

Основы программирования

Занятие 3

TECT № 2

<https://goo.gl/Rg08sy>

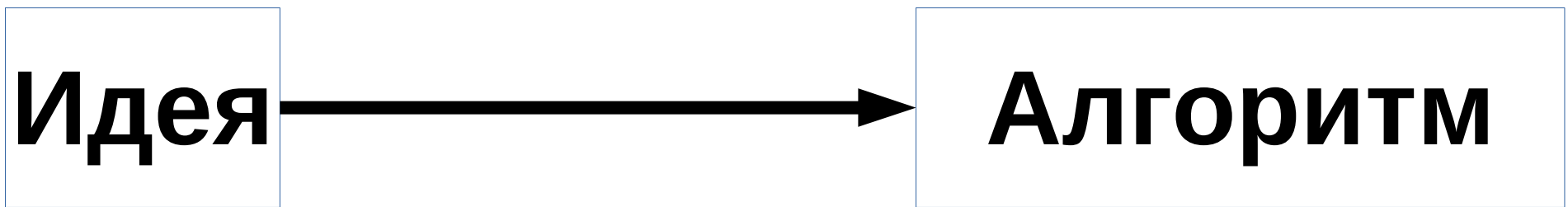


Жизненный цикл программного продукта

С чего начинается разработка любой программы?

С чего начинается разработка
любой программы?

**В основе любой
программы лежит идея**



Идея



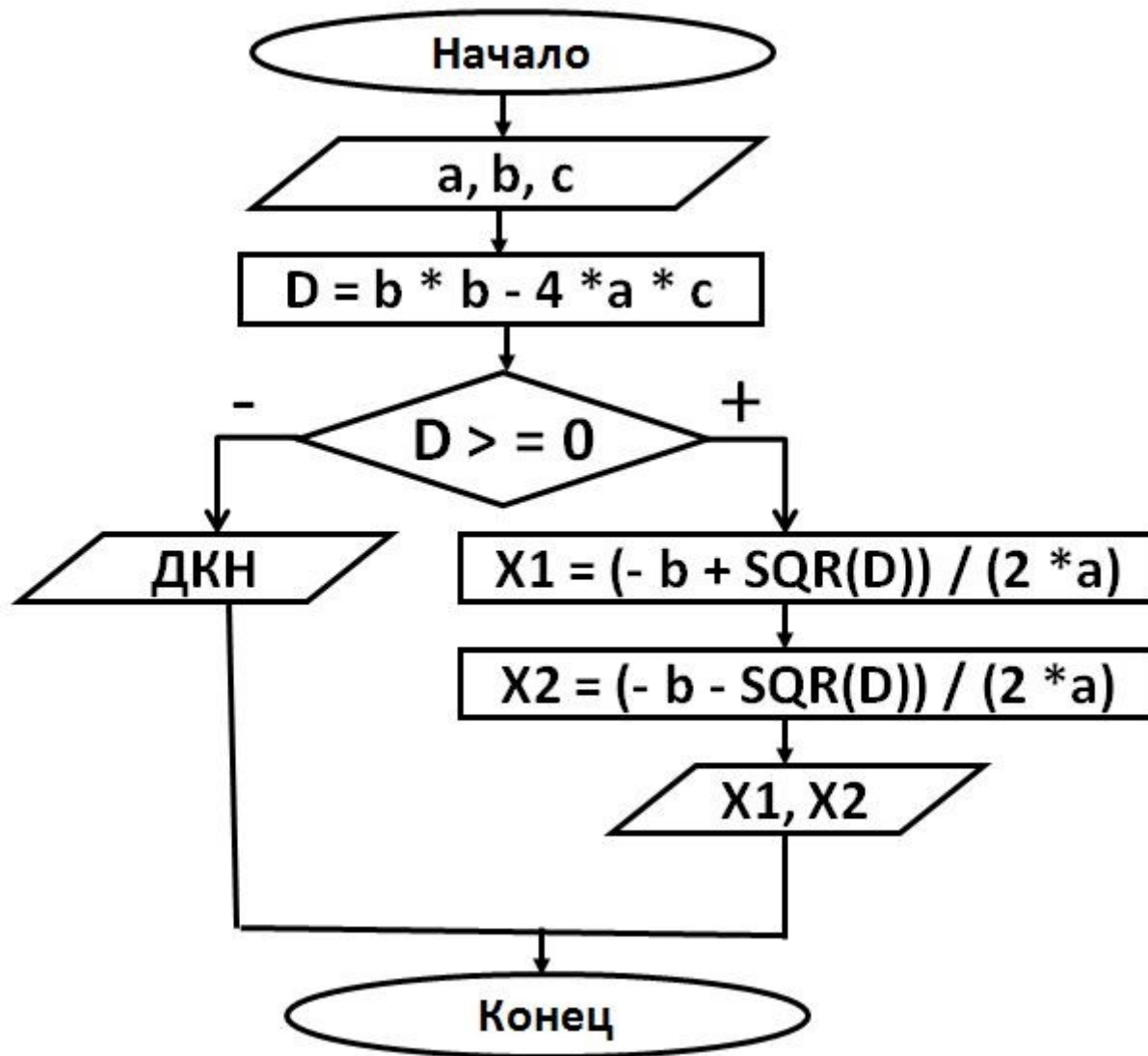
Алгоритм

Формализация:

Вербальная:

- «В голове»
- «На словах»
- «На бумажке»
- Графическая:
 - Блок-схема (ДРАКОН-схема)
 - Структурограммы (диаграммы Насси-Шнейдермана)
 - UML-диаграмма
- Псевдокод

Блок-схемы



Дружелюбный русский алгоритмический язык, который обеспечивает наглядность

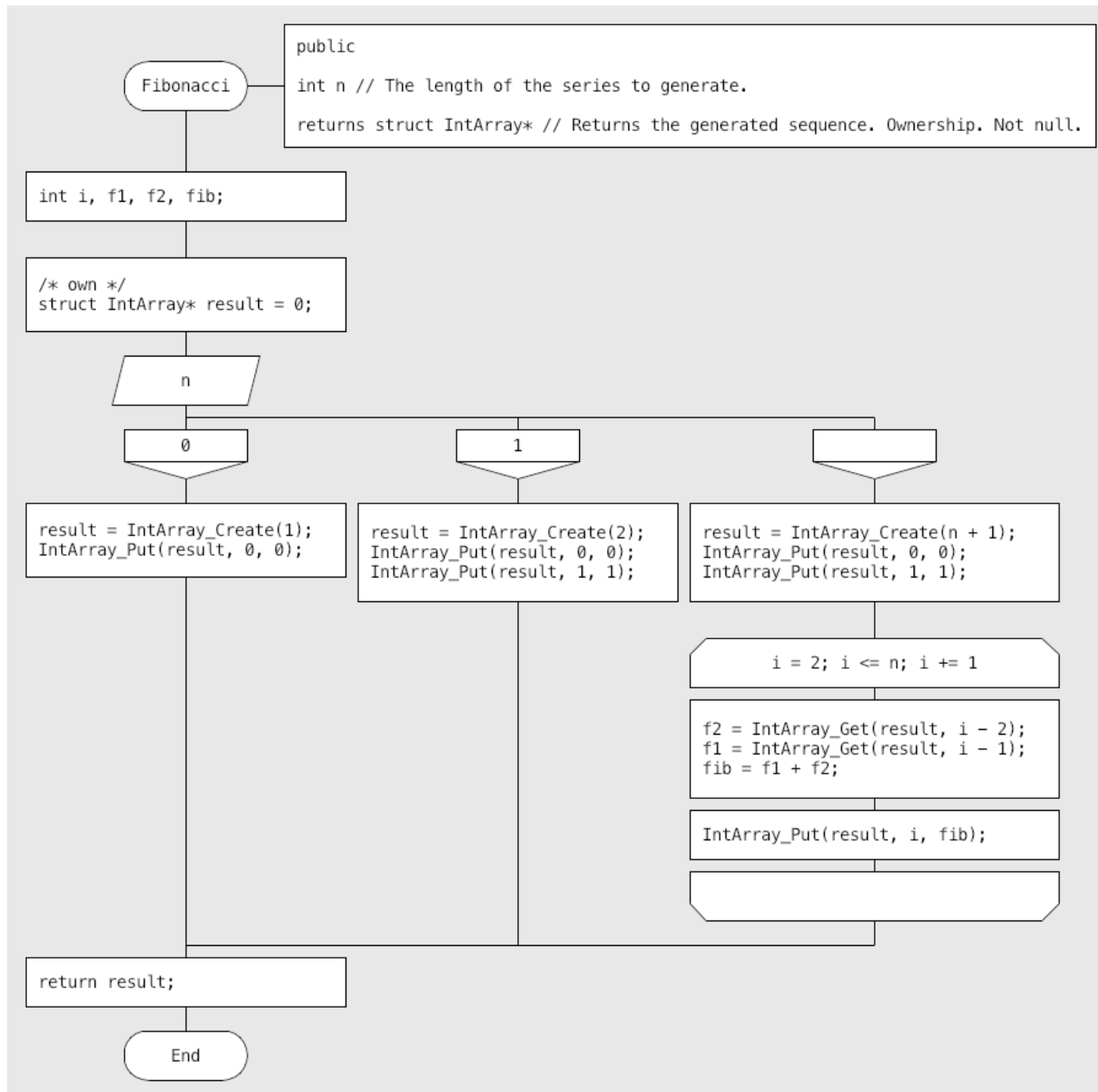
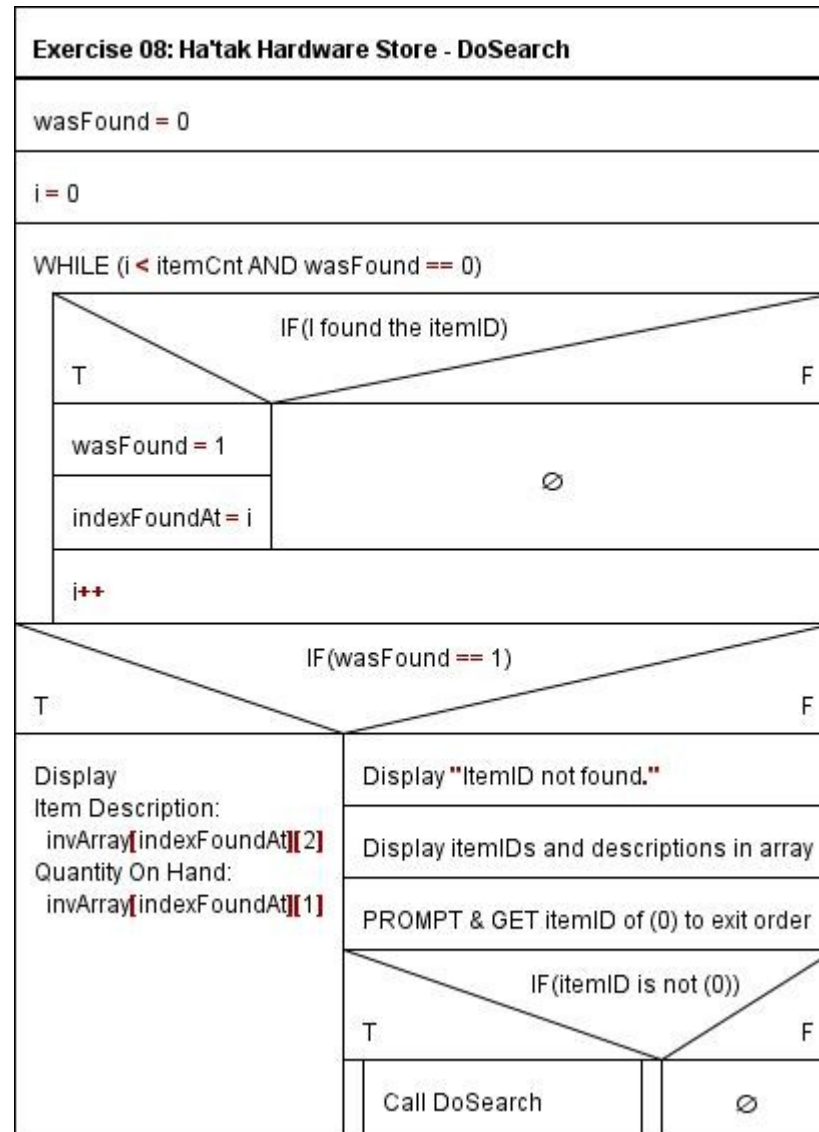


Диаграмма Насси — Шнейдермана

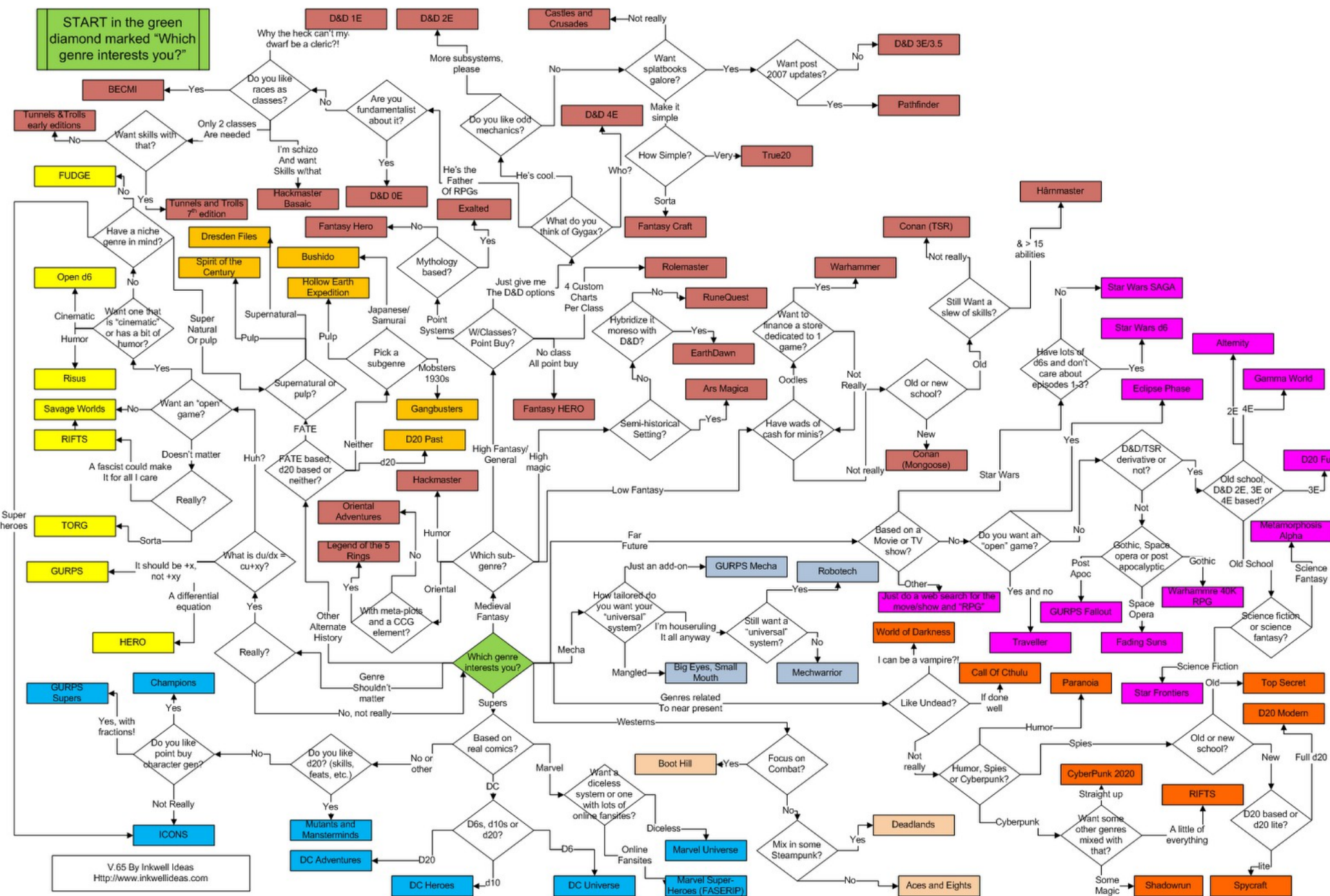


Достоинства блок-схем

- Простота
- Наглядность
- Алгоритм легко модифицируется

Недостатки блок-схем

- Подходят только для проектирования алгоритмов в процедурном стиле
- Хороши только для коротких алгоритмов



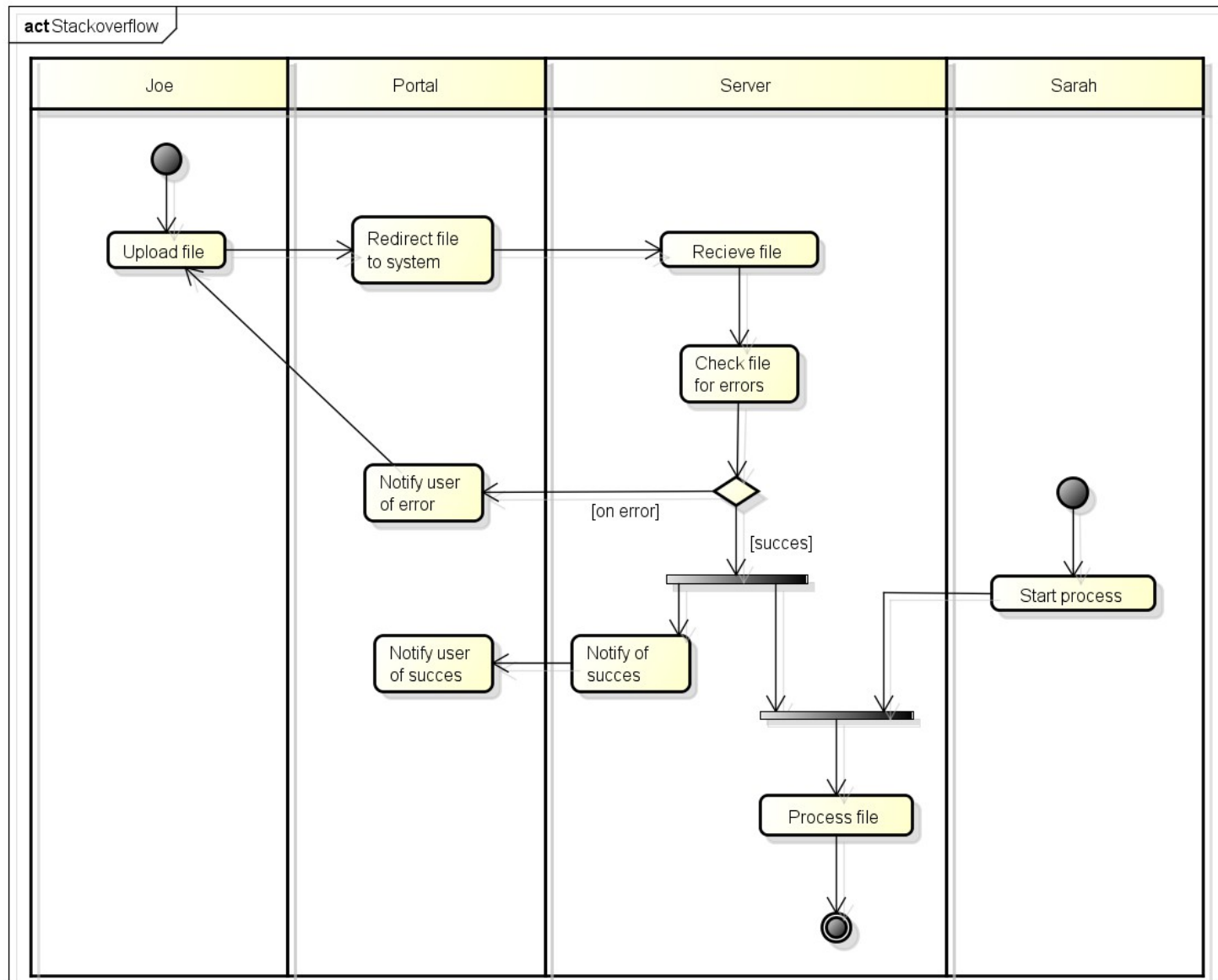
UML (Unified Modeling Language)

- Структурные диаграммы:
 - Диаграмма классов
 - Диаграмма компонентов
 - Диаграмма композитной/составной структуры
 - Диаграмма кооперации (UML2.0)
 - Диаграмма развёртывания
 - Диаграмма объектов
 - Диаграмма пакетов
 - Диаграмма профилей (UML2.2)

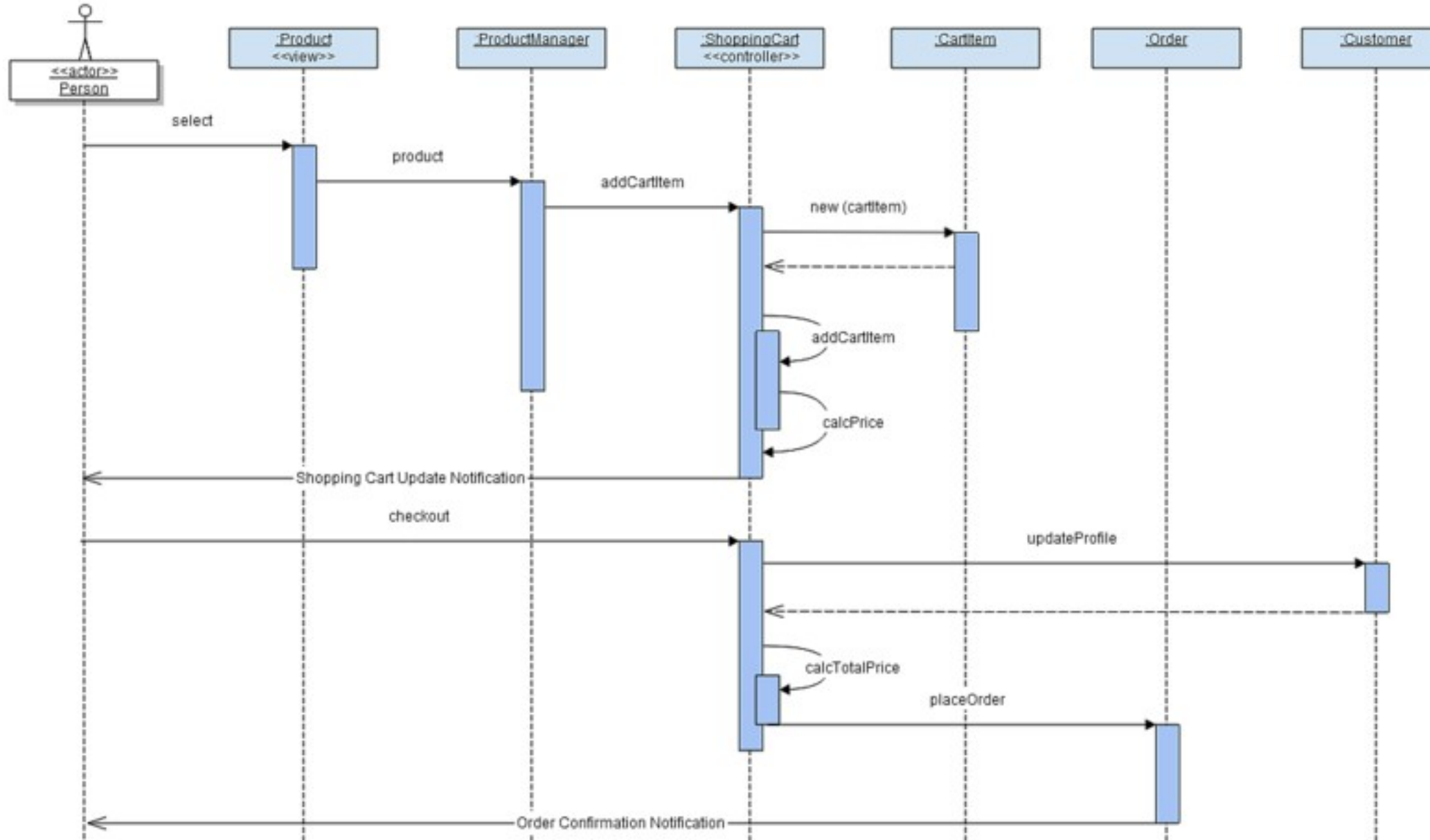
UML (Unified Modeling Language)

- Диаграммы поведения:
 - Диаграмма деятельности
 - Диаграмма состояний
 - Диаграмма вариантов использования
 - Диаграммы взаимодействия:
 - Диаграмма коммуникации (UML2.0) / Диаграмма кооперации (UML1.x)
 - Диаграмма обзора взаимодействия (UML2.0)
 - Диаграмма последовательности
 - Диаграмма синхронизации (UML2.0)

Пример диаграммы деятельности



Пример диаграммы последовательности



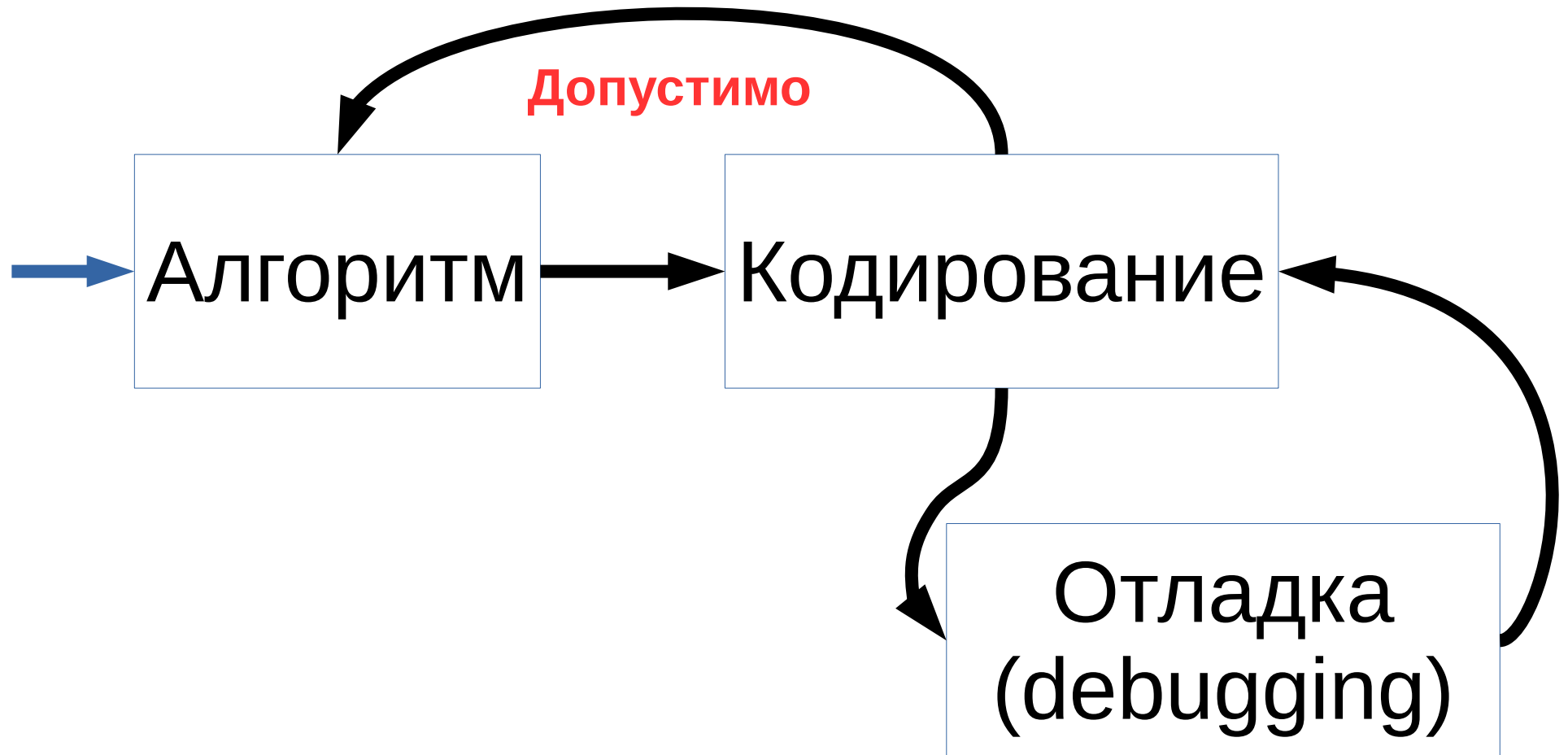
UML

- Достоинства

- Подходят для проектирования сложных алгоритмов
- Очень удобны для ООП
- Есть кодогенерация
- Удобно работать в команде

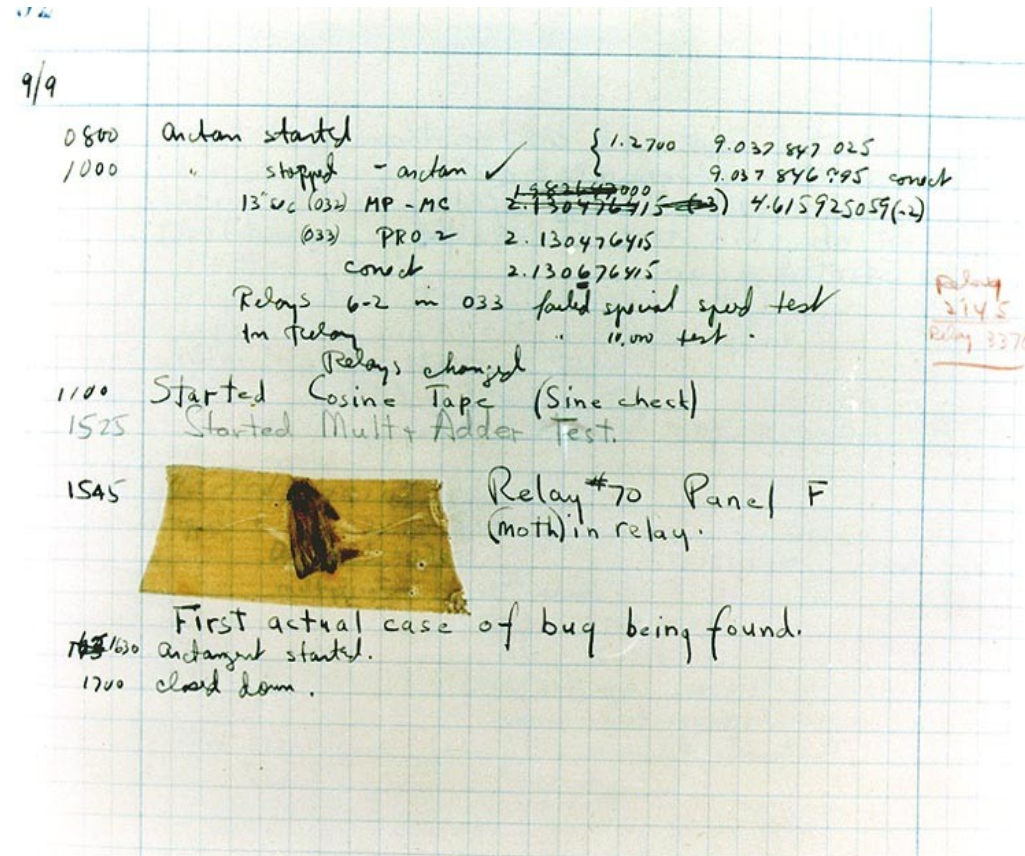
- Недостатки

- Для большинства алгоритмов средней сложности можно потратить больше времени на изучение UML и создание диаграмм, чем на непосредственно программирование

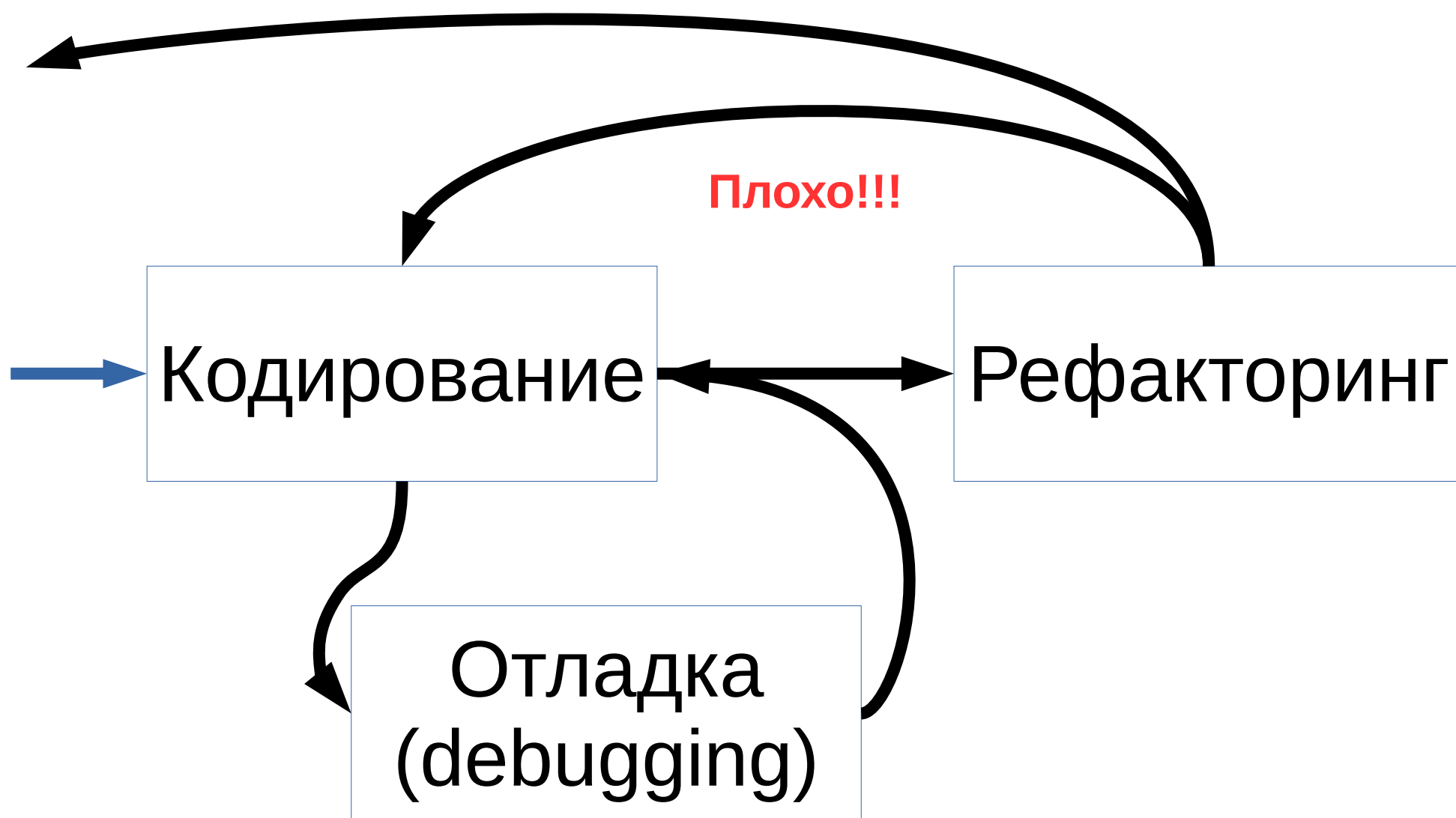


Отладка

- Процесс поиска и исправления ошибок в коде (без изменения алгоритма)
- По легенде слово изобрела Грейс Хоппер, хотя возможно и раньше

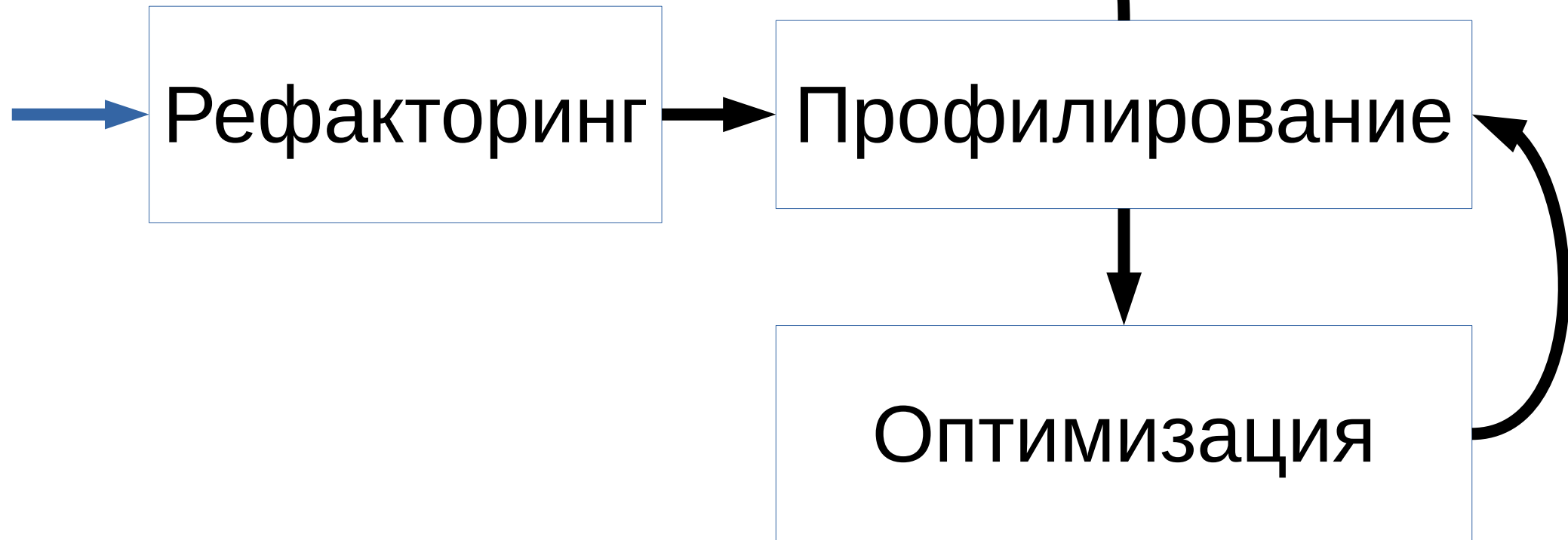


Возврат к алгоритму недопустим!!!



- Рефакторинг – процесс изменения исходного кода без изменения алгоритма и реализации
- «Переписываем программу начисто»

К кодированию
Можно менять реализацию.
Но не радикально!



Иногда приходится
возвращаться к кодированию...

→ Оптимизация



Тестирование



Эксплуатация

Новая идея, новая версия...

Инструменты программирования

Спасибо за внимание

- Вопросы?
- Предложения?
- Просьбы?
- Жалобы?