

Реализовать алгоритм: дано 10 чисел,  
подсчитать итоговое значение по  
правилу

- Если число четное – прибавить к  
итоговому значению
- Если число нечетное – отнять от  
итогового значения
- Иначе – пропустить

10 1 6 7 9 8 5 4 3 9

+10 - 1 + 6 - 7 - 9 + 8 - 5 + 4 - 3 - 9

# **Языки программирования**

Зайнуллин Айдар Фердинатович



**Индекс TIOBE**

# Компилируемые и Интерпретируемые

- **Компилируемые**: C, C++, Pascal
- **Интерпретируемые**: Visual Basic Script (VBScript), JavaScript, Python, PHP
- **Условно компилируемые**: C# и остальные языки .Net, Java для Java-машины

# Универсальные и специализированные

- **Универсальные:** семейство *Pascal/Delphi*, *C/C++*, *C#*, *Java*
- **Специализированные:**
  - Математические вычисления: *Fortran*, *F#*
  - Математическое моделирование: *MatLab*, *Wolfram (Mathematica)*
  - Искусственный интеллект: *LISP*
  - На основе передачи сообщений: *Small Talk*
  - Многопоточные приложения *Cw*
  - Веб-разработка: *Perl*, *PHP*, *JavaScript*
  - Базы данных: *SQL*
  - Компьютерные игры: *Lua*, *Unity*, *Godot*, *Twine*
  - Компьютерная графика: *MEL (Maya)*, *MAX Script (3ds Max)*
  - Бухгалтерия: *1C*

# Алгоритмические и Языки описания данных

- *Алгоритмические*: *Pascal, C++, Java, C# и т.д.*
- *Языки описания данных*: *XML, XAML, JSON, HTML, DDL SQL*

# Низкоуровневые и Высокоуровневые

- *Низкоуровневые*: *Assembler, CIL*
- *Высокоуровневые*: *любой объектно-ориентированный или поддерживающий сложные типы данных язык*

# **Объектно-Ориентированные и Структурные языки программирования**



# Assembler

В начале 1950-х годов была осуществлена идея использования символьных имен вместо адресов данных и замены цифровых кодов операций на мнемонические (словесные) обозначения. Язык программирования, реализующий данный подход, получил название Ассемблер (от англ. Assembler – сборщик) .

Программа, записанная на Ассемблере, не может обрабатываться непосредственно процессором. Возникла необходимость преобразования текста программы, записанной на данном языке, в машинный код. Для решения этой задачи были созданы *трансляторы*.

*Транслятор* – это программа, преобразующая текст, представленный на одном из языков программирования, в эквивалентную программу на машинном языке.

# Fortran

**Fortran** (Formula Translation – трансляция формул) – первый язык программирования высокого уровня, имеющий транслятор.

Создан в период с 1954 по 1957 год группой программистов под руководством Джона Бэкуса в корпорации IBM.

Фортран широко используется в первую очередь для научных и инженерных вычислений. Одно из преимуществ современного Фортрана – большое количество написанных на нём программ и библиотек подпрограмм. Среди учёных, например, ходит такая присказка, что любая математическая задача уже имеет решение на Фортране, и, действительно, можно найти среди тысяч фортрановских пакетов и пакет для перемножения матриц, и пакет для решения сложных интегральных уравнений, и многие, многие другие

# Basic

**Basic** (от BASIC, сокращение от англ. Beginner's All-purpose Symbolic Instruction Code – универсальный код символических инструкций для начинающих) – семейство высокоуровневых языков программирования.

Был разработан в 1963 году профессорами Дартмутского колледжа Томасом Курцем и Джоном Кемени.

Язык предназначался для обучения программированию и получил широкое распространение в виде различных диалектов, прежде всего как язык для домашних компьютеров.

# Pascal

**Pascal** – язык назван в честь выдающегося французского математика, физика, литератора и философа Блеза Паскаля, который создал первую в мире механическую машину, складывающую два числа.

Язык Паскаль был создан Никлаусом Виртом в 1968–1969 годах после его участия в работе комитета разработки стандарта языка Алгол–68. Он был опубликован в 1970 году Виртом как небольшой и эффективный язык, чтобы способствовать хорошему стилю программирования, использовать структурное программирование и структурированные данные.

# С

**Си** (англ. **C**) — стандартизированный процедурный язык программирования, разработанный в начале 1970-х годов сотрудниками Bell Labs Кеном Томпсоном и Деннисом Ритчи как развитие языка Би. Си был создан для использования в операционной системе UNIX. С тех пор он был перенесён на многие другие операционные системы и стал одним из самых используемых языков программирования.

Си ценят за его эффективность. Он является самым популярным языком для создания системного программного обеспечения. Его также часто используют для создания прикладных программ. Несмотря на то, что Си не разрабатывался для новичков, он активно используется для обучения программированию. В дальнейшем синтаксис языка Си стал основой для многих других языков.

# Java

**Java** — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно компилируются в специальный байт-код, поэтому они могут работать на любой виртуальной Java-машине (JVM) вне зависимости от компьютерной архитектуры.

Дата официального выпуска — 23 мая 1995 года.

# C++

**C++** (автор Бьёрн Страуструп) – компилируемый статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования как процедурное программирование, модульность, отдельная компиляция, обработка исключений, абстракция данных, типы (объекты), виртуальные функции, объектно-ориентированное программирование, обобщенное программирование, контейнеры и алгоритмы, сочетает свойства как высокоуровневых, так и низкоуровневых языков.

# Классификация языков программирования

[https://cdn3.mindmeister.com/ru/129081511/\\_?fullscreen=1](https://cdn3.mindmeister.com/ru/129081511/_?fullscreen=1)



# Типизация языков программирования

Языки программирования бывают **типизированными** и **нетипизированными** (бестиповыми)

# Нетипизированный язык программирования

Все данные в таких языках считаются цепочками бит произвольной длины и не делятся на типы

# Статическая и динамическая типизация

```
int data;  
data = 20;  
data = "Hello";
```

**ERROR**

# Статическая и динамическая типизация

```
data = 8  
data = "Hello"
```

# Сильная и слабая типизация

```
5 + '6'; // '56'  
4 * '3'; // 12  
2 + true; // 3  
false - 4; // -4
```

# Сильная и слабая типизация

```
foo = "x"  
foo = foo + 5  
Traceback (most recent call last):  
File "<pyshell#3>", line 1, in ?  
foo = foo + 5  
TypeError: cannot concatenate 'str' and 'int' objects
```

# ERROR

# Явная и неявная типизация

```
int a = 7
```

явная

```
a = 7
```

неявная