

Систематизация и анализ паттернов
проектирования на основе стандартов теории
решения изобретательских задач

Зиненко Ольга Игоревна

532 группа

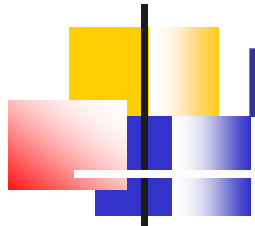
Научные руководители:

ст. пр. Одинцов И.О.

ст.н.с. Рубин М.С.

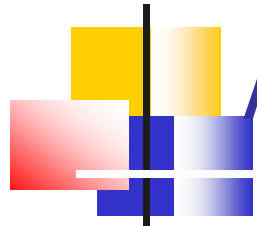
Санкт-Петербургский Государственный Университет

2010



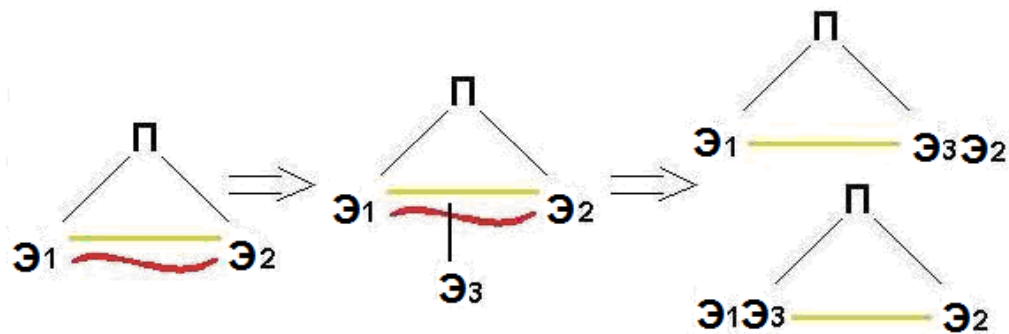
Цель и задачи работы

- **Цель:** разработка единого алгоритма решения задач из области программирования на основе стандартов теории решения изобретательских задач и паттернов проектирования и его программная реализация
- **Задачи:**
 1. Анализ литературы по теме работы
 2. Создание картотеки задач из области программирования
 3. Адаптация стандартов ТРИЗ для области программирования
 4. Анализ алгоритма применения стандартов АИСТ-2010
 5. Анализ, систематизация и перевод паттернов проектирования на общесистемный язык
 6. Описание единого алгоритма применения стандартов ТРИЗ и паттернов для решения задач АИСТ-2010-П
 7. Разработка программного приложения, реализующего работу алгоритма



Адаптация стандартов и паттернов

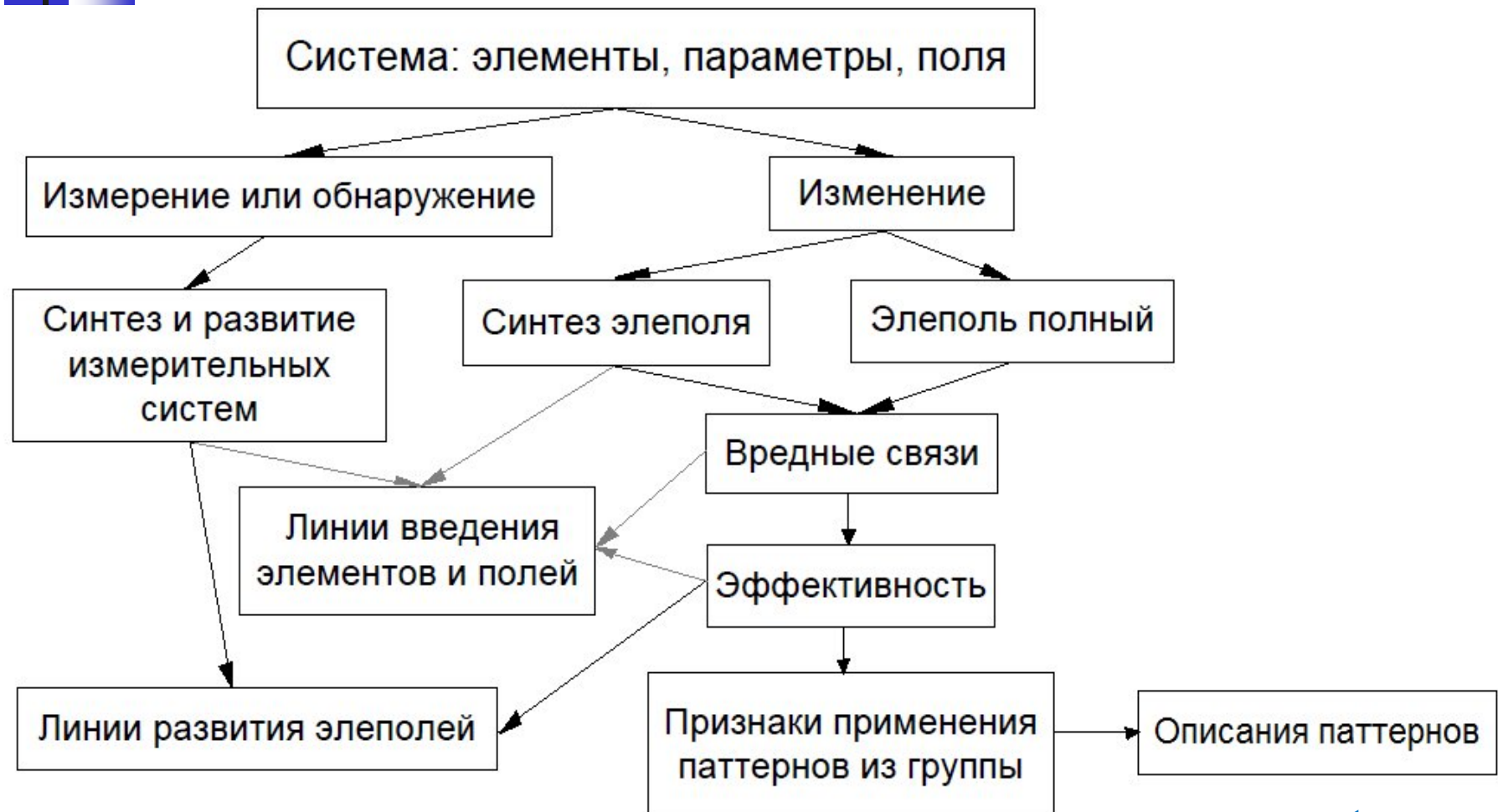
- Базовые понятия ТРИЗ и система стандартов для области программирования

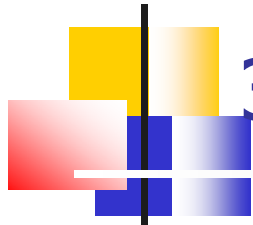


- Перевод паттернов на общесистемный язык
 - Основные абстракции (объекты, классы, конгломераты)
 - Основные понятия (запрос, интерфейс, наследование, делегирование и т.д.)
 - Признаки применения в терминах элементов



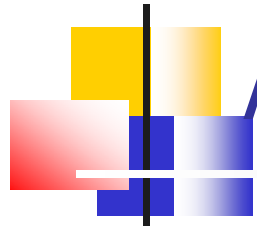
Схема АИСТ-2010-П



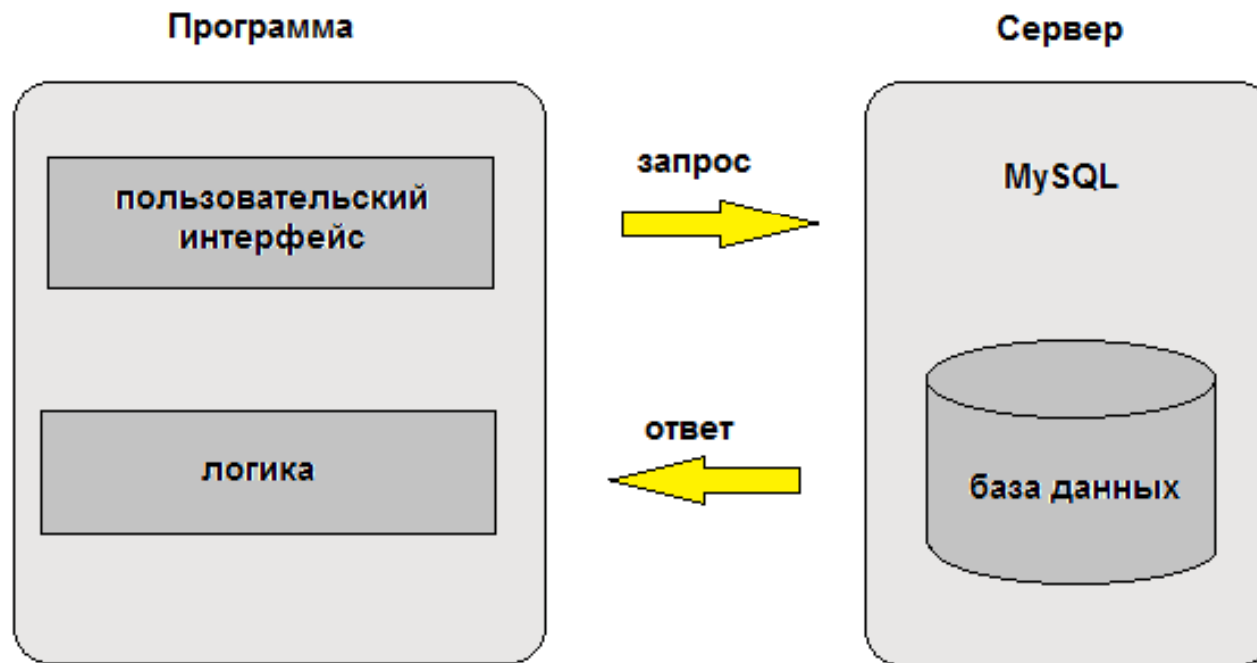


Этапы решения задач

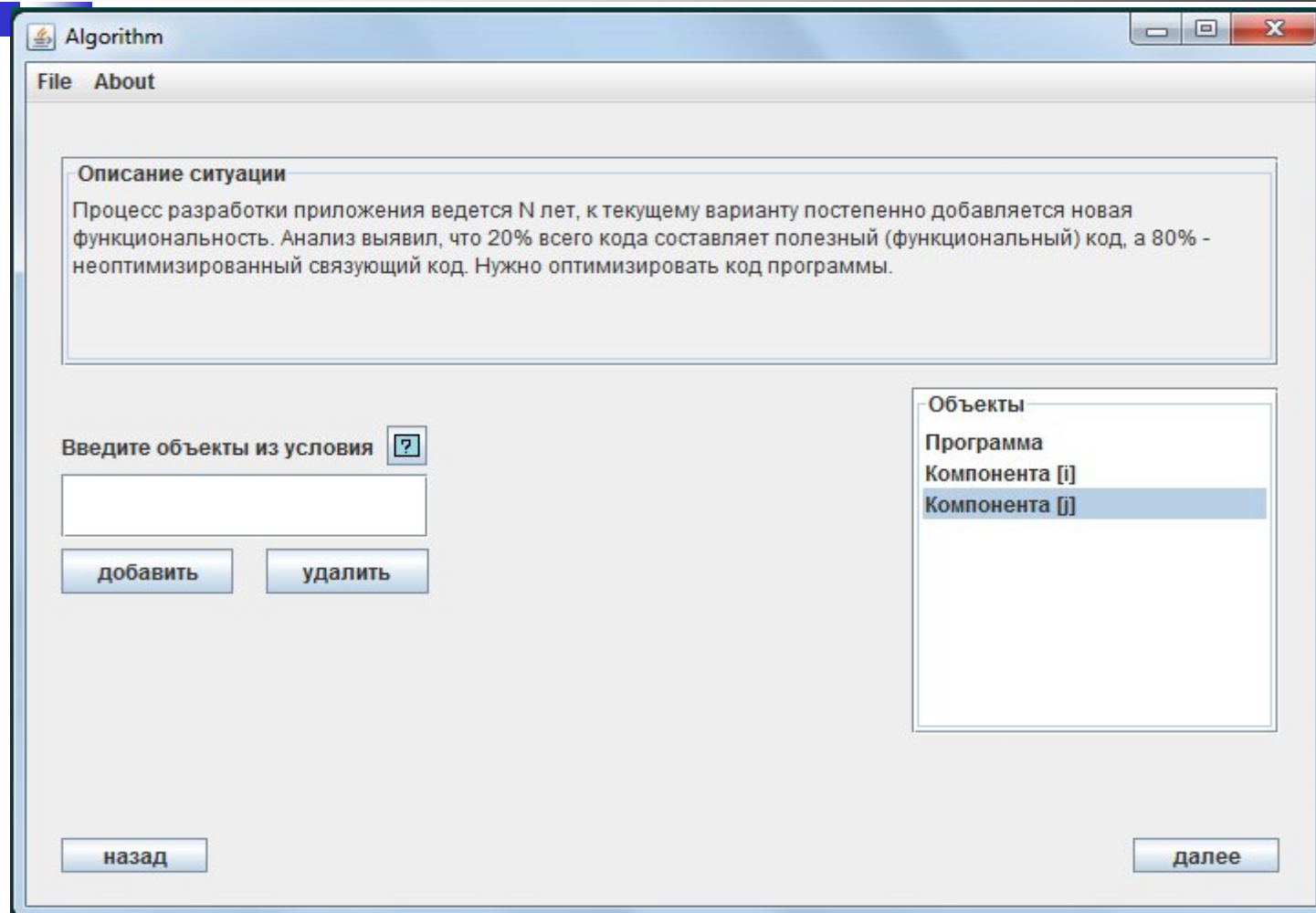
Этап	Входные данные	Выходные данные	Внутренняя структура
1. Формулировка задачи	Описание проблемной ситуации	Элеполю (полный или неполный)	<ul style="list-style-type: none">• Описание проблемной ситуации• Объекты-претенденты на участие в построении элеполя• Параметризация объектов• Создание элеполюльной структуры
2. Анализ задачи	Элеполюльная структура	Рекомендации по использованию стандартов ТРИЗ	<ul style="list-style-type: none">• Синтез и развитие измерительных систем• Создание элеполюльной структуры• Устранение вредных связей• Повышение эффективности• Линии развития
3. Выбор паттерна	Использованные стандарты	Рекомендации по использованию паттернов, описание, графическая структура, схема применения	<ul style="list-style-type: none">• Выбор паттерна по названию или признакам применения• Вывод информации по подходящему паттерну• Переход к этапу 1 или 2



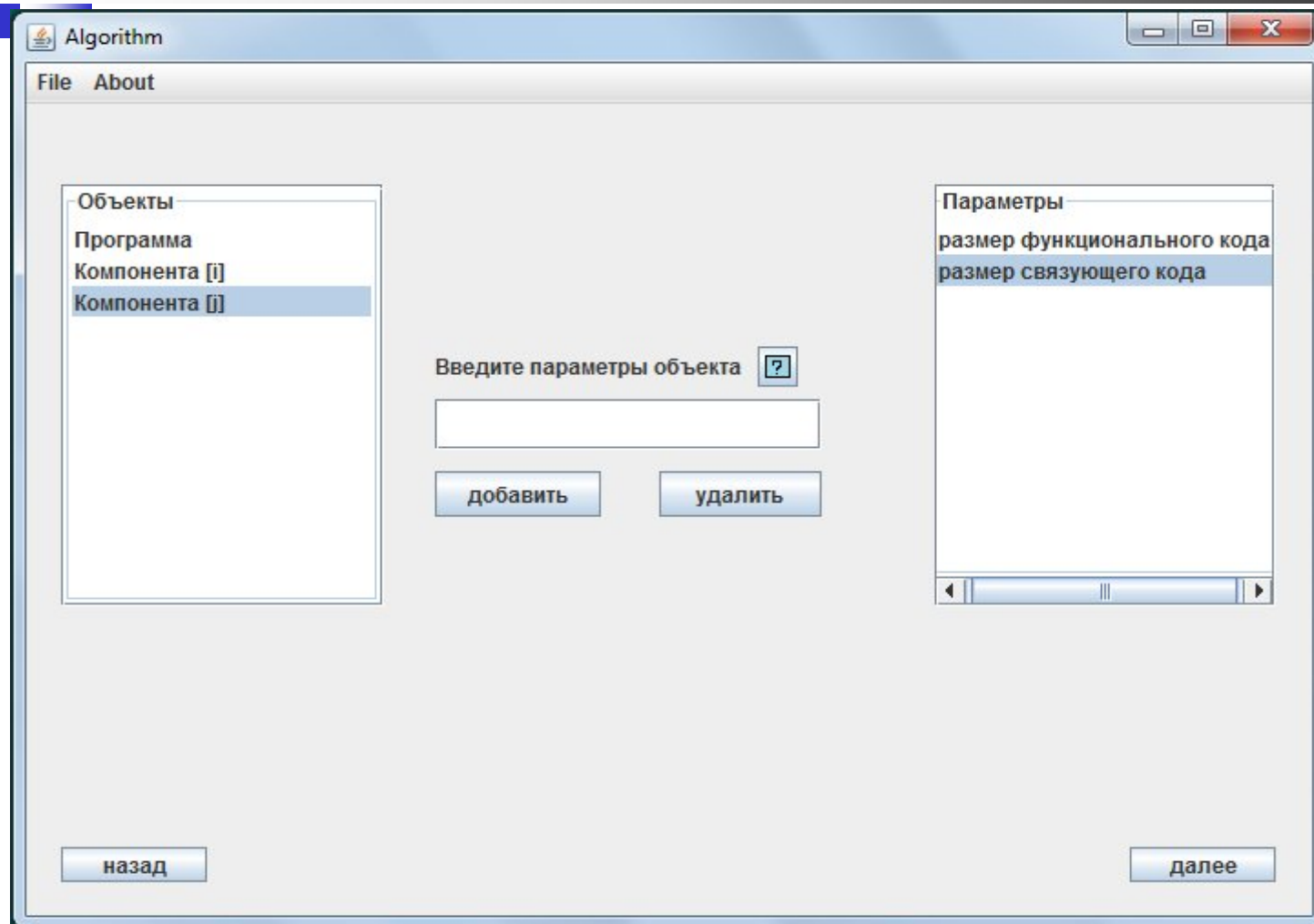
Архитектура системы



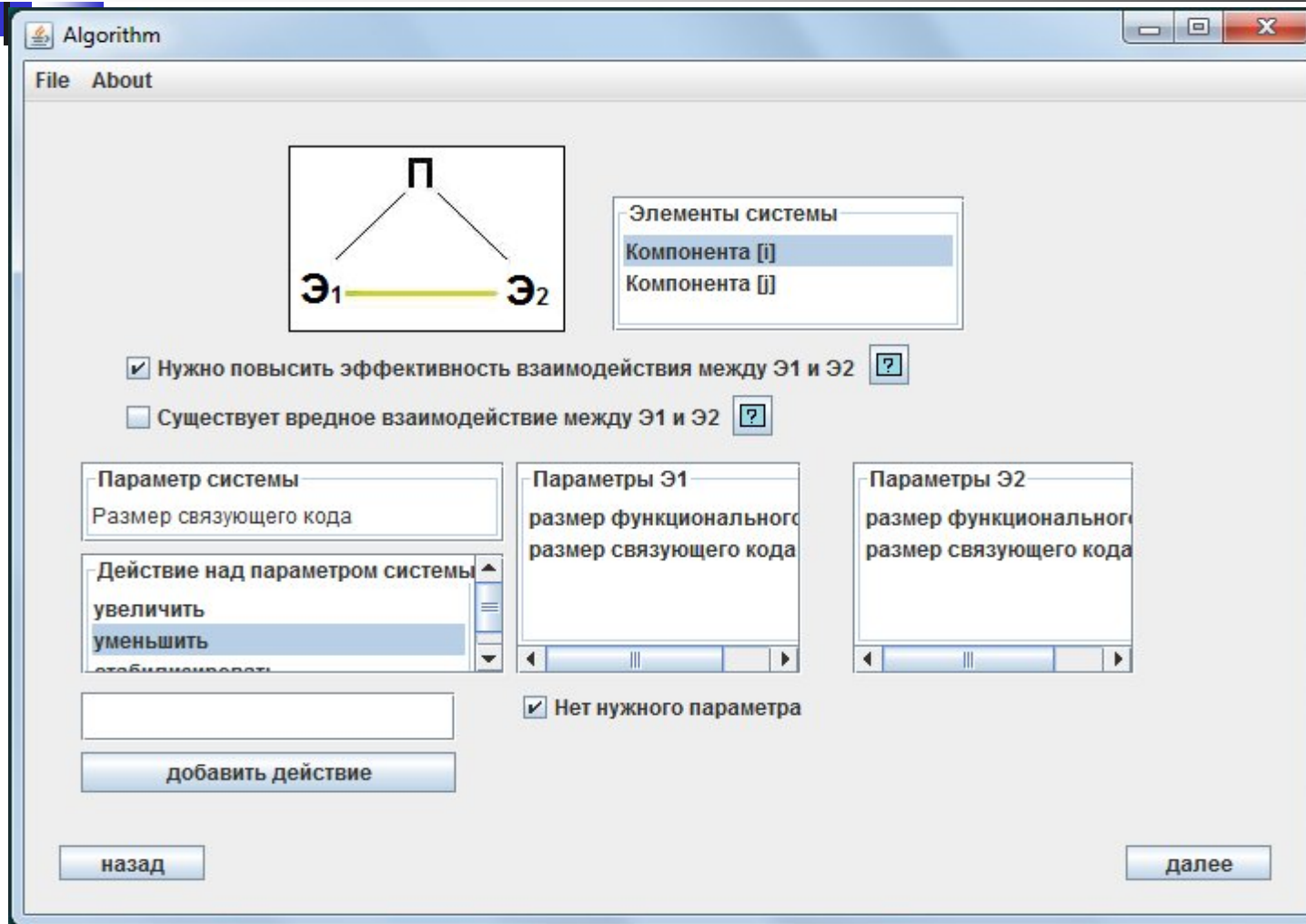
Интерфейс: выбор объектов



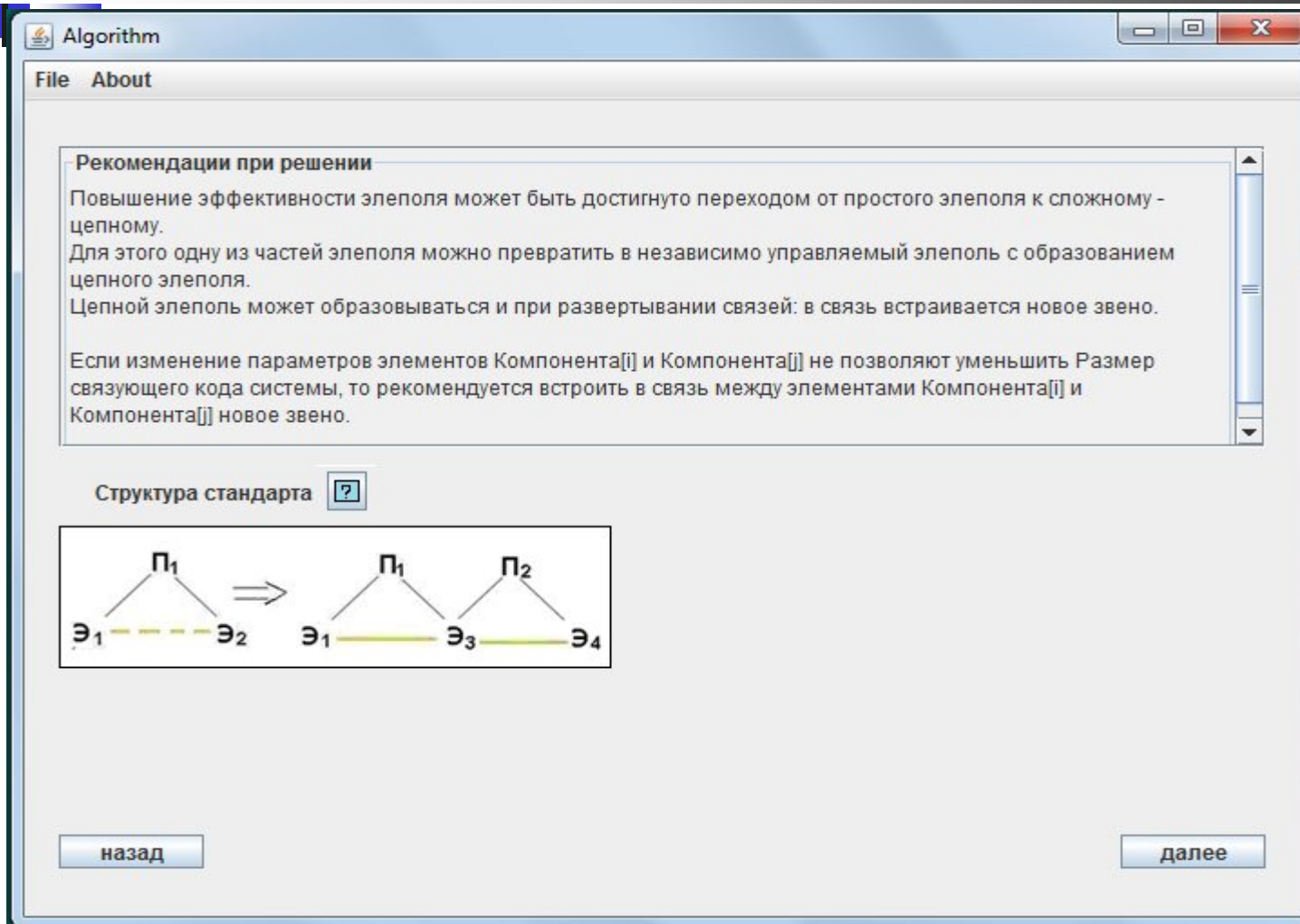
Интерфейс: параметризация объектов



Интерфейс: выбор группы стандартов




Интерфейс: стандарты



Интерфейс: выбор паттерна


Algorithm


File About

Для того, чтобы перейти к описанию паттерна, нужно выбрать его по названию или признаку применения 

Выбрать паттерн по названию

Выбрать паттерн по признакам применения

<выберите паттерн> 


Нужно объединить все связи системы в одном элементе 

назад далее

Интерфейс: рекомендации по выбору и применению паттерна

Algorithm

File About

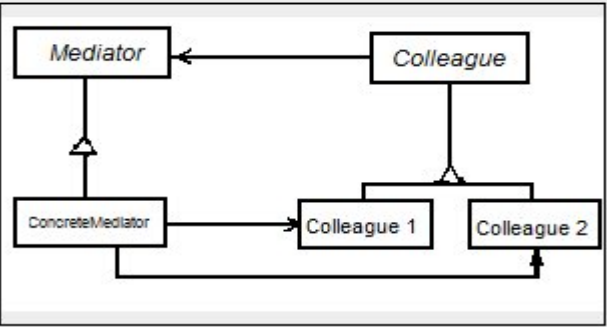
Посредник 

Посредник - паттерн поведения элементов, предоставляющий единый центр взаимодействия определенной группы элементов, которые должны быть взаимосвязаны друг с другом. Определяет элемент, содержащий сведения о способах взаимодействия множества элементов. Посредник обеспечивает слабую связанность системы, избавляя объекты от необходимости явно ссылаться друг на друга, и позволяя тем самым независимо изменять взаимодействия между ними.

Признаки применения

- Имеются элементы, связи между которыми сложны и четко определены.
- Нельзя повторно и независимо использовать элемент (например, в другом контексте), поскольку он обменивается информацией со многими другими элементами.

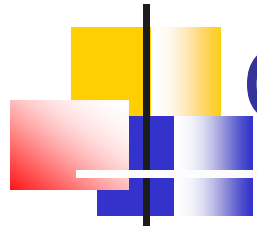
Схема использования



```
classDiagram
    class Mediator
    class Colleague
    class ConcreteMediator
    class Colleague1["Colleague 1"]
    class Colleague2["Colleague 2"]
    Mediator <|-- ConcreteMediator
    Colleague <|-- Colleague1
    Colleague <|-- Colleague2
    Mediator <--> Colleague
    ConcreteMediator <--> Colleague1
    ConcreteMediator <--> Colleague2
```

Mediator - посредник.
Определяет интерфейс для обмена информацией с объектами Colleague.
ConcreteMediator - конкретный посредник.
Реализует кооперативное поведение, координируя действия объектов Colleague.
Владеет информацией о коллегах и подсчитывает их.
Классы Colleague - коллеги:
Каждый класс Colleague «знает» о своем объекте-распорядителе Mediator.
Все коллеги обмениваются информацией только с

назад далее



Основные результаты

- Новая система стандартов ТРИЗ адаптирована для области программирования
- Паттерны переопределены в терминах новых элементов - объединений классов и объектов
- Паттерны систематизированы в соответствии со стандартами ТРИЗ
- Разработан алгоритм АИСТ-2010-П для решения задач из области программирования
- Создано программное приложение, представляющее собой справочно-обучающую систему
- Собрана картотека задач для проверки алгоритма