

1. Секция

Исходный код 1: Т

```
ext
```

Замечание. Тут замечание.

Определение 1 (Lol)

тут что-то определить

Теорема 2 (A Great Theorem)

Something Great

2. Секция

Следствие 3

Lol

4

Lol

Теорема 5

Lol

```
int main(int argc, char **argv) {  
    return 0;  
}
```

```
int main(int argc, char **argv) {  
    return 0;  
}
```

Замечание. Тут замечание.¹

В этой главе — вероятно самой «информатической» в нашей книге — мы введем некоторые алгоритмические понятия и опишем несколько полученных алгоритмов, необходимых для реализации алгебраических методов. Точное определение алгоритмических границ, в которых будут из

¹Lol

Определение 6

В этой главе — вероятно самой «информатической» в нашей книге — мы введем некоторые алгоритмические понятия и опишем несколько получисленных алгоритмов, необходимых для реализации алгебраических методов. Точное определение алгоритмических границ, в которых будут из

укоризненным способом семантику построений, используемых в алгоритмах 1. После изложения общих направлений¹ мы приступим к исследованию первого фундаментального алгоритма этой книги: дихотомического алгоритма возведения в степень. Для этого будут использоваться два метода построения алгоритмов. Один очень близок к математическим рассуждениям, другой — чисто алгоритмический. При этом у нас появится возможность открыть первый рекурсивный алгоритм, представленный в данной работе. Начав с первого примера алгоритма, мы перейдем к изучению на чальных понятий программирования на языке Ада. И здесь тоже речь пойдет не об изложении полного курса языка Ада, а только о пред-

¹Лол Кек Чебурек!

ставлении нескольких ключевых понятий, необходимых для понимания приведенных программ. Вообще говоря, все понятия будут представлены (по мере надобности) на примерах, во избежание их формального описания (по следующему вопросу лучше обратиться к справочному изданию: Reference manual for the Ada¹ programming language 1). И наконец, завершит эту главу разработка совершенно неэффективного метода вычисления определителей целочисленных матриц. Обсуждаемый метод, основанный на математическом определении детерминанта, позволяет наилучшим образом использовать все возможности современных компьютеров и получить идеальное приближение к бесконечности на машине! Кроме того, обсуждение ситуации позволит нам несколько пополнить представление о программировании на языке Ада.

3. Введение в алгоритмику

Прежде чем приступить к описанию любого алгоритма, нужно говорить общепринятые соглашения, используемые в работе. В информатике чаще, чем в любой другой дисциплине, встречается взаимное непонимание между пользователем и разработчиком новых программ. Поэтому так важно сформулировать аксиоматику, позволяющую связно рассуждать об алгоритмах. Множество аксиом, которое мы собираемся сейчас описать, известно как аксиоматика Хоара [86], в честь исследователя, который первым формализовал семантику структур и действий последовательной алгоритмики.

3.1. Терминология и обозначения

А л г о р и т м — это последовательность более или менее элементарных действий, которая позволяет пошагово решить поставленную задачу за конечное время. Можно формально по индукции определить, что такое алгоритм, нижеследующим образом. Цель этого введения в алгоритмику — изучение и представление всех терминов, фигурирующих в предыдущем определении (стиль определения, который в информатике называется *грамматикой*). Можно описать алгоритм в терминах состояний системы: состояние системы до применения алгоритма, состояние той же системы после

¹Heh, ADA

применения алгоритма. Эти два состояния образуют то, что называется сп ец иф икац ией задачи. Таким же образом можно, описывая поэ