



# Оглавление

<b>Введение</b> .....	<b>7</b>
<b>Глава 1. Понятия «алгоритм» и «программа»</b> .....	<b>8</b>
<b>Глава 2. Python. Первые шаги</b> .....	<b>13</b>
<b>Глава 3. Вывод информации на экран</b> .....	<b>21</b>
<b>Глава 4. Переменные величины. Ввод данных в программу. Инструкция присваивания</b> .....	<b>31</b>
<b>Глава 5. Варианты действий в программе</b> .....	<b>45</b>
5.1. Два варианта действий .....	45
5.2. Один, но не обязательный вариант действий .....	54
5.3. Три и более вариантов действий .....	59
<b>Глава 6. Повторение действий в программе</b> .....	<b>65</b>
6.1. Инструкция for .....	65
6.2. Инструкция while .....	75
6.3. Преобразование одной инструкции цикла в другую .....	83
<b>Глава 7. Программируем простейшие игры</b> .....	<b>92</b>
7.1. Игра «Чет или нечет?» .....	92
7.2. Игра «Кубик» .....	94
7.3. Игра «Отгадай число» .....	95
7.4. Игра «Карты» .....	96
7.5. Проверка знания таблицы умножения .....	98
7.6. Игра «Предметы на столе» .....	98
<b>Глава 8. Повторение повторений</b> .....	<b>102</b>
<b>Глава 9. «Обрабатываем» натуральное число</b> .....	<b>113</b>
9.1. Выделение цифр .....	113
9.2. Определение $m$ -й справа цифры числа.....	114
9.3. Определение $m$ -й слева цифры числа .....	115
9.4. Определение суммы цифр числа .....	116
9.5. Определение максимальной цифры числа.....	116
9.6. Определение минимальной цифры числа .....	117

9.7. Определение номера максимальной цифры числа при счете справа налево .....	117
9.8. Определение номера минимальной цифры числа при счете справа налево .....	118
<b>Глава 10. Типовые задачи обработки набора чисел .....</b>	<b>126</b>
10.1. Суммирование всех чисел набора.....	126
10.2. Суммирование чисел набора, которые обладают некоторыми свойствами (удовлетворяют некоторому условию) .....	127
10.3. Подсчет количества чисел набора, которые обладают некоторыми свойствами .....	128
10.4. Определение среднего арифметического тех чисел набора, которые обладают некоторыми свойствами .....	129
10.5. Определение порядкового номера некоторого значения в заданном наборе .....	131
10.6. Определение максимального значения в наборе чисел .....	132
10.7. Определение порядкового номера максимального значения в наборе чисел .....	134
10.8. Определение максимального значения тех чисел набора, которые удовлетворяют некоторому условию .....	135
10.9. Нахождение второго по величине максимального числа набора .....	136
10.9.1. Поиск числа, которое стояло бы на предпоследнем месте, если бы числа набора были отсортированы по убыванию.....	137
10.9.2. Нахождение числа набора, больше которого только максимальное.....	139
10.10. Нахождение количества максимальных элементов набора .....	140
10.11. Нахождение третьего максимума.....	141
<b>Глава 11. Работа со строками .....</b>	<b>145</b>
11.1. Общие вопросы .....	145
11.2. Типовые задачи обработки строк .....	148
11.3. Преобразования «число ↔ строка» .....	163
<b>Глава 12. Использование списков.....</b>	<b>165</b>
12.1. Общие вопросы .....	165
12.2. Заполнение списка значениями.....	166
12.3. Вывод списка на экран.....	171
<b>Глава 13. Типовые задачи обработки списков .....</b>	<b>176</b>
13.1. Расчеты .....	176
13.1.1. Суммирование элементов списка .....	176

13.1.2. Нахождение суммы элементов списка с заданными свойствами (удовлетворяющих некоторому условию) .....	177
13.1.3. Нахождение количества элементов списка с заданными свойствами .....	178
13.1.4. Нахождение среднего арифметического значения элементов списка с заданными свойствами.....	179
13.2. Поиск и отбор нужных элементов .....	181
13.2.1. Вывод на экран элементов с заданными свойствами .....	181
13.2.2. Запись всех элементов списка с заданными свойствами в другой список .....	181
13.2.3. Вывод на экран индексов элементов списка с заданными свойствами.....	182
13.2.4. Поиск индекса первого элемента списка с заданными свойствами.....	182
13.3. Работа с максимальными/минимальными элементами списка.....	183
13.3.1. Определение индекса максимального элемента списка .....	184
13.3.2. Определение количества максимальных/минимальных элементов списка .....	185
13.3.3. Нахождение второго по величине (второго максимального или второго минимального) значения списка .....	186
13.4. Перестановки элементов .....	188
13.4.1. Обмен местами двух элементов списка .....	188
13.4.2. Удаление элемента из списка .....	188
13.4.3. Циклический сдвиг элементов списка влево .....	191
13.4.4. Вставка элемента в список .....	192
13.4.5. Циклический сдвиг элементов списка вправо .....	194
13.4.6. Перестановка всех элементов списка в обратном порядке .....	195
13.5. Проверка соответствия списка в целом некоторому условию .....	197
13.5.1. Проверка факта наличия в списке элемента с заданными свойствами (удовлетворяющего некоторому условию).....	197
13.5.2. Проверка факта наличия в списке элемента с заданным значением .....	203
13.5.3. Проверка того факта, что все элементы списка соответствуют некоторому условию.....	203
13.5.4. Проверка списка на упорядоченность .....	203
13.6. Задача «Слияние (объединение) списков» .....	204
<b>Глава 14. Использование словарей.....</b>	<b>208</b>
14.1. Общие вопросы .....	208
14.2. Создание словаря.....	209
14.3. Обращение к отдельному элементу словаря .....	210
14.4. Перебор элементов словаря.....	211
14.5. Некоторые другие средства для работы со словарями.....	211
14.6. Частотный словарь .....	212

14.7. Словари со значениями разных типов.....	213
<b>Глава 15. Использование файлов .....</b>	<b>216</b>
15.1. Общие вопросы .....	216
15.2. Запись информации в файл .....	218
15.3. Чтение информации из файла.....	221
15.4. Изменение файлов .....	232
15.4.1. Запись в файл новой строки.....	232
15.4.2. Замена строки файла.....	233
<b>Глава 16. Об использовании функций.....</b>	<b>235</b>
<b>Приложение 1. Служебные (ключевые) слова языка Python.....</b>	<b>250</b>
<b>Приложение 2. Разрабатываем графический интерфейс программы .....</b>	<b>251</b>
П2.1. Общие вопросы .....	251
П2.2. Создание виджетов .....	255
П2.3. Размещаем виджеты .....	259
П2.4. Доступ к значениям в виджетах .....	262
П2.5. Изменение конфигурации виджетов .....	266
П2.6. Заставляем виджеты работать .....	268
П2.7. Итоги.....	279
П2.8. Задания для самостоятельной работы .....	280
<b>Приложение 3. Ответы к заданиям. Программы решения задач, предложенных для самостоятельной работы .....</b>	<b>284</b>
<b>Литература.....</b>	<b>395</b>



# Введение

*Когда человек хочет передвинуть гору, он начинает с того, что убирает маленькие камни.*

Восточная мудрость


Эта книга для тех, кто хочет научиться программировать на языке программирования Python. Ее отличие в том, что в ней главное – не описание языка программирования, которое представлено в большинстве книг по программированию. В данной книге рассматриваются особенности разработки программ, описываются методы решения типовых задач, встречающихся при разработке, распространенные алгоритмы, даются советы и т. п. Это учебник, в котором в доступной форме излагаются вопросы, с которыми сталкивается человек, начинающий осваивать этот непростой, но захватывающий и очень интересный процесс – программирование.

В первой части книги (главы 1–6) рассматриваются особенности разработки компьютерных программ и соответствующие инструкции языка Python. После их изучения читатель сможет разработать ряд простейших игр, описанных в главе 7. В завершающей части книги (главы 8–16) рассматриваются основные структуры данных языка Python (строки, списки, словари), вопросы, связанные с работой с файлами, а также с совершенствованием программ на основе использования функций.

По всем темам, рассмотренным в книге, приводятся большое число примеров и задач, подробная методика их решения и соответствующие программы с комментариями. Дается ряд полезных советов.

В приложении 2 описаны особенности разработки программ на языке Python с графическим пользовательским интерфейсом.

В приложении 3 приводятся ответы к заданиям и программы решения задач, предложенных в книге для самостоятельной работы.



# Глава 1.

## Понятия «алгоритм» и «программа»

*Алгоритм решения задачи* – точное описание порядка действий, которые надо выполнить для решения задачи.

Приведем ряд примеров.

Вот старинная задача: «Однажды крестьянину понадобилось перевезти через реку волка, козу и капусту. У крестьянина есть лодка, в которой может поместиться, кроме самого крестьянина, только одно существо или предмет – или волк, или коза, или капуста. Если крестьянин оставит без присмотра волка с козой, то волк съест козу; если крестьянин оставит без присмотра козу с капустой – коза съест капусту. Как крестьянину перевезти на другой берег все свое имущество в целостности и сохранности?»

Чтобы решить задачу, крестьянин должен действовать так:

- 1) погрузить на лодку козу;
- 2) перевезти ее на другой берег;
- 3) выгрузить ее;
- 4) вернуться;
- 5) погрузить на лодку волка;
- 6) перевезти его на другой берег;
- 7) выгрузить его;
- 8) погрузить на лодку козу;
- 9) вернуться с ней на первый берег;
- 10) выгрузить ее;
- 11) погрузить на лодку капусту;
- 12) перевезти ее на другой берег;
- 13) оставить капусту на этом берегу;
- 14) вернуться на первый берег;
- 15) погрузить на лодку козу;
- 16) перевезти ее на другой берег;

- 17) выгрузить козу;
- 18) остаться на втором берегу.

Возможен и второй вариант решения задачи (найдите его). Самостоятельно составьте также алгоритм решения такой задачи: «К реке подошла группа из 20 солдат, которым нужно переправиться на другой берег. Река глубокая и бурная, и ее без лодки переплыть невозможно. Солдаты увидели двух мальчиков с лодкой. Лодка такова, что в ней размещается только один солдат, только один мальчик или только два мальчика. Как солдатам переправиться?»

Можно говорить об алгоритме приготовления торта в домашних условиях (только в этом случае мы алгоритм называем «рецептом»).

Пример из математики. Представьте, что товарищ принес вам чертеж прямоугольника и просит сообщить ему площадь этой фигуры. Какие действия вы должны выполнить, чтобы решить поставленную перед вами задачу? Вот ответ:

- 1) узнать (измерить) длину одной из сторон прямоугольника и запомнить (или записать) ее;
- 2) узнать (измерить) длину второй стороны фигуры и запомнить (или записать) ее;
- 3) рассчитать площадь, перемножив два найденных значения;
- 4) сообщить результат товарищу.

Каждый алгоритм рассчитан на какого-то *исполнителя* – человека (группу людей) или устройство (робот и др.), который выполняет действия, описанные в алгоритме.

## Задание для самостоятельной работы

Представьте, что на стержень *A* надеты три диска:

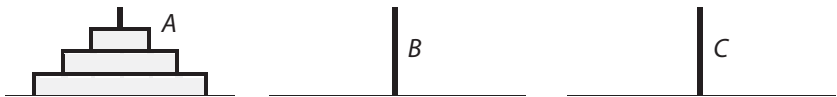


Рис. 1.1

Задача состоит в том, чтобы перенести диски со стержня *A* на стержень *C*, используя стержень *B* как промежуточный. При этом должны соблюдаться три условия:

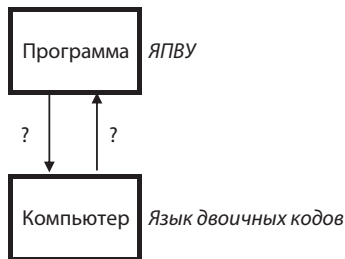
- 1) за один «ход» можно переносить лишь один диск;
- 2) нельзя класть больший диск на меньший;
- 3) снятый диск нельзя отложить в сторону – он должен быть надет на один из стержней.

Опишите алгоритм решения задачи.

Понятие *«программа решения задачи»* возникает, когда речь идет о решении задачи на компьютере. Программа решения задачи – это алгоритм решения данной задачи, записанный в виде (на языке), «понятном» данному компьютеру. А какой язык «понимает» компьютер? В нем, как в техническом устройстве, информация может быть представлена в двоичном виде – в виде последовательности условных нулей и единиц. Значит, и описание действий, которые нужно выполнить для решения (в программировании их называют «командами»), должно поступать в компьютер в двоичном виде. Поэтому в первых электронных компьютерах программы представляли из себя последовательность двоичных чисел<sup>1</sup>:

1010 1101 0101 1000 ...

Представляете, как трудно было программисту найти нужное место в программе, чтобы что-то изменить или добавить, найти и исправить ошибку? Чтобы устранить этот недостаток, были разработаны так называемые «языки программирования высокого уровня» (ЯПВУ)<sup>2</sup> – Фортран, Алгол, Бейсик и др. Программы на ЯПВУ оформляются в привычном человеку<sup>3</sup> виде: числа записываются как десятичные, а команды и другие служебные слова – на естественном (пусть и иностранном, но достаточно понятном) языке: PRINT, IF, BEGIN и т. п. Однако при этом получается противоречие – человеку-программисту удобно разрабатывать и читать программу, но компьютер такую программу не «поймет»!



**Рис. 1.2**

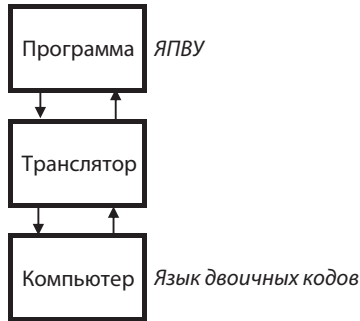
<sup>1</sup> Так как числа в двоичной системе «длинные», то для записи программ использовалась также восьмеричная или шестнадцатеричная система счисления.

<sup>2</sup> Мы не рассматриваем использование для разработки программ языков ассемблера, имевшее место до применения ЯПВУ.

<sup>3</sup> ЯПВУ поэтому так и называются, что они, так сказать, «ориентированы» на человека, в отличие от машиноориентированных языков (языков двоичных кодов и ассемблеров).



Как же поступить, чтобы компьютер мог выполнять программу на ЯПВУ? А так, как можно общаться двум людям, один из которых знает только китайский язык, а второй – только русский: нужен переводчик! Поэтому во всех современных системах программирования предусмотрен транслятор – системная программа, осуществляющая перевод прикладной программы, написанной программистом на ЯПВУ, в язык машинных кодов и ее выполнение (см. рис. 1.3)<sup>4</sup>.



**Рис. 1.3**

Существуют два вида трансляторов:

- 1) интерпретаторы;
- 2) компиляторы.

Интерпретатор читает очередную команду программы и сразу ее выполняет, не переводя всю программу в машинный код.

Компилятор читает всю программу целиком, делает ее перевод и создает законченный вариант программы на «машинном языке», который затем и выполняется.

Вопрос о преимуществах и недостатках каждого варианта транслятора здесь обсуждать не будем<sup>5</sup>, а скажем, что язык программирования Python<sup>6</sup>, как правило, работает в режиме интерпретации.

В качестве примера программы на Python приведем программу решения задачи расчета площади любого прямоугольника:

```
#Ввод размеров прямоугольника
a = int(input('Введите длину прямоугольника в см'))
b = int(input('Введите высоту прямоугольника в см'))
#Расчет площади
pl = a * b
```

<sup>4</sup> Транслятор обычно выполняет также диагностику ошибок и др.

<sup>5</sup> Подумайте над этим вопросом.

<sup>6</sup> Язык Python иногда называют языком программирования не просто высокого, а «очень высокого уровня».

```
#Вывод ответа на экран  
print('Площадь прямоугольника равна', p1, 'кв. см')
```

Итак, Python – это язык программирования, на котором разрабатываются программы. Чтобы сделать этот процесс удобным для вас и других программистов, разработаны *системы программирования*, включающие транслятор, текстовый редактор (позволяющий копировать, удалять и перемещать фрагменты программ), справочную систему, средства отладки (поиска ошибок в программе) и др. Имеется несколько вариантов языка Python и систем программирования для него.

В данной книге используется вариант CPython с версией 3.6.0 языка, который распространяется по свободной лицензии Python Software Foundation License.

## Контрольные вопросы

1. Что такое «алгоритм решения задачи»?
2. Какие вы знаете способы записи алгоритма?
3. В чем особенности алгоритма, который называют «программой»?
4. Почему языки программирования высокого уровня так называются?
5. Что такое «транслятор»? Какие функции он выполняет?
6. Какие виды трансляторов вы знаете? В чем особенность каждого вида?
7. Что включает в себя система программирования?