

ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Курс лекций

О дисциплине

- Цель: формирование первых, основополагающих знаний, умений, навыков и компетенций у студентов в области выбранного профиля подготовки.

Задачи

- *теоретический компонент:*

- получить базовые представления о целях и задачах компонента программирования в профессиональной деятельности бакалавра по направлению ИБ;
- иметь представление о роли дисциплины «языки программирования» в объеме получаемых навыков будущего специалиста;
- изучить основные понятия и разделы программирования;

- *познавательный компонент:*

- формирование необходимого объема знаний о прикладном программировании и вычислительных методах;
- ознакомиться с техническими, алгоритмическими, программными и технологическими решениями, используемыми в профессиональной подготовке будущего специалиста;
- получить базовые навыки в пошаговой детализации разработки структурных алгоритмов и реализации прикладных программ;

- *практический компонент:*

- выработать практические навыки аналитического и экспериментального исследования основных методов и средств, используемых в области, изучаемой в рамках данной дисциплины;
- развить творческую самостоятельность при решении задач в предметной области посредством применения парадигмы программирования.

О дисциплине

- ПК-2 способностью применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач

Знать:

- принципы и этапы компиляции программы;

Уметь:

- разработать программу в определенной среде программирования;

Владеть:

- навыками разработки программного обеспечения на основе структурного подхода.

Трудоемкость дисциплины

4 зачетных единицы (144) часов:

Лекции – **18 ч.**

Лабораторные работы – **16 ч.**

Консультация – **1 ч.**

Экзамен **0,25 ч.**

Самостоятельная работа - **108,75 ч.**

Разделы дисциплины

№	Наименование раздела
1	Введение в программирование. Принципы и этапы компиляции программы
2	Основы алгоритмизации вычислительных процессов
3	Язык программирования C#
4	Структурированные типы данных
5	Функции и структурное программирование
6	Работа с файлами

Лабораторные работы

№	Наименование	часы
1	Знакомство с средой программирование Visual Studio C#	2
2	Операторы выбора	2
3	Организация циклов	2
4	Массивы	2
5	Строки	2
6	Функции	2
7	Работа с файлами	4

Основная и дополнительная литература

1. Троелсен, Э. С# и платформа .NET = C# and the .Net Platform [Текст] / Эндрю Троелсен. - СПб. : Питер, 2007. - 796 с.
2. Кузин А.В. Программирование на языке Си/А.В.Кузин, Е.В.Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с. Режим доступа: <https://znanium.com/catalog/document?pid=505194>
3. Алгоритмы: построение и анализ = Introduction to Algorithms [Текст] / Т. Кормен [и др.]; [пер. с англ. И. В. Красикова, Н. А. Ореховой, В. Н. Романова; под ред. И. В. Красикова].- 2-е изд. - Москва ; Санкт-Петербург ; Киев : Вильямс, 2013. - 1296 с. (ентл 19)
4. 2. Литвиненко В.А. Программирование на С++ задач на графах: Учебное пособие / Литвиненко В.А. - Таганрог: Южный федеральный университет, 2016. - 83 с. Режим доступа: <http://znanium.com/catalog/product/997083>.
5. 3. Хорев П. Б. Объектно-ориентированное программирование с примерами на С#: Учебное пособие [Электронный ресурс] / Хорев П. Б. - НИЦ ИНФРА-М, 2016. Режим доступа: <https://znanium.com/catalog/product/529350>

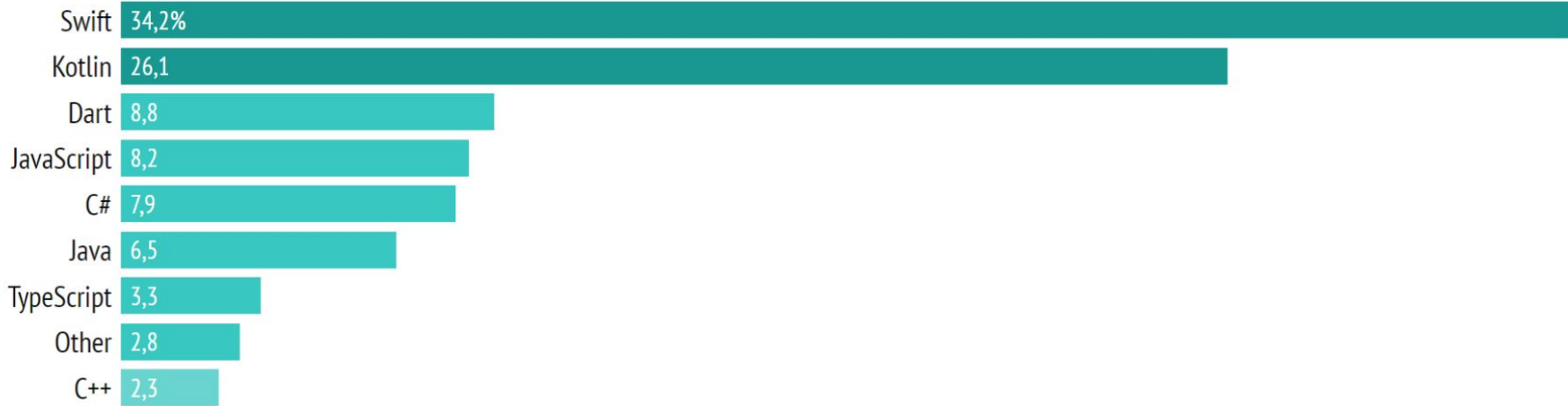
Топ языков программирования по версии IEEE в 2021 году

Рейтинг включает 55 языков. Он составлен на базе 11 параметров, собранных по восьми онлайн-ресурсам — соцсетям, программистским площадкам и сайтам для поиска работы, таким как GitHub, StackOverflow, Google Search, Trends, Twitter, Reddit, Hacker News и CareerBuilder.

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	95.4
3	C	  	94.7
4	C++	  	92.4
5	JavaScript		88.1
6	C#	   	82.4
7	R		81.7
8	Go	 	77.7
9	HTML		75.4
10	Swift	 	70.4

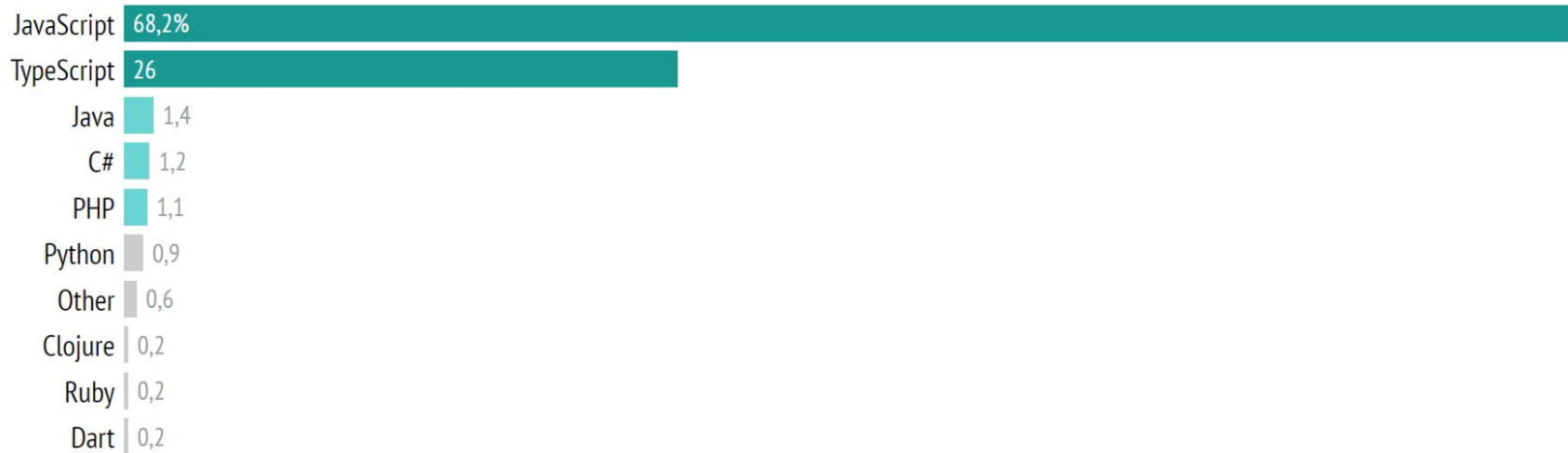
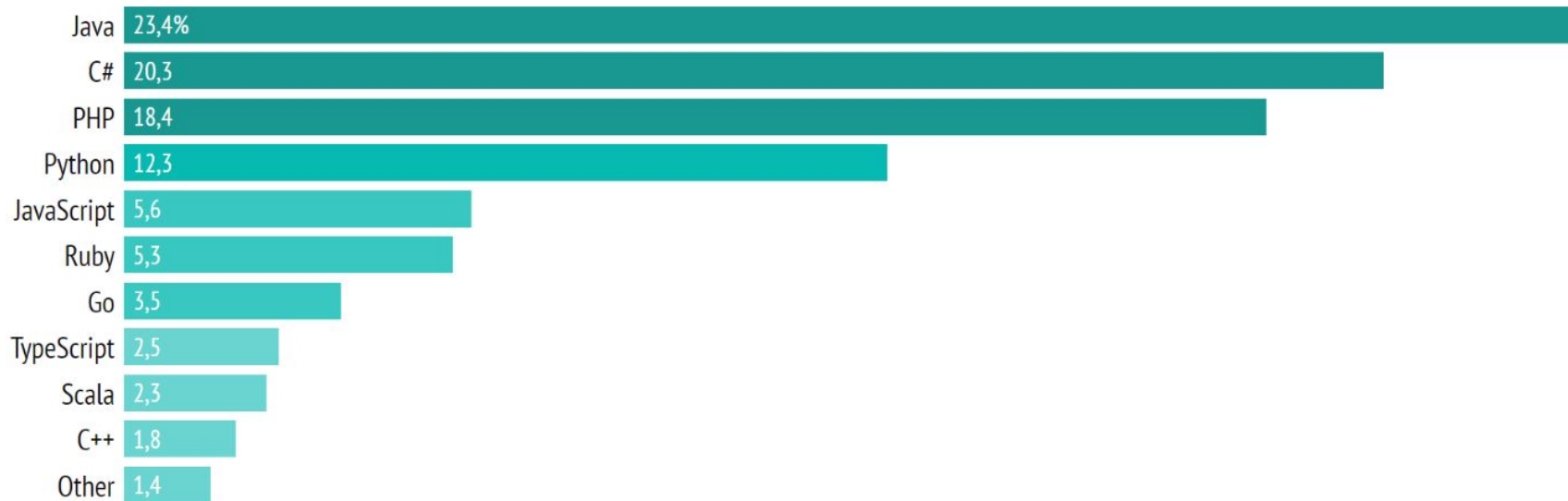
Языки программирования с разбивкой по сферам использования

[Mobile](#) [Back-end](#) [Front-end](#) [Data processing](#) [Desktop](#) [System](#) [QA automation](#) [Full Stack](#) [IoT](#) [GameDev](#)

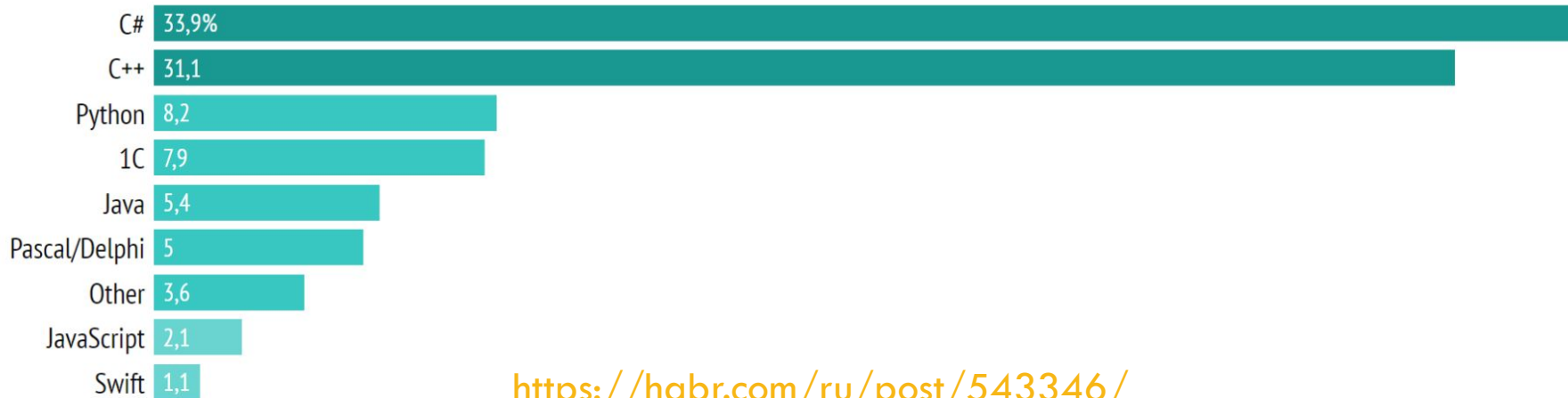


[Mobile](#) [Back-end](#) [Front-end](#) [Data processing](#) [Desktop](#) [System](#) [QA automation](#) [Full Stack](#) [IoT](#) [GameDev](#)





Mobile Back-end Front-end Data processing Desktop System QA automation Full Stack IoT GameDev



<https://habr.com/ru/post/543346/>

Журнал Яндекса Парктикума

<https://thecode.media/best-2021/>

JavaScript

Кому подходит: начинающим программистам и тем, кому нужен быстрый старт.

Средняя зарплата на начало 2021 года: 133 тысячи (в прошлом году — 110 тысяч).

Python

Кому подходит: всем, кто хочет научиться писать универсальные скрипты или работать с нейросетями.

Средняя зарплата на начало 2021 года: 136 тысяч (в прошлом году — 128 тысяч).

Питон в 2021 году пригодится почти везде:

написать скрипт для сервера;

запрограммировать нейросеть;

решить любую классическую задачу в программировании;

Журнал Яндекс Парктикума

<https://thecode.media/best-2021/>

Java

Кому подходит: тем, кто хочет штурмовать корпорации или разрабатывать универсальный софт для разных платформ.

[Как подготовить резюме для крупной компании?](#)

Средняя зарплата на начало 2021 года: 151 тысяча (в прошлом году — 134 тысячи).

Преимущество Java в том, что код можно запустить почти на любом железе, где есть поддержка JVM — виртуальной машины Java:

кнопочные мобильные телефоны;

смартфоны;

компьютеры почти с любым процессором и любой операционной системой;

микроконтроллеры и встраиваемые устройства (например, контроллер для автоматического открывания ворот);

телевизоры, микроволновки и другая умная техника.

Журнал Яндекс Парктикума

<https://thecode.media/best-2021/>

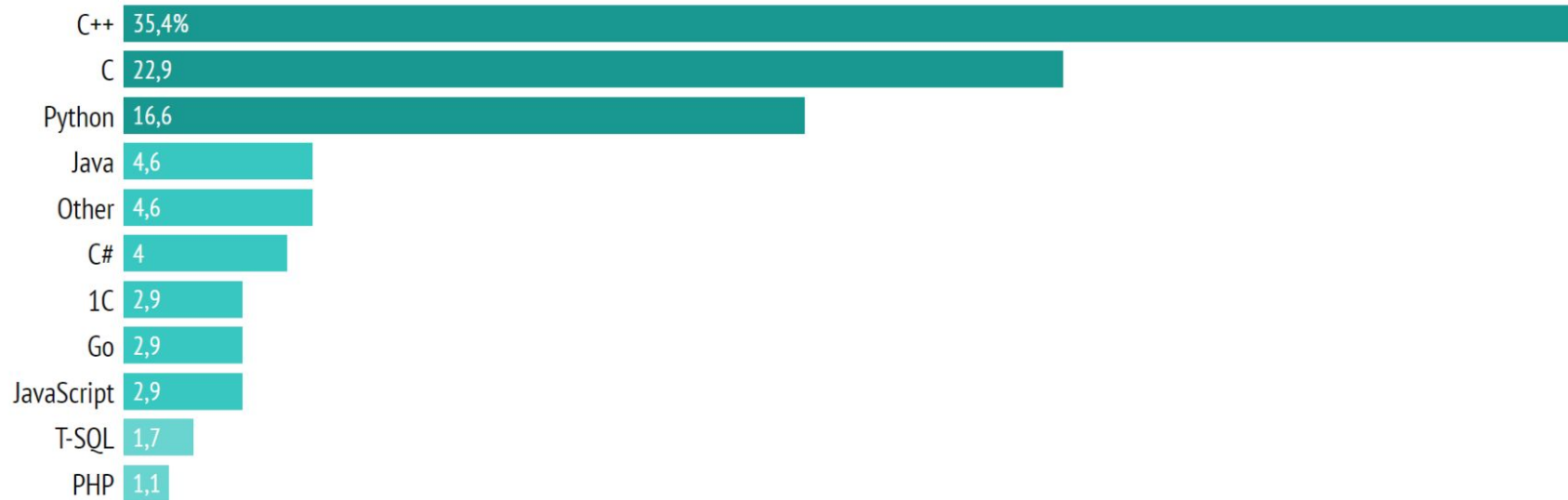
- Rust
- **Кому подходит:** программистам, которым не нравится сложность С или С++, но нравится их мощность и возможности.
- **Средняя зарплата** на конец 2020 года: 207 тысяч (статистических данных на начало 2021 года ещё нет).
- Если вы хотите писать очень быстрые программы, работать с железом напрямую или делать надёжный софт для серверов, то можно начать с Rust. Этот язык позволяет писать в любом стиле, хоть в ООП, как в С++, хоть в функциональном, как Haskell. А при желании можно вообще отказаться от этого и писать в чисто процедурном стиле, как Pascal.

- Swift
- **Кому подходит:** поклонникам операционных систем Apple — MacOS, iOS и WatchOS.
- **Средняя зарплата** на начало 2021 года: 184 тысячи (в прошлом году — 142 тысячи).

А как же С

- Язык С и его младшие братья — это отличный вариант для тех, кто пишет сложный софт под конкретное железо, гонится за быстродействием и управляет беспилотниками силой мысли. Но точно не для начинающих.
- Если Python — это автомобиль с автоматической коробкой передач, то С — это железная руда, из которой вы сможете сделать что угодно: хоть ракету, хоть мотоцикл, хоть автомобиль.

Mobile Back-end Front-end Data processing Desktop System QA automation Full Stack IoT GameDev



<https://habr.com/ru/post/543346/>

Лекция 1 Понятие алгоритма и его свойства

- **Алгоритм** - это определенным образом организованная последовательность действий, за конечное число шагов приводящая к решению задачи.

Разновидности алгоритмов

- *вычислительные* – как правило, работают с простыми видами данных (числа, векторы, матрицы), но зато процесс вычисления может быть длинным и сложным;
- *информационные* – напротив, реализуют сравнительно небольшие процедуры обработки (например, поиск элементов, удовлетворяющих определенному признаку), но для больших объемов информации;
- *управляющие* – непрерывно анализируют информацию, поступающую от тех или иных источников, и выдают результирующие сигналы, управляющие работой тех или иных устройств.

Свойства алгоритма

- *Дискретность*
- *Понятность*
- *Определенность*
- *Результативность*
- *Корректность*
- *Массовость*

Способы описания алгоритма



- Описание алгоритма
 - Словесное
 - Графическое
 - Псевдокод
 - Программная

Словесное описание

- Например, алгоритм нахождения наибольшего общего делителя двух натуральных чисел можно представить как следующую последовательность действий:
 - 1) задание двух чисел;
 - 2) если числа равны, то выбор любого из них в качестве ответа и остановка, в противном случае – продолжение выполнения алгоритма;
 - 3) определение большего из чисел;
 - 4) замена большего из чисел разностью большего и меньшего из чисел;
 - 5) повтор алгоритма с шага 2.

ГОСТ 19.701-90 Схемы алгоритмов



Действие

Начало, завершение программы (подпрограммы)



Действия

Счетные циклы



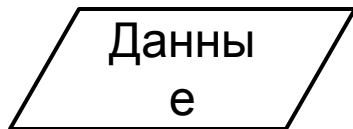
Действие

Обработка данных



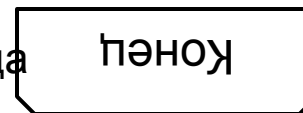
Начало

Любые циклы



Данные

Операции ввода-вывода



Имя

Вызов процедур

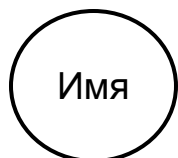


Условия

Ветвления, выбор, итерационные и поисковые циклы

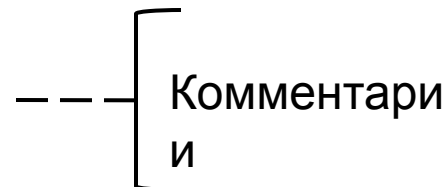


Имя



Имя

Маркировка разрывов линий



Комментарии

Пояснения к операциям

Псевдокоды

Структура	Псевдокод	Структура	Псевдокод
Следование	<p>< Действие 1 > < Действие 2 ></p>	Выбор	<p>Выбор < код > < код 1 > : < Действие 1 > < код 2 > : < Действие 2 > ... Все-выбор</p>
Ветвление	<p>Если < Условие > то < Действие 1 > иначе < Действие 2 > Все-если</p>	Цикл с заданным количеством повторений	<p>Для < индекс > = < n >, < k >, < h >, < Действие > Все-цикл</p>
Цикл-пока	<p>Цикл-пока < Условие > < Действие 1 > Все цикл</p>	Цикл-до	<p>Выполнять < Действие > До < Условие ></p>

Псевдокод реализации поискового цикла

$i = 1$

Цикл-пока $i \leq n$ и $A[i] \neq y$

$i = i + 1$

Все-цикл

Если $i \geq n$

то Вывести «Элемент найден»

иначе Вывести «Элемент не найден»

Все-если

