу компоновки элементов ЭВА в модули на начальном этапе проектирования

Решаемые в процессе работы задачи:

- произвести анализ предметной области и определить цели и задачи создаваемого программного продукта;
- сформировать основные концепции и требования к входным и выходным данным;
- рассмотреть возможные методы решения задачи на конкретном примере с целью выбора наиболее удобного для формализации метода;
- осуществить выбор требуемого лингвистического, информационного, аппаратного и общесистемного программного обеспечения



Критерии компоновки модулей:

1. Минимизация числа модулей i -го уровня

$$N_i = \sum_{j \in J_i} x_{i,j}$$

2. Минимизация типов модулей i -го уровня

$$k_{i\hat{a}\hat{\delta}} = 1 - nm / N_{i-1}$$

3. Минимизация избыточности реализации

$$\Delta m_i = \sum_{k=1}^{N_i} \Delta m_{i,k}$$

4. Минимизация числа внешних выводов

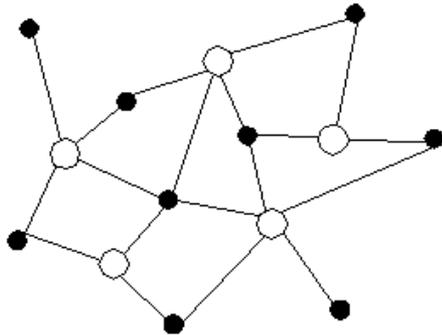
$$S_i = \sum_{k=1}^{N_i} s_{i,k}$$

5. Минимизация числа межмодульных соединений

$$R_i = \sum_{r=1}^{S_i} 1 / p_r$$

Математическая постановка задачи компоновки

1. Определение способа представления:



Граф (аналитический метод)

| | | | | | |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |

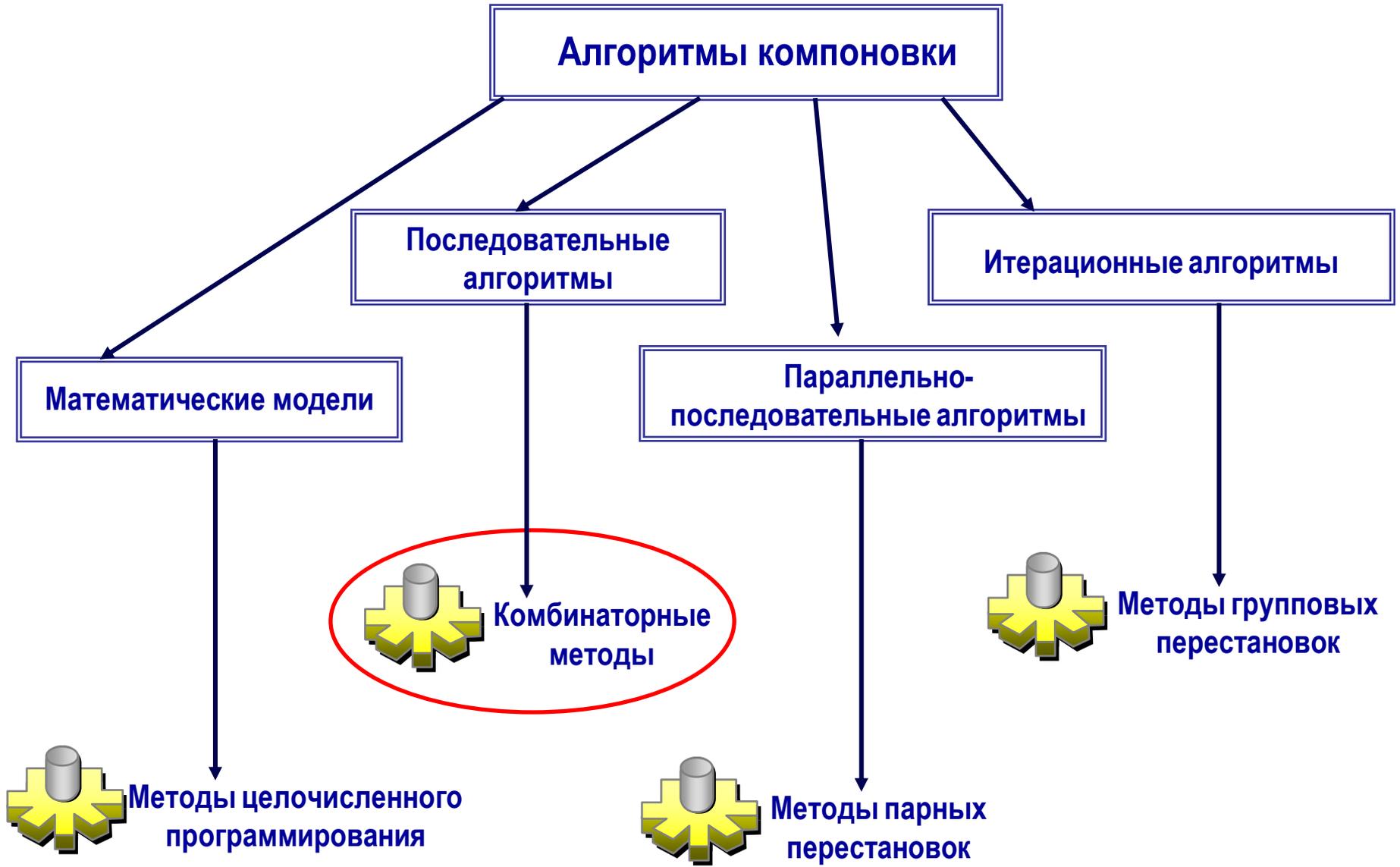
Матрица (матричный метод)

2. Основные условия компоновки:

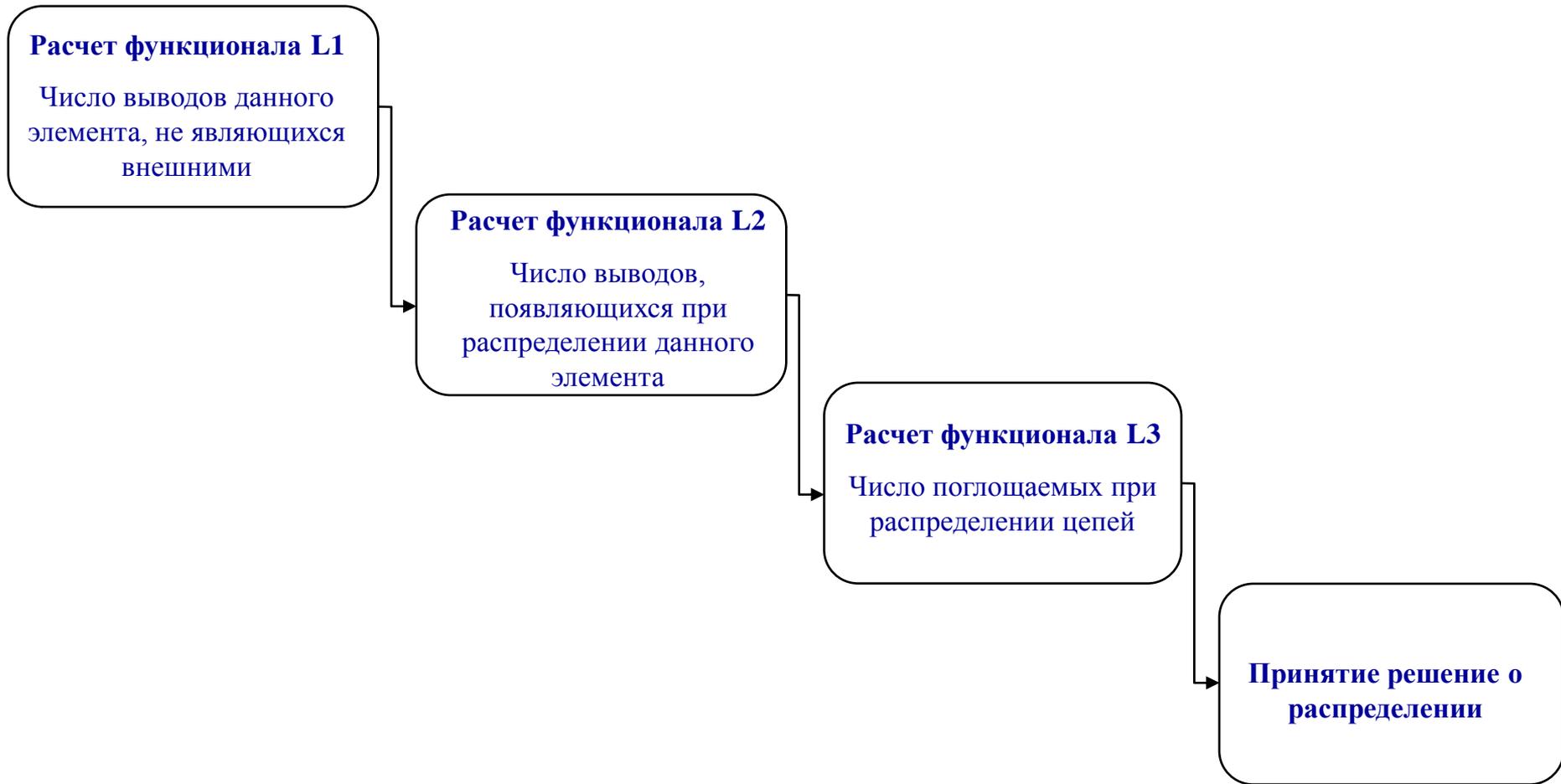
1. $\bigcup_{i=1}^N T_i = E$
2. $\forall ij \in \{1..N\} (T_j \cap T_i = \emptyset)$
3. $\forall i \in \{1..N\} (T_i \neq \emptyset)$

3. Ограничения компоновки в зависимости от уровня иерархии и решаемой задачи

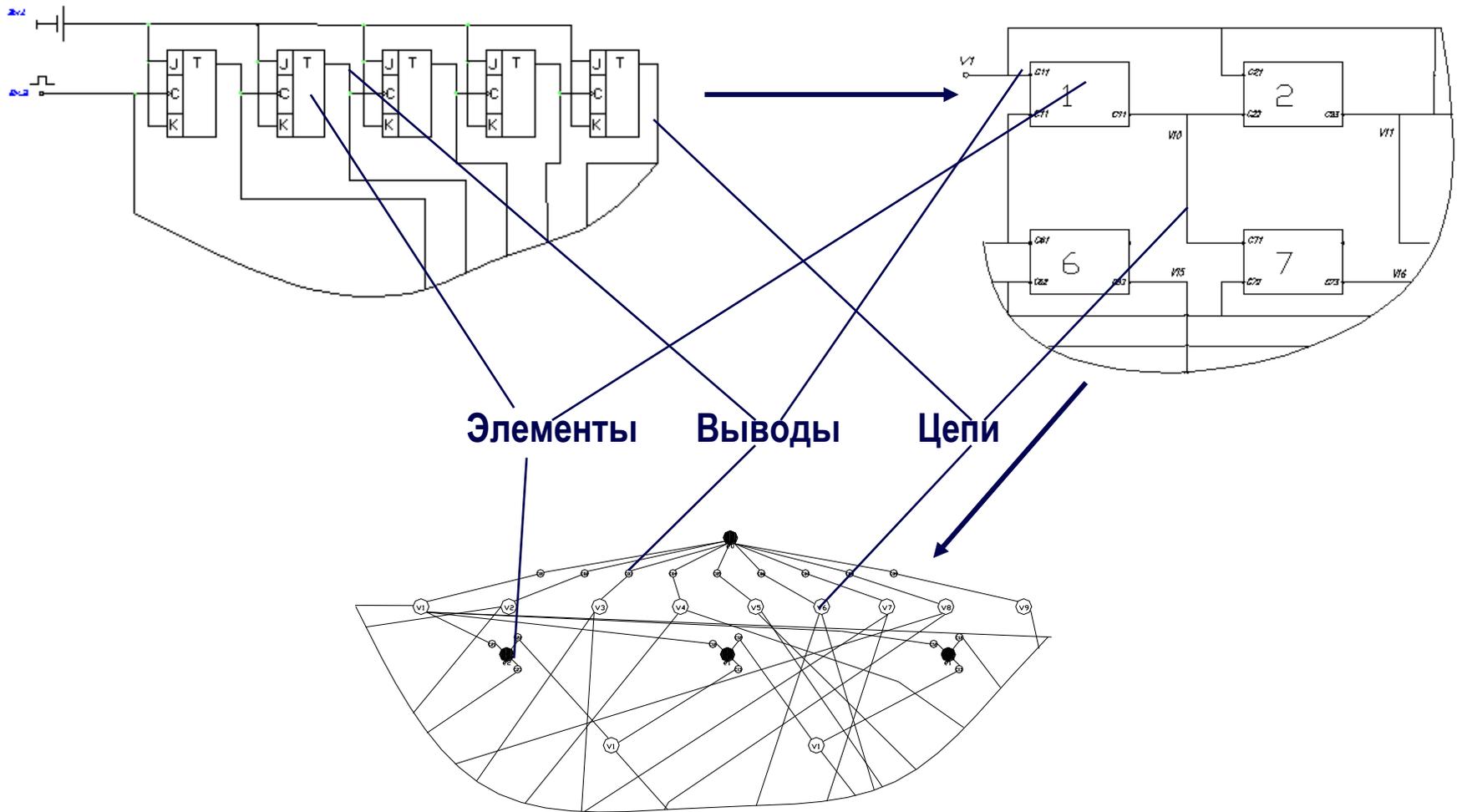
Классификация алгоритмов

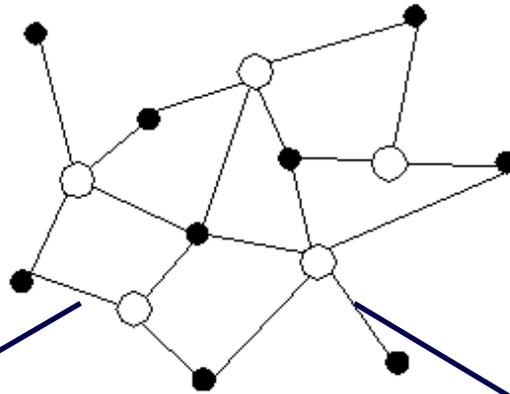


Последовательное назначение элементов



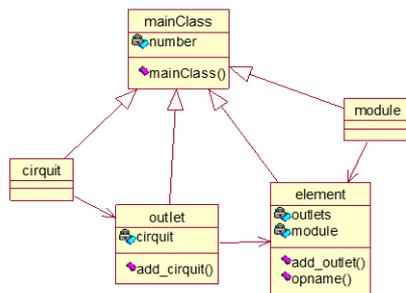
Последовательность формализации





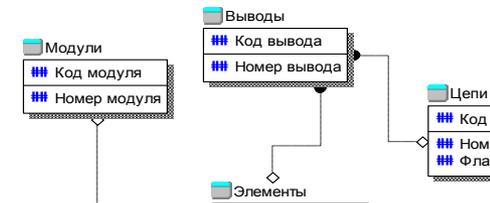
АНАЛИТИЧЕСКАЯ МОДЕЛЬ

Объектно-ориентированный подход



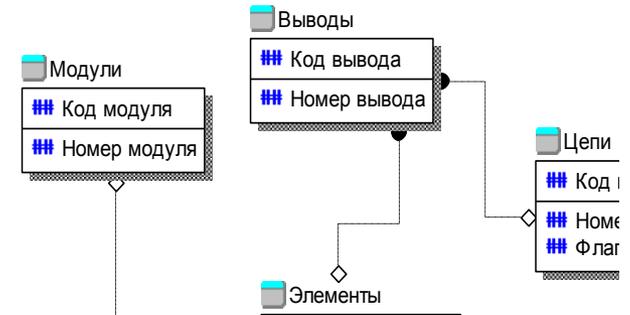
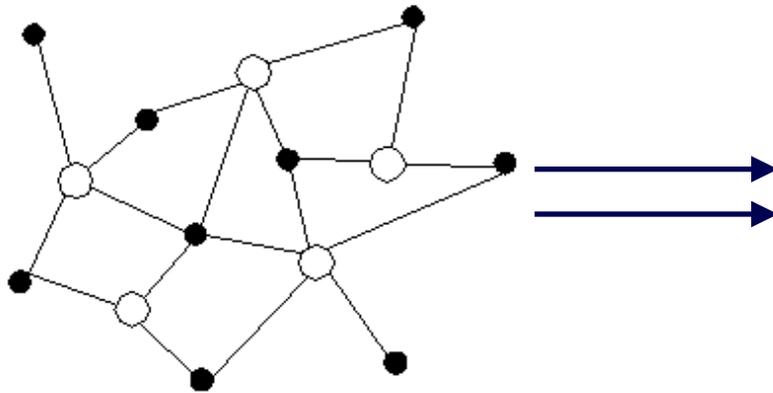
```
class TElement {  
    private int number;  
    private ArrayList outlets;  
    private TModule module;  
    .....  
}
```

Реляционный подход (с использованием СУБД)



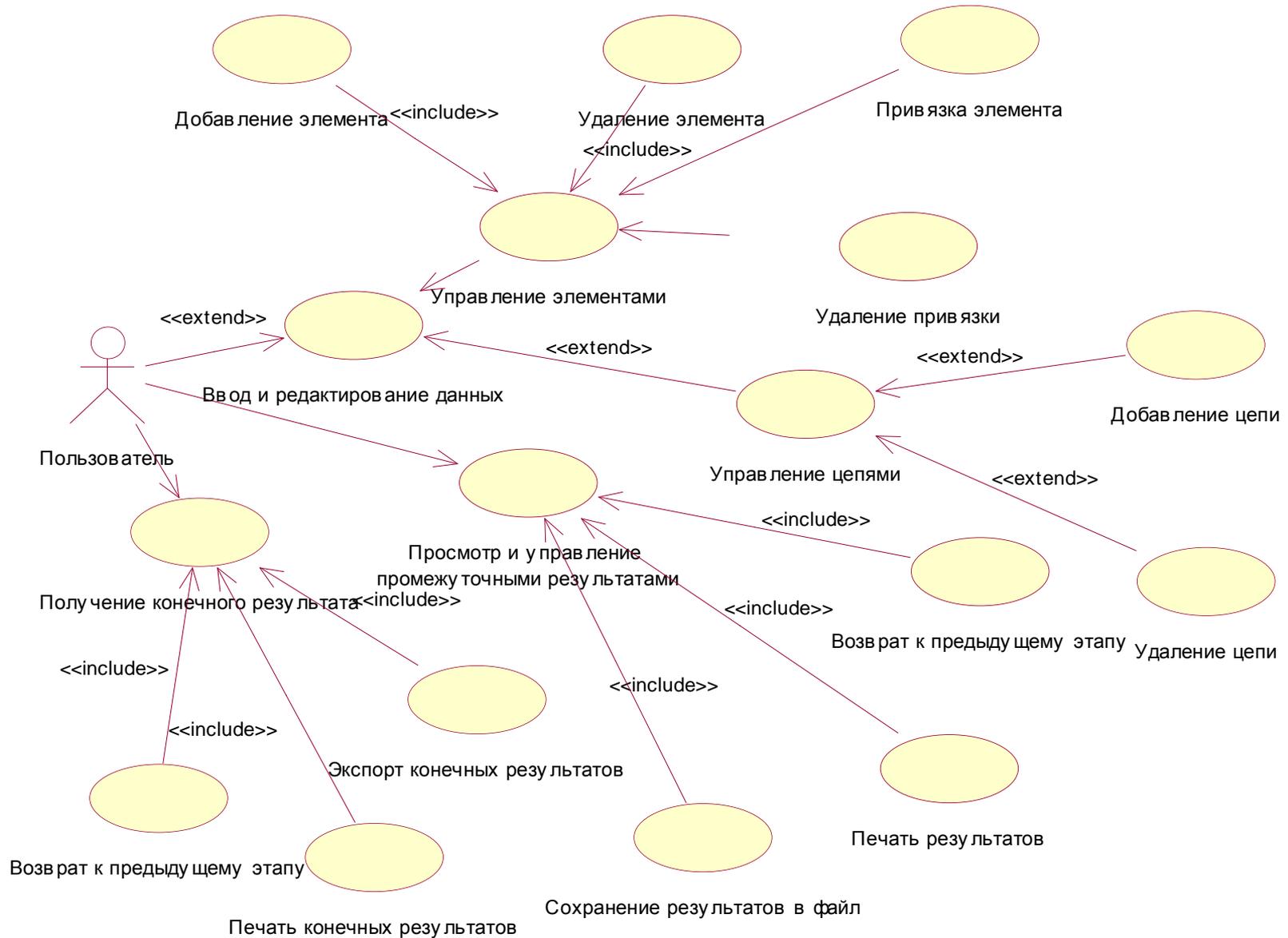
```
CREATE TABLE elements (  
    el_id AUTOINCREMENT  
    , el_num INTEGER NOT NULL  
    , el_mod_id INTEGER NOT NULL  
)
```

Преимущества реляционного подхода

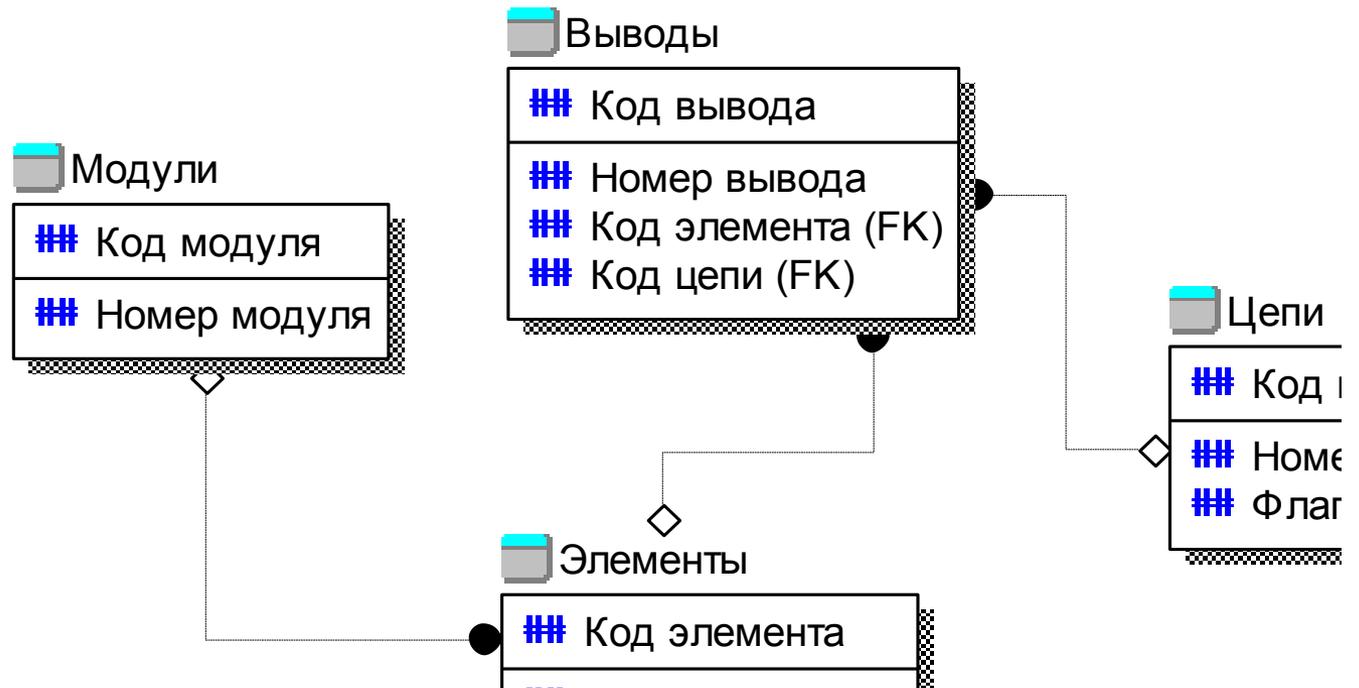


1. Скорость выбора по заданным критериям
2. Простота реализации
3. Надежность
4. Оптимизация основных операций

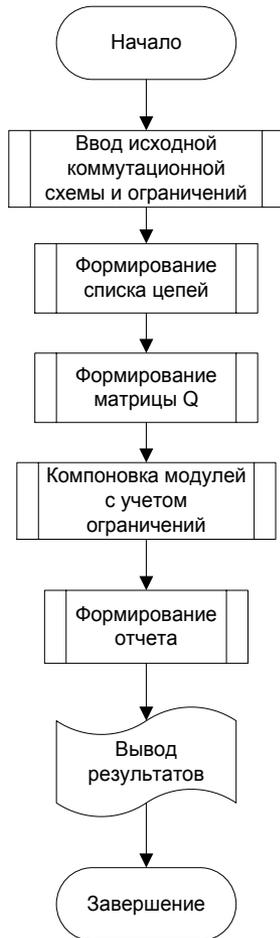
Диаграмма вариантов использования



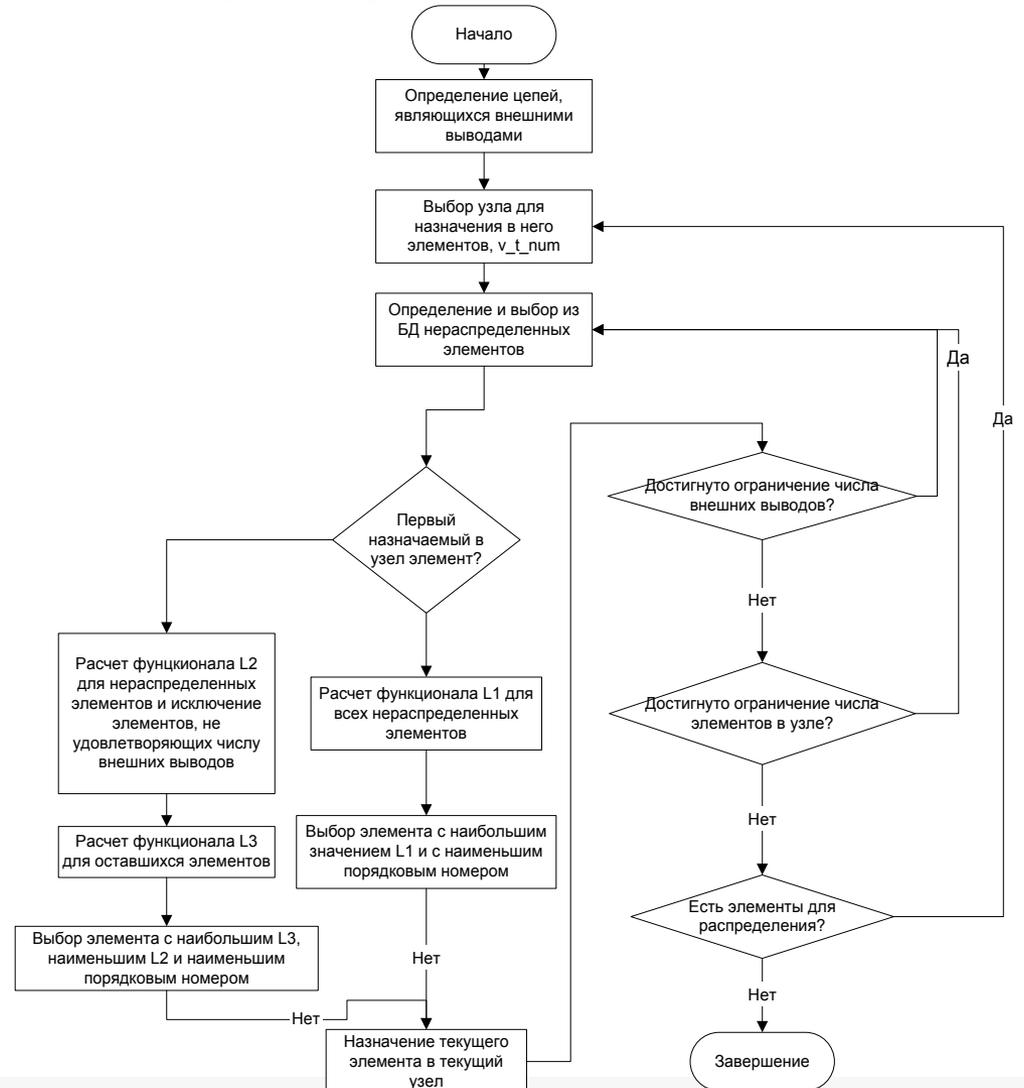
Структура базы данных



Основной алгоритм ПО



Алгоритм реализации компоновки



Представление входных и выходных данных

Компоновка: Ввод данных

Данные о схеме

Название
Автомат включения/выключения нагрузки

Описание
Автомат включения/выключения нагрузки, рассчитанный на часовой цикл предназначен для обеспечения периодического включения и выключения нагрузки, через заданные пользователем с клавиатуры интервалы

Цепи

| Номер | Внешняя |
|-------|---------|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |

Элементы

| Номер | Выводы |
|-------|--------|
| 1 | 3 |
| 2 | 3 |
| 3 | 2 |
| 4 | 4 |

Ограничения

Число внешних выводов
8

Число элементов в модуле
4

Список цепей

| Цепь | Элемент | Вывод |
|------|---------|-------|
| C1 | e1 | Ср11 |
| | e2 | Ср21 |
| | e3 | Ср32 |
| C2 | e1 | Ср12 |
| | e3 | Ср31 |

Отмена Расчет

Компоновка: результат

Число элементов: **20**

Число цепей: **18**

Число внешних цепей: **9**

Число модулей: **6**

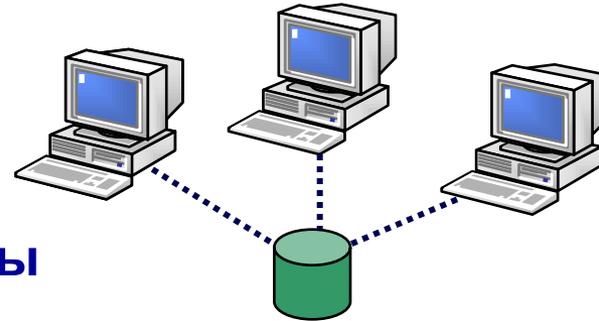
Max элементов в модуле: **4**

Max выводов модуля: **8**

Состав модулей:

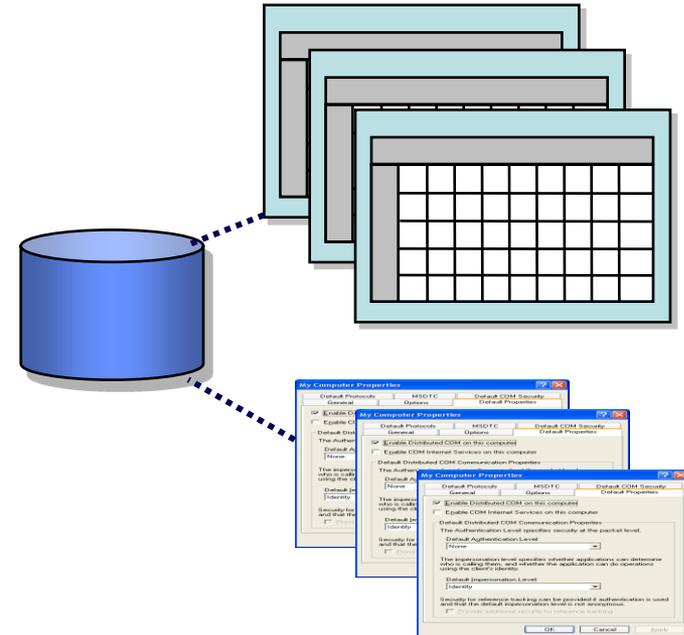
| № модуля | № элемента | Вн. вывод |
|----------|------------|-----------|
| T1 | e18 | 1 |
| | e19 | 1 |
| | e20 | 2 |
| T2 | e1 | 1 |

Печать Закрыть



1. Возможность совместной работы

2. Высокая степень расширяемости



В работе были рассмотрены и решены следующие задачи:

- рассмотрены основные способы компоновки, осуществлена их формализация и алгоритмизация при рассмотрении на конкретном примере;
- разработан новый способ хранения и обработки данных о принципиальных и коммутационных схемах с использованием СУБД;
- разработана информационная модель, обеспечивающая возможность полной реализации рассмотренного способа компоновки;
- создано прикладное программное обеспечение;
- осуществлено тестирование разработанного ПО на примере рассматриваемой схемы, давшее удовлетворительные результаты