

Алгоритмизация и программирование

Лекция 1

Введение

Любой человек ежедневно встречается с множеством повседневных и профессиональных задач. Для решения многих из них существуют определенные правила (инструкции, предписания), объясняющие, как решать определенную задачу. В процессе решения можно применять готовые правила или формулировать собственные.

Чем точнее и понятнее описаны правила решения задач, тем быстрее человек овладеет ими и будет эффективнее их применять. Решение многих задач человек передает техническим устройствам –автоматам, роботам и т. д. Их применение предъявляет очень строгие требования к точности описания правил и последовательности выполнения действий. Поэтому разрабатываются специальные алгоритмы для четкого и строгого описания различных правил.

-
- Алгоритмизация — это раздел информатики, изучающий методы и приемы построений алгоритма, а также их свойства. Она является основным, базовым компонентом компьютерной грамотности в современном компьютерном мире.

13.10.2023

Алгоритмизация — это раздел информатики, изучающий методы и приемы построений алгоритма, а также их свойства. Она является основным, базовым компонентом компьютерной грамотности в современном компьютерном мире.

Для достижения положительных результатов важную роль играет умение разрабатывать оптимальный алгоритм решения поставленной задачи,

что требует от исполнителя наличия определенных навыков алгоритмизации и системного анализа, а также знания математики, физики, химии, экономики и других дисциплин.

1. Понятие алгоритма

ПОНЯТИЕ АЛГОРИТМА

- Алгоритм – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

Алгоритм – это точное и понятное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи.

ИЗ ИСТОРИИ



- Само слово «алгоритм» происходит от имени средневекового ученого аль Хорезми.
- Родился в 783 году.
- Первый известный арабский трактат по алгебре.
- Algorithmi – латинское написание имени аль-Хорезми.

13.10.2023

Название "алгоритм" произошло от латинской формы имени среднеазиатского математика аль-Хорезми - Algorithmi.

- Исполнитель алгоритма — это система, способная выполнить действия, предписываемые алгоритмом

13.10.2023

Исполнитель алгоритма — это некоторая абстрактная или реальная (техническая, биологическая или биотехническая) система, способная выполнить действия, предписываемые алгоритмом.

Исполнителя характеризуют: среда, элементарные действия, система команд, отказы.

Среда (или обстановка) — это "место обитания" исполнителя.

Каждый исполнитель может выполнять команды только из некоторого строго заданного списка - *системы команд* исполнителя.

Для каждой команды должны быть заданы условия применимости (в каких состояниях среды может быть выполнена команда) и описаны результаты выполнения команды. После вызова команды исполнитель совершает соответствующее *элементарное действие*. *Отказы* исполнителя возникают, если команда вызывается при недопустимом для нее состоянии среды.

В информатике универсальным исполнителем алгоритмов является компьютер.

2. Свойства алгоритмов

Можно выделить следующие *основные свойства алгоритмов*:

СВОЙСТВА АЛГОРИТМОВ

- 1) Понятность для исполнителя
- 2) Дискретность (прерывность, отдельность)
- 3) Определенность
- 4) Результативность
- 5) Массовость

13.10.2023

1) *Понятность* для исполнителя - т.е. исполнитель алгоритма должен знать, как его выполнять.

2) *Дискретность* (прерывность, отдельность) - т.е. алгоритм должен представлять процесс решения задачи как последовательное выполнение простых или ранее определенных шагов.

3) *Определенность* - т.е. каждое правило алгоритма должно быть четким, однозначным и не оставлять места для разночтений.

4) *Результативность* (или конечность). Это свойство состоит в том, что алгоритм должен приводить к решению задачи за конечное число шагов.

5) *Массовость* - означает, что алгоритм решения задачи разрабатывается в общем виде, т.е. он должен быть применим для некоторого класса задач, различающихся лишь исходными данными. При этом исходные данные могут выбираться из некоторой области, которая называется *областью применимости алгоритма*.

3. Формы представления алгоритмов

Наиболее распространенными *формами представления алгоритмов* являются: словесная, графическая, псевдокоды и программная.

1) Словесная форма записи представляет собой описание последовательных этапов обработки данных на естественном языке (например, на русском).

СЛОВЕСНАЯ ФОРМА ПРЕДСТАВЛЕНИЯ АЛГОРИТМА

ЗАДАЧА:

Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

АЛГОРИТМ:

- 1) задать два числа;
- 2) если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма;
- 3) определить большее из чисел;
- 4) заменить большее из чисел разностью большего и меньшего из чисел;
- 5) повторить алгоритм с шага 2.

13.10.2023

Пример. Записать алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

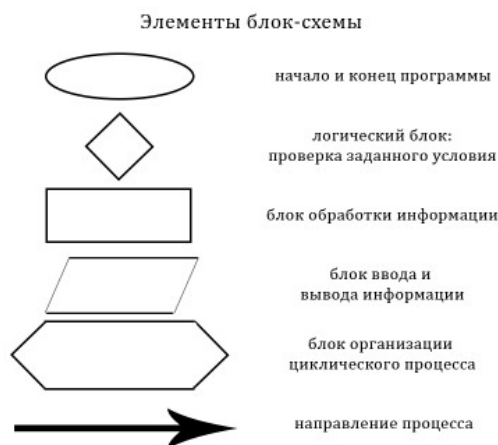
Алгоритм: 1) задать два числа; 2) если числа равны, то взять любое из них в качестве ответа и остановиться, в противном случае продолжить выполнение алгоритма; 3) определить большее из чисел; 4) заменить большее из чисел разностью большего и меньшего из чисел; 5) повторить алгоритм с шага 2.

Описанный алгоритм применим к любым натуральным числам и должен приводить к решению поставленной задачи за конечное число шагов.

Словесный способ не имеет широкого распространения, поскольку такие описания:

- а) строго не формализуемы;
- б) страдают многословностью записей;
- в) допускают неоднозначность толкования отдельных предписаний.

ГРАФИЧЕСКИЙ СПОСОБ ОПИСАНИЯ АЛГОРИТМОВ



13.10.2023

2) Графический способ представления алгоритмов является более компактным и наглядным по сравнению со словесным. При графическом исполнении алгоритм изображается в виде последовательности связанных между собой функциональных блоков, каждый из которых соответствует выполнению одного из действий. Такое графическое представление называется схемой алгоритма или **блок-схемой**. В блок-схеме каждому типу действий соответствует геометрическая фигура, называемая *блочным символом*. Блочные символы соединяются *линиями переходов*, определяющими очередность выполнения действий.

3) Псевдокод представляет собой систему обозначений и правил, предназначенную для единообразной записи алгоритмов. Он занимает промежуточное место между естественным и формальным языками.

ПСЕВДОКОД

Пример:

- 1) задать два числа x и y ;
- 2) ЕСЛИ $x=y$, ТО НОД= x и КОНЕЦ;
- 3) ЕСЛИ $x>y$, ТО $x=x-y$, ИНАЧЕ $y=y-x$;
- 4) ПЕРЕЙТИ в пункт 2.

13.10.2023

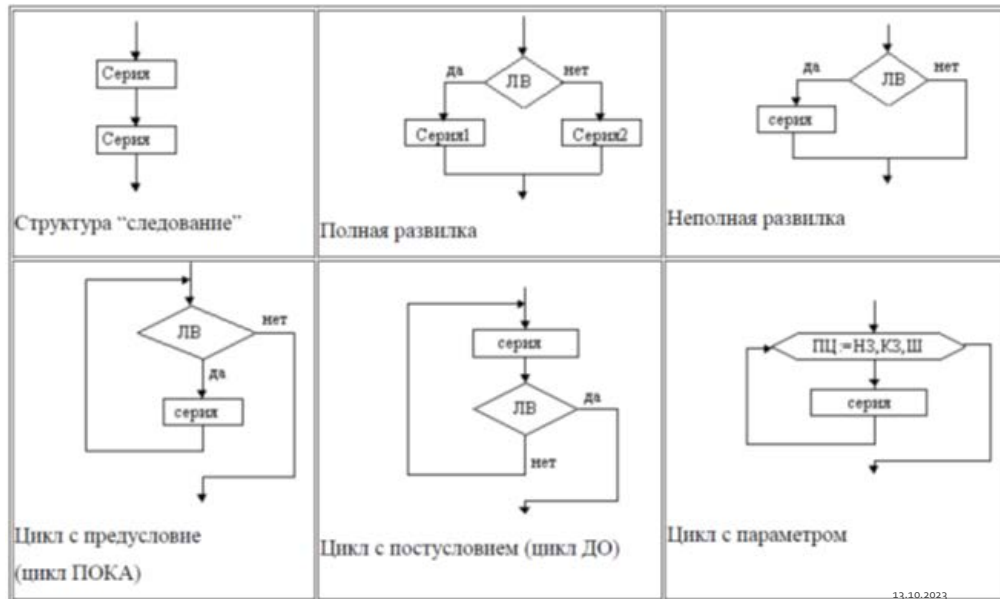
9

С одной стороны, он близок к обычному естественному языку, поэтому алгоритмы могут на нем записываться и читаться как обычный текст. С другой стороны, в псевдокоде используются *служебные слова* и математическая символика, что приближает запись алгоритма к общепринятой математической записи. Служебные слова выделяются в печатном тексте жирным шрифтом, а в рукописном тексте подчеркиваются для того, чтобы их можно было отличить от остального текста.

Пример. 1) задать два числа x и y ; 2) ЕСЛИ $x=y$, ТО НОД= x и КОНЕЦ; 3) ЕСЛИ $x>y$, ТО $x=x-y$, ИНАЧЕ $y=y-x$; 4) ПЕРЕЙТИ в пункт 2.

- 4) Программная форма представляет собой тексты программ, написанных на различных языках программирования.

На слайде приведены графические обозначения на блок-схемах.



13.10.2023

10

На схемах СЕРИЯ обозначает один или несколько любых операторов; УСЛОВИЕ есть логическое выражение (ЛВ) (если его значение ИСТИНА, переход происходит по ветви ДА, иначе

— по НЕТ). На схеме цикла с параметром использованы обозначения: ПЦ — параметр цикла, НЗ

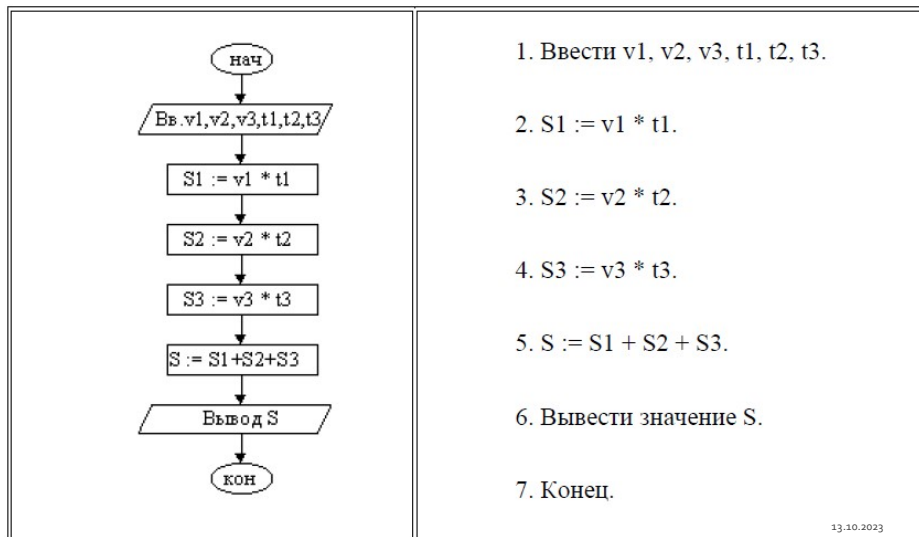
— начальное значение параметра цикла, КЗ — конечное значение параметра цикла, Ш — шаг изменения параметра цикла.

Начало и конец алгоритма на блок-схемах обозначают овалом, вводимые и выводимые переменные записываются в параллелограмме.

Виды алгоритмов

Линейные алгоритмы

ЛИНЕЙНЫЕ АЛГОРИТМЫ



13.10.2023

11

Простейшие задачи имеют линейный алгоритм решения. Это означает, что он не содержит проверок условий и повторений.

Пример 1. Пешеход шел по пересеченной местности. Его скорость движения по равнине 1 км/ч, в гору — 2 км/ч и под гору — 3 км/ч. Время движения соответственно 1, 2 и 3 ч. Какой путь прошел пешеход?

Для проверки работоспособности алгоритма необходимо задать значения входных переменных, вычислить конечный результат по алгоритму и сравнить с результатом ручного счета.

Разветвленный алгоритм

1. Задачи на ветвление

Разветвляющимся называется такой **алгоритм**, в котором в зависимости от истинности или ложности заданного условия выбирается один из нескольких возможных вариантов (путей) вычислительного процесса. Каждый такой вариант называется **ветвью алгоритма**.

Возможных вариантов может быть два или три.

Ветвление - такая форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна, либо другая последовательность действий.

Признаком разветвляющегося алгоритма является наличие операций проверки *условия*.

Условие – это выражение логического типа. Оно может включать в себя константы, имена переменных, арифметические операции, операции отношения, логические операции, скобки.

Условие может быть истинным или ложным, то есть принимать одно из двух значений: ИСТИНА или ЛОЖЬ.

Различают два вида условий: простые и составные.

Простым условием называется выражение, составленное из двух арифметических выражений или двух текстовых величин, связанных одним из знаков: $<$, \leq , $>$, \geq , $=$, \neq

Такое условие часто называют операцией отношения.

Составными условиями называются условия, состоящие из нескольких простых и соединенные знаками логических операций И, ИЛИ, НЕ.

ПРИМЕРЫ составных условий.

$(A > 5)$ И $(B < 2)$

$(X = Y)$ ИЛИ $(50 \leq C - 1)$

НЕ («собака» = «конура») И («кошка» = D)

Составное условие вида «X И Y» истинно тогда и только тогда, когда истинны оба условия X и Y, в остальных случаях – ложно.

Составное условие вида «X ИЛИ Y» истинно тогда, когда истинно хотя бы одно условие X или Y. Условие вида «НЕ X» истинно, если X ложно, и наоборот.

Для обозначения операции проверки условия в блок-схеме алгоритма используется блок, изображаемый ромбом, внутри которого указывается проверяемое условие. Этот блок имеет один вход и два или три выхода.

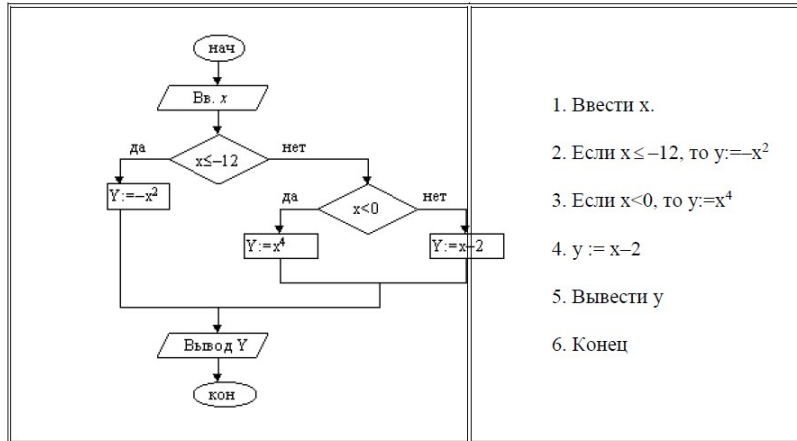
Если выходов два, то они обозначаются, ДА и НЕТ. Если условие истинно, то вычислительный процесс «идет» по ветви ДА; если ложно – по ветви НЕТ.

Если выходов три (это чаще всего бывает, когда в условии переменная или выражение сравниваются с нулем), то они обозначаются: >0 , <0 , $=0$. При написании разветвляющихся программ используется условный оператор.

Пример 1. Вычислить значение функции

ВЫЧИСЛИТЬ ЗНАЧЕНИЕ ФУНКЦИИ

$$y = \begin{cases} -x^2 & \text{при } x \leq -12, \\ x^4 & \text{при } -12 < x < 0, \\ x-2 & \text{при } x \geq 0. \end{cases}$$



1. Ввести x.
2. Если $x \leq -12$, то $y := -x^2$
3. Если $x < 0$, то $y := x^4$
4. $y := x - 2$
5. Вывести y
6. Конец

13.10.2023

14

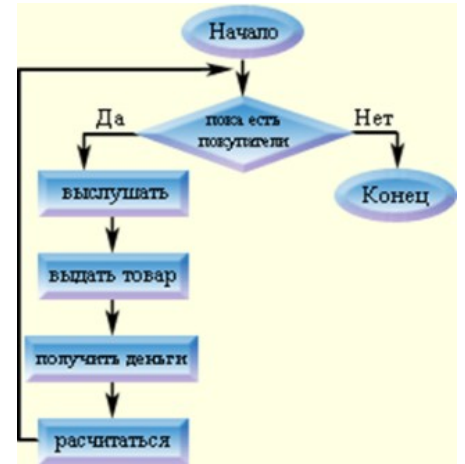
При тестировании алгоритмов с развилкой необходимо подбирать такие исходные данные, чтобы можно было проверить все ветви. В приведенном выше примере должно быть по крайней мере три тестовых набора.

Циклические алгоритмы

Циклическим называют алгоритм, в котором получение результата обеспечивается многократным выполнением одних и тех же операций. Иными словами, циклическим называют алгоритм, в котором определенная последовательность действий повторяется либо заданное количество раз, либо до тех пор, пока не выполнится некоторое условие. Повторяющаяся последовательность действий называется телом цикла.

Цикл - такая форма организации действий, при которой одна и та же последовательность действий (тело цикла) совершается несколько раз (или ни разу) до тех пор, пока выполняется некоторое условие.

ЦИКЛ С ПРЕДУСЛОВИЕМ (ПОКА)



13.10.2023

Цикл с предусловием (цикл-пока)

Выполнение начинается с проверки условия, поэтому этот цикл и называется циклом с предусловием. Переход к выполнению серии действий (телу цикла) осуществляется тогда, когда условие истинно, в противном случае происходит выход из цикла. Условие в данном случае называется условием продолжения цикла. Возможен такой вариант, что тело цикла не выполнится ни разу. Для того, чтобы цикл не повторялся бесконечное число раз, в теле цикла необходимо изменять значение параметра цикла так, чтобы за конечное количество «шагов» условие стало ложным.

ЦИКЛ С ПОСТУСЛОВИЕМ (ЦИКЛ-ДО)



13.10.2023

16

Цикл с постусловием (цикл-до)

Цикл с постусловием начинается с выполнения тела цикла и лишь затем проверяется условие. Тело цикла обязательно выполнится хотя бы один раз. Если условие ложно, то повторяется выполнение тела цикла, если истинно – то осуществляется выход из цикла. Поэтому данное условие называют условием прекращения цикла. Чтобы не было заикливания, в теле цикла необходимо изменять значение параметра цикла так, чтобы за конечное количество «шагов» условие стало истинным.

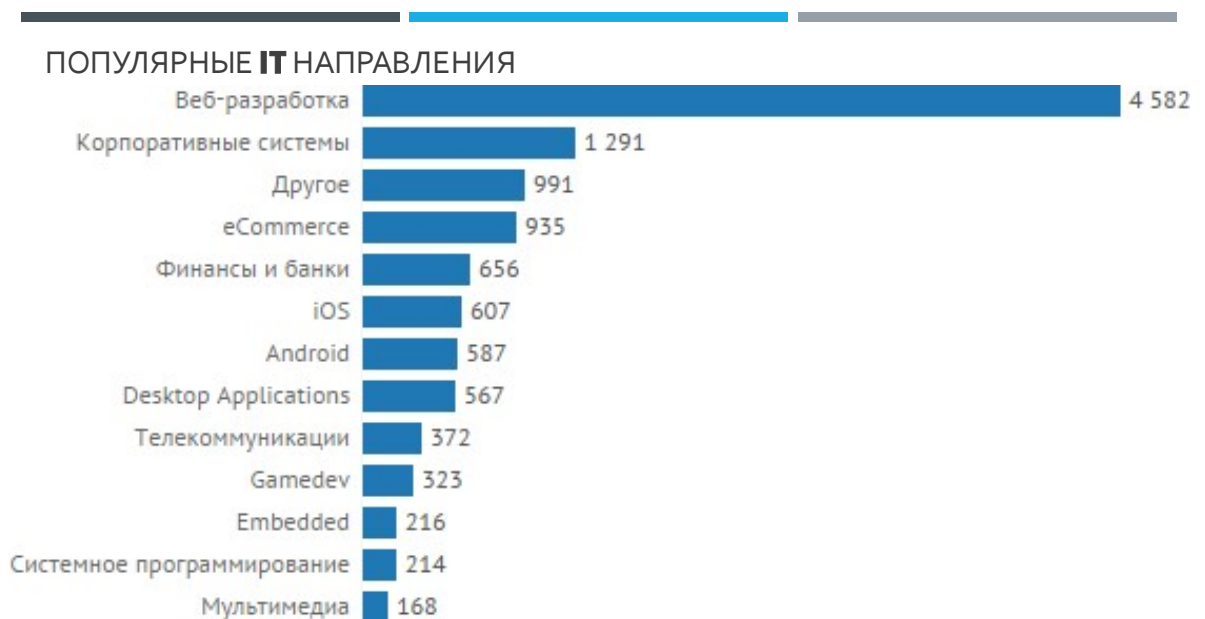
Если какие-либо операторы необходимо выполнить несколько раз, то их не переписывают каждый раз заново, а организуют цикл.

Итак, алгоритм – это точно определенное описание способа решения задачи в виде последовательности действий. Такое описание называется *формальным*, а процесс – *формализацией*.

Для представления алгоритма в виде, понятном компьютеру, служат *языки программирования*. Изучение программирования разумно начинать, собственно, с разработки алгоритмов, не акцентируя первоначально внимания на записи алгоритма на том или ином языке программирования. После написания программы с помощью *трансляторов* либо переводится в *машинный код* (последовательность чисел, с которой работает процессор), либо исполняется

Обзор языков программирования

В сфере IT есть разные направления разработки, вот самые распространенные:



Веб-разработка — это разработка сервисов и инструментов, которыми мы пользуемся в интернете;

Системная разработка — разработка программ, с помощью которых компьютер выполняет задачи: операционных систем, драйверов;

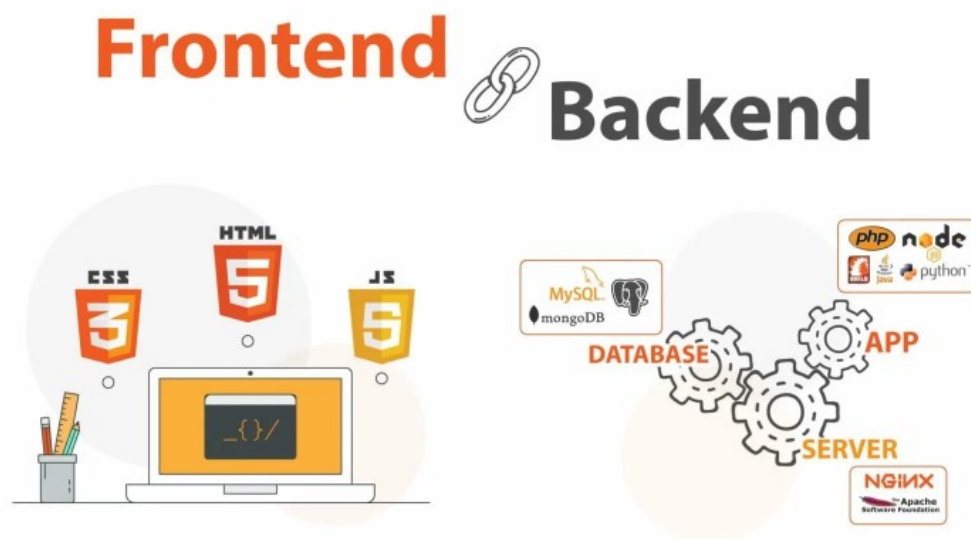
Десктоп-разработка — разработка программ для запуска под управлением операционной системы;

Геймдев — разработка игр;

Мобильная разработка — разработка приложений для телефонов и планшетов;

Data Science и анализ данных — хранение, обработка и анализ больших объемов данных.

Мир веб-разработки — если говорить только о программистах без околорайтишных специальностей — делится на две категории: фронтенд и бэкенд.



13.10.2023

18

Фронтенд (англ. frontend — «внешняя часть») — это всё, что видит пользователь на экране или чего касается курсором мыши, например кнопка на сайте. Когда пользователь нажимает на кнопку, в игру вступает бэкенд (англ. backend — «внутренняя часть»).

Его задача — организовать взаимодействие с базами данных и другими серверами, настроить поиск нужной информации и отправить её обратно пользователю.

Фронтенд – визуальная сторона, бэкенд – все, что под капотом.

Фронтенд-разработчики создают интерфейс для взаимодействия с приложением, задают структуру веб-страницы на языке разметки HTML, определяют её внешний вид с помощью CSS, а на языке программирования JavaScript (JS) добавляют динамический контент, например всплывающие окна.

Помимо JavaScript используют язык TypeScript, расширяющий возможности JavaScript, и фреймворки: React, Vue или Angular. Двадцать лет назад, чтобы называть себя полноценным фронтендером, достаточно было знать HTML и CSS.

Сегодня из-за обилия фреймворков и технологий фронтенд по сложности догнал бэкенд. Бэкенд-разработчики отвечают за логику работы сервера: сохранение данных в базу, хранение файлов, механика авторизации. Бэкенд-разработчики пишут на разных языках программирования: JavaScript, C#, Go, PHP, Python, Java.

В 2009 году появилась среда Node.js, позволяющая разрабатывать бэкенд на JavaScript. Зная один язык, программист способен выполнять задачи как фронтенда, так и бэкенда. Такой специалист-универсал называется фулстек-разработчиком.

HTML и CSS

Пример HTML кода

```

1 <!DOCTYPE html>
2 <html lang="ru">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>WebDevkin | Новости о фронтенде</title>
8   <link href="css/main.css" rel="stylesheet">
9 </head>
10
11 <body>
12   <h2>WebDevkin. Новости фронтенда с bombardir.ru</h2>
13   <br />
14   <table>
15     <tbody>
16       <tr>
17         <td class="news">
18           <h3>Ламное новости</h3>
19           <ul>
20             <li><php foreach ($news as $item) {>
21               <li>
22                 <a href="">php echo $item['link']; >> target="_blank"</php echo $item['title']; >></a>
23               </li>
24             </li>
25           </ul>
26         </td>
27       </tr>
28     </tbody>
29   </table>
30
31   <td class="articles">
32     <h3>Актуальные статьи</h3>
33     <ul>
34       <li><php foreach ($articles as $item) {>
35         <li>
36           <a href="">php echo $item['link']; >> target="_blank"</php echo $item['title']; >></a>
37           <br>
38           <img src="">php echo $item['image']; >> alt=""
39           <br>
40           <php echo $item['text']; >>
41         </li>
42       </li>
43     </ul>
44   </td>
45 </tr>
46 </tbody>
47 </table>
48 </body>
49 </html>

```

13.10.2023

19

HTML (Hyper Text Markup Language) — язык разметки, который задаёт структуру веб-страницы с помощью тегов. Теги указывают браузеру, как отображать содержимое: «это заголовок», «это абзац», «это картинка», «это ссылка».

ПРИМЕР CSS КОДА

```

body {
  background: #FFF;
  color: #000;
  font-family: Arial, sans-serif;
  font-size: 14px;
}

#header {
  background: #F5DEB3;
  width: 100%;
  height: 55px;
}

#container {
  background: #FFD700;
  margin: auto auto;
  text-align: center;
  width: 80%;
  height: 400px;
}

```

13.10.2023

20

CSS (Cascading Style Sheets) — каскадные таблицы стилей, которые описывают внешний вид страницы: шрифт, фон, цвета, видимость элементов и многое другое. К каждому элементу страницы может применяться несколько стилей, а каскадность определяет порядок применения стилей к элементу — например, сначала увеличить размер шрифта, потом сделать текст курсивом, а в конце покрасить его в синий цвет.

HTML – скелет веб-страницы, CSS – отвечает за внешний вид

JavaScript

ПРИМЕР JAVASCRIPT КОДА

```

const car1 = {
  maker: 'Ford',
  model: 'Fiesta',
  drive() {
    console.log(`Driving a ${this.maker} ${this.model} car!`)
  }
}

const anotherCar = {
  maker: 'Audi',
  model: 'A4'
}

car1.drive.bind(anotherCar)()
//Driving a Audi A4 car!

const car2 = {
  maker: 'Ford',
  model: 'Fiesta'
}

const drive = function(kmh) {
  console.log(`Driving a ${this.maker} ${this.model} car at ${kmh} km/h!`)
}

drive.call(car2, 100)
//Driving a Ford Fiesta car at 100 km/h!
drive.apply(car2, [100])
//Driving a Ford Fiesta car at 100 km/h!

```

13.10.2023

21

JavaScript — язык-монополист во фронтенд-разработке. С его помощью программисты оживляют страницы, добавляя динамический контент: по щелчку мыши меняется фон, при скролле вниз подгружается новый контент, проверяются данные в формах при покупке авиабилетов и всплывают уведомления в соцсетях.

JavaScript — язык фронт- и бэкенда одновременно.

Чтобы каждый раз не писать с нуля базовую логику приложения, разработчики используют фреймворки — заранее написанный код или шаблоны, задающие архитектуру приложения. JS-фреймворки, которые чаще всего используют разработчики: React, Angular и Vue.js

Преимущества языка:

Подходит для клиентской и серверной стороны.

Огромное количество библиотек.

Низкий порог входа.

Популярность.

Особенности языка:

Слабая типизация, которая позволяет складывать строку с числом. Эта особенность породила огромное количество мемов.

Запутанный синтаксис.

Python

ПРИМЕР ПЫТОНКОДА

```
import numpy
from numpy.random import random_integers as rand
import matplotlib.pyplot as pyplot

def maze(width=81, height=51, complexity=.75, density=.75):
    # Only odd shapes
    shape = ((height // 2) * 2 + 1, (width // 2) * 2 + 1)
    # Adjust complexity and density relative to maze size
    complexity = int(complexity * (5 * (shape[0] + shape[1])))
    density = int(density * (shape[0] // 2 * shape[1] // 2))
    # Build actual maze
    Z = numpy.zeros(shape, dtype=bool)
    # Fill borders
    Z[0, :] = Z[-1, :] = 1
    Z[:, 0] = Z[:, -1] = 1
    # Make aisles
    for i in range(density):
        x, y = rand(0, shape[1] // 2) * 2, rand(0, shape[0] // 2) * 2
        Z[y, x] = 1
        for j in range(complexity):
            neighbours = []
            if x > 1: neighbours.append((y, x - 2))
            if x < shape[1] - 2: neighbours.append((y, x + 2))
            if y > 1: neighbours.append((y - 2, x))
            if y < shape[0] - 2: neighbours.append((y + 2, x))
            if len(neighbours):
                y_, x_ = neighbours[rand(0, len(neighbours) - 1)]
                if Z[y_, x_] == 0:
                    Z[y_, x_] = 1
                    Z[y_ + (y - y_) // 2, x_ + (x - x_) // 2] = 1
                    x, y = x_, y_
    return Z

pyplot.figure(figsize=(10, 5))
pyplot.imshow(maze(80, 40), cmap=pyplot.cm.binary, interpolation='nearest')
pyplot.xticks([], pyplot.yticks([]))
pyplot.show()
```

Python — простой высокоуровневый язык с большими возможностями. Язык хорош для начинающих разработчиков, подходит для разных задач и

применяется во многих сферах, от машинного обучения до создания игр и проведения научных исследований.

Философия Python: красивое лучше, чем уродливое, явное лучше, чем неявное, простое лучше, чем сложное, а сложное лучше, чем запутанное.

Преимущества языка:

Лаконичность: на Python нужно писать гораздо меньше кода для выполнения команд.

Python применяют в веб-разработке, в машинном обучении, аналитике данных, играх, научных исследованиях.

Востребованность на рынке: на Python пишут и в небольших студиях, и в огромных корпорациях.

Особенности языка:

Машинное обучение и наука о данных требуют знания соответствующих разделов математики.

C++

ПРИМЕР C++ КОДА

```
// 01-01.cpp - пример простейшей программы
#include <iostream> // стандартный модуль C++
#include <stdio.h> // стандартный модуль C
using namespace std; // пространство имен
int main()
{
    setlocale(LC_ALL, "Russian");
    float r,h,V;
    const float pi=3.14159;
    cout<<"РАСЧЕТ ОБЪЕМА КОНУСА"<<endl;
    cout<<"введите радиус: ";
    cin>>r;
    cout<<"введите высоту: ";
    cin>>h;
    V=(1./3.)*pi*r*r*h; // 1: деление
    cout<<"объем равен: "<<V;
    getchar();
    return 0;
}
```

13.10.2023

23

C++ или просто «плюсы» — объектно-ориентированный язык, на котором пишут сложные сервисы, требующие скорости и производительности. Изначально C++ создавался, как усовершенствованный вариант языка C, который сохранял бы с ним обратную совместимость. При этом добавились

ООП, обработка исключений, перегрузка операторов и другие полезные нововведения.

Плюсы вездесущи. Питонисты, тренирующие нейронные сети, используют библиотеки, написанные на низкоуровневых языках C и C++. Плюсы дали жизнь таким играм, как Fortnite, Borderlands, Mass Effect, и сейчас на нём разрабатывают S.T.A.L.K.E.R. 2. Часть игрового движка Unity написана на плюсах, а браузер Google Chrome — целиком. С помощью C++ пишут драйверы и сложные программные интерфейсы.

C++ даёт полный контроль над управлением памятью. Это можно рассматривать и как преимущество, и как недостаток, поскольку разработчик получает контроль над ресурсами, но требует высокой квалификации.

Преимущества языка:

C++ совмещает в себе качества как низкоуровневых, так и высокоуровневых языков.

Для C++ разработчиков написано множество библиотек.

C++ можно использовать в разработке чего угодно: от беспилотных автомобилей до веб-браузеров, видеоигр и микроконтроллеров.

После изучения C++ другие языки будут казаться проще.

Особенности языка:

Более сложный для изучения язык, чем Python.

Сложный синтаксис.

C#

ПРИМЕР C# КОДА

```
// программа вводит с консоли число
// и выводит его на консоль
#include <iostream>
using namespace std;
int main()
{
    int a;
    cout << "input number\n";
    cin >> a;
    cout << "number = " << a << "\n";
    return 0;
}
```


 комментарий

24

C#, или «си-шарп», — объектно-ориентированный язык с автоматическим управлением памятью, разработанный компанией Microsoft для платформы .NET.

На C# пишут программы для экосистемы Microsoft и веб-приложения. Для машинного обучения используют фреймворк ML.NET, а игры создают на движке Unity. Написали код и теперь нужно перенести на другую платформу? Фреймворк Mono поможет с переездом на все популярные платформы: Windows, macOS, Linux, Android и iOS.

C# – уверенный середнячок с большим потенциалом.

Преимущества языка:

Наличие ООП;

Кроссплатформенность;

Автоматическая сборка мусора и управление памятью;

Много библиотек и готовых решений;

Особенности языка:

C# менее гибкий, так как в основном зависит от платформы .NET.

Java

ПРИМЕР JAVAКОДА

```

        if (lastName != null ? !lastName.equals(person.lastName)
            : person.lastName != null) {
            return false;
        }
        return true;
    }

    public int hashCode() {
        int result = firstName != null ? firstName.hashCode()
            : 0;
        result = 31 * result +
            (lastName != null ? lastName.hashCode() : 0);
        return result;
    }

    public String toString() {
        return "Person(" + firstName + "," + lastName + ")";
    }
}

```

13.10.2023

24

Java — кроссплатформенный язык с поддержкой ООП. Код на Java одинаково запускается и на умном холодильнике, и на микроконтроллере, и на кнопочном телефоне. Всё дело в Java Virtual Machine (JVM) — посреднике между кодом и железом.

Java работает по принципу: «Код один, платформы разные».

Компилятор Java создаёт из написанного программистом кода байт-код. Виртуальная машина, учитывая аппаратную платформу, на которой запускается программа, загружает байт-код, проверяет и выполняет его. Получается, что один и тот же код одинаково просто запускается на разных платформах. Плюс виртуальная машина автоматически управляет памятью.

Преимущества языка:

Кроссплатформенность — Java работает на большом количестве операционнок.

Наличие ООП.

Большое сообщество.

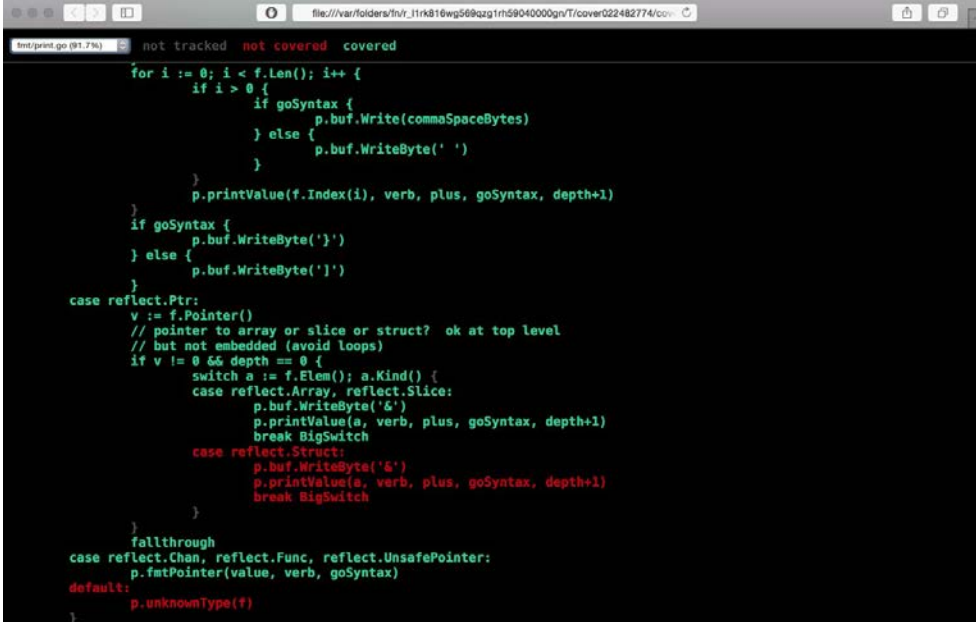
Особенности языка:

Дополнительный слой в виде виртуальной машины немного снижает производительность.

Многословный код — иногда нужно написать больше кода, чем на других языках.

Go

ПРИМЕР GO КОДА



```

not tracked not covered covered
for i := 0; i < f.Len(); i++ {
    if i > 0 {
        if goSyntax {
            p.buf.WriteByte(',')
        } else {
            p.buf.WriteByte(' ')
        }
    }
    p.printValue(f.Index(i), verb, plus, goSyntax, depth+1)
    if goSyntax {
        p.buf.WriteByte(',')
    } else {
        p.buf.WriteByte(' ')
    }
}
case reflect.Ptr:
    v := f.Pointer()
    // pointer to array or slice or struct? ok at top level
    // but not embedded (avoid loops)
    if v != 0 && depth == 0 {
        switch a := f.Elem(); a.Kind() {
            case reflect.Array, reflect.Slice:
                p.buf.WriteByte('&')
                p.printValue(a, verb, plus, goSyntax, depth+1)
                break BigSwitch
            case reflect.Struct:
                p.buf.WriteByte('&')
                p.printValue(a, verb, plus, goSyntax, depth+1)
                break BigSwitch
        }
    }
}
fallthrough
case reflect.Chan, reflect.Func, reflect.UnsafePointer:
    p.fmtPointer(value, verb, goSyntax)
default:
    p.unknownType(f)

```

25

В 2009 году компания Google создала язык Go (или Golang, сокращение от Google Language). Он должен был заменить C++ и Java в разработке бэкенда веб-приложений и сетевых сервисов. Разработчики сохранили скорость «плюсов» и сделали синтаксис более выразительным и лаконичным: в языке всего 25 ключевых слов.

Основная фишка Golang — многопоточность. На Go можно дёшево (всего за несколько килобайт памяти) и быстро запускать независимые друг от друга функции — горутины — и не опасаться нехватки памяти. На Golang написали Docker и Kubernetes.

Go простой, как Python, но мощный, как C++.

Преимущества языка:

Простота кода — всего 25 ключевых слов;

Низкий риск ошибок;

Автоматическая очистка памяти и документирование кода;

Поддержка Google.

Особенности языка:

Невозможно управлять памятью;

Нет тернарных операторов.

Kotlin

ПРИМЕР KOTLIN КОДА

```

1  /* Block comment */
2  package hello
3  import kotlin.collections.* // line comment
4
5  /**
6   * Doc comment here for 'SomeClass'
7   * @see Iterator#next()
8   */
9  @Deprecated("Deprecated class")
10 private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
11     fun foo(nullable : String?, r : Runnable, f : () -> Int,
12         fl : FunctionLike, dyn: dynamic) {
13         println("length\nis ${nullable?.length} \e")
14         val ints = java.util.ArrayList<Int?>(2)
15         ints[0] = 102 + f() + fl()
16         val myFun = { -> " " };
17         var ref = ints.size
18         ints.lastIndex + globalCounter
19         ints.forEach lit@ {
20             if (it == null) return@lit
21             println(it + ref)
22         }
23     }
24     fun dynamicCall()

```

13.10.2023

26

Kotlin создан на основе Java и не содержит многих библиотек, встраиваемых в разработку Android-приложений. В 2019 Google объявил Kotlin предпочтительным языком для разработки приложений под Android.

Разработала Kotlin питерская компания JetBrains и назвала в его честь острова Kotlin в Финском заливе, на котором расположен город Кронштадт. Kotlin используют в Google, Pinterest, Instagram, TikTok.

Так как старые мобильные приложения написаны на Java, а новые создаются на Kotlin, Android-разработчики должны знать оба языка.

Преимущества языка:

Совместимость с Java;

Компактный язык без кусков избыточного кода.

Особенности языка:

Низкая скорость;

Небольшое сообщество единомышленников.

Swift

ПРИМЕР SWIFT КОДА

```

1 //
2 // Rule.swift
3 // MyGame
4 //
5 // Created by Marcus Rossel on 31.01.16.
6 // Copyright © 2016 Marcus Rossel. All rights reserved.
7 //
8
9 import UIKit
10
11 struct Rule {
12     var name: String
13     var photo: UIImage
14     var description: String
15
16     init?(name: String, photo: UIImage, description: String) {
17         guard !name.isEmpty else { return nil }
18
19         self.name = name
20         self.photo = photo
21         self.description = description
22     }
23 }

```

14.10.2023

27

Swift — это компилируемый язык программирования, который разработала компания Apple для разработки iOS, Mac, Apple TV и Apple Watch.

Преимущества языка:

Высокая скорость работы программ и выполнения кода;

Строгая типизация данных;

Лаконичный синтаксис;

Есть среда Playground, в которой можно сразу же увидеть результат программирования.

Особенности языка:

Язык предназначен только для разработки под Apple.