

Оглавление

Здравствуй, дорогой друг!	5
Глава 1. Мир компьютерных устройств	6
Глава 2. Данные и двоичный код	8
Глава 3. Логика и алгоритмы	15
Глава 4. Псевдокод	24
Глава 5. Обзор языков программирования	30
Глава 6. Пишем программы на языке Swift	33
Глава 6.1. Константы, переменные и массивы	35
Глава 6.2. Операторы	38
Глава 6.3. Циклы	43
Глава 6.4. Функции	49
Глава 6.5. Классы и объекты	53
Глава 7. Практические задачи и задания	65
Заключение	87





Здравствуй, дорогой друг!

Как здорово, что ты держишь в руках эту книгу! Ведь в ней тебя ждёт столько всего нового, неизведанного и очень интересного! Эта книга станет твоим проводником во вселенную программирования, раскроет перед тобой мир логики и алгоритмов, научит новому образу мышления и восприятия реальности, позволит по-новому взглянуть на окружающий мир. Ты узнаешь, какие бывают языки программирования и для чего они нужны, напишешь свои первые настоящие программы.

После прочтения этой книги ты сможешь выбрать свой путь в мире программирования, который тебе ближе, — от создания игр для мобильных устройств до программирования роботов для космической станции. Навыки, которые ты получишь благодаря этой книге, позволят тебе в будущем с лёгкостью изучить любой язык программирования, потому что ты уже будешь обладать мышлением программиста, а уже на чём именно писать программы — это не имеет значения. Если ты любишь придумывать что-то новое, стремишься к творчеству и хочешь принимать участие в создании будущего всего человечества, эта книга определённо для тебя.

Для тренировки практических навыков программирования мы будем использовать онлайн-компилятор, который не требует установки на компьютер, поэтому в процессе чтения книги тебе потребуется компьютер с подключением к интернету. Каждую строчку кода, каждую программу ты сразу сможешь проверить и увидеть, как это работает.

Эта книга будет максимально близка тебе, если твой возраст от 7 до 12 лет.



Глава 1

Мир компьютерных устройств

- *Компьютер и электронные устройства*
- *Что такое информация?*
- *Области применения*
- *Польза для человека*
- *Программирование как процесс со стороны человека и машины*
- *Синхронность эволюции машин и человека*

Живя в современном обществе, мы каждую минуту взаимодействуем с другими людьми через обмен информацией. Например, слова «вымой руки и учи уроки» — информация, которую можно превратить в действие. Или слова «серый пушистый кот» — информация, описывающая пушистость кота и его цвет. Мало того — мы взаимодействуем со множеством электронных устройств тоже через обмен информацией. Утром слышим сигнал будильника — этот сигнал и есть информация о том, что нужно просыпаться. А вечером настраиваем будильник на утренний подъём — будильник получает информацию от нас, во сколько нужно зазвонить. Мы взаимодействуем с окружающим миром точно так же, как и с будильником, — через обмен информацией.

А как же электронные устройства получают информацию от нас? Через нажатие кнопок на клавиатуре, через касание экрана, через микрофон или камеру. После получения информации от нас устройство обрабатывает её и выдаёт результат: выводит текст или картинку на экран, воспроизводит звук — выполняет свою задачу.

Любое взаимодействие с электронными устройствами — это обмен информацией, и, чтобы устройство понимало, чего мы от него хотим, оно должно уметь получать и «обдумывать» информацию от нас. Таким образом, любое электронное устройство следует научить принимать и обрабатывать информацию, это и называется программирование. В результате программирования «рождается» программа. Программ-

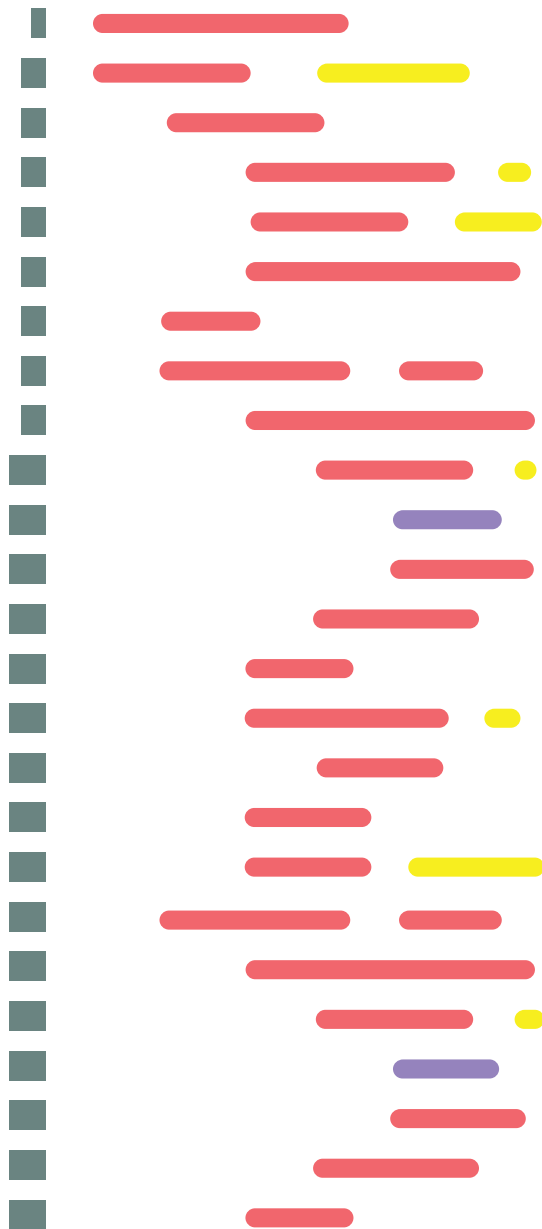


Также существует ещё множество специфических устройств ввода информации, например, для получения информации о состоянии полёта бортовой компьютер самолёта использует датчики высоты, наклона, компас и так далее.

Материнская плата является связующим звеном для всех компонентов компьютера. Называется она так потому, что объединяет все отдельные устройства в одно — персональный компьютер.

Как же «думает» центральный процессор компьютера? Процессор мыслит двоичными кодами, а именно символами, у которых бывает два состояния — 1 или 0. Включено или выключено. Есть сигнал или нет. Всю поступающую и сохраняемую информацию компьютер переводит в двоичные коды.

В самом начале эпохи компьютеров для ввода программы человеку приходилось общаться с компьютером практически посредством двоичных кодов, это было очень неудобно, программы представляли собой очень много текста, и программистам было достаточно трудно работать с такими программами. Чтобы упростить себе задачу, программисты придумали языки программирования. Язык программирования — это язык, который больше похож на человеческий, на котором удобно писать программы. Но ведь компьютер не поймёт, что написано на таком человеческом языке программирования? Верно, для этого существует «компилятор», это тоже такая программа, которая переводит человеческий язык в понятный для компью-



Строки 2 и 3 — мы объявили переменные «Число1» и «Число2», тип у этих переменных — целое число. Ты спросишь, что такое «переменная»? Можно представить переменную контейнером, в котором будет храниться какое-то значение. В нашем примере мы описали, что «Число1» — это переменная типа «Целое число». Это значит, что в «Число1» мы можем положить только целое число.

Строка 4 — переменная «Максимум», тип тоже «Целое число», в этой переменной мы будем хранить максимальное значение.

Строка 5 — функция ВВОД (Число1, Число2) — это функция ввода чисел пользователем, т. е. пользователь на этом шаге введёт значения переменных «Число1» и «Число2».

Строка 6 — собственно, это и есть поиск максимального числа. В этой строке выполняется проверка «если Число1 больше, чем Число2, то в Максимум сохранить Число1, иначе — в Максимум сохранить Число2». Как ты видишь, все слова программы написаны на русском языке и легко понять, что происходит в той или иной строке.

Строка 7 — выводим найденное максимальное число, показываем пользователю.

Строка 8 — завершение выполнения программы.

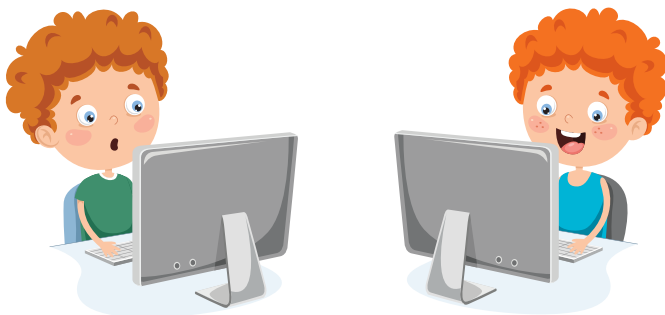
Что будет, если при запросе двух чисел ты введёшь числа «10» и «20»? Какое число выдаст программа при выполнении? Правильно, программа выдаст число «20», потому что оно больше, чем число «10», и это мы

Практические задачи и задания

Теперь у тебя есть фундаментальные знания в программировании, и, чтобы лучше усвоить процесс программирования, почувствовать себя программистом, мы будем писать программы. Потому что практика и только практика ведёт к совершенству!

Начнём с простых программ, мы будем давать тебе задание, а ты пишешь программы. Ниже будет расположен ответ (готовая программа) с подробными комментариями, но не подглядывай в ответ, пока не напишешь сам. Также в описании задач мы будем давать тебе подсказки, а первую задачу напишем и разберём вместе. Помни, что правильных вариантов решения задач в программировании бесконечно много, и ты можешь смело придумать свой вариант решения.

Рекомендуем писать программы сразу в среде онлайн-компилятора Swift, мы используем вот этот онлайн-компилятор: <http://online.swiftplayground.run>. Для запуска программы нажми синюю кнопку Run наверху слева.



Глава 7



Задача 1. Поиск максимума в массиве

В зоопарке решили устроить гонки животных. Для этого выбрали четырёх зверей: зайца, страуса, лисицу и волка. Скорость каждого животного известна:

Заяц: 10 км/ч

Страус: 13 км/ч

Лисица: 8 км/ч

Волк: 17 км/ч

Какое животное победит?

Решение

Очевидно, что победит животное, которое быстрее всего бегаёт. По условию задачи видно, что это волк, потому что его скорость 17 км/ч. Значит, для того чтобы программа смогла найти победителя, нужно найти животное, у которого самая большая максимальная скорость.

И как же будет выглядеть наша программа?

Т.к. у нас четыре зверя, то нам потребуется массив, в котором мы будем хранить значения скоростей животных. Затем нам нужно будет найти максимум в этом массиве. И затем вывести животное, которое бегаёт быстрее всех. Для простоты решения договоримся, что животные в массиве следуют в таком порядке: заяц, страус, лисица и волк. Мы написали код, и программа в онлайн-компиляторе выглядит вот так:

```
let animalsSpeed = [10, 13, 8, 17]
var maxSpeed = 0
var maxIndex = 0

for i in 0..<animalsSpeed.count {
  if animalsSpeed[i] > maxSpeed {
    maxSpeed = animalsSpeed[i]
    maxIndex = i
  }
}

if maxIndex == 0 { print("Победил заяц") } else
if maxIndex == 1 { print("Победил страус") } else
```