Современные языки программирования высокого уровня.

Для демонстрации отличий в синтаксисе языков программирования была выбрана простая олимпиадная задача 2016 года:

Всероссийская олимпиада школьников по информатике, муниципальный этап
Санкт-Петербург, 12 декабря 2016 года

Задача A. Любитель нулей

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Саша очень любит нули. Но нули на конце числа не кажутся ему интересными. Разумеется, ведущие нули тоже не интересуют Сашу.

Саша считает красоту числа равной количество нулей в его десятичной записи, за исключением нулей в конце числа. Разумеется, ведущих нулей в записи быть не должно. Например, красота числа 100500 равна 2.

По заданному числу k выясните, чему равна его красота по мнению Саши.

Формат входных данных

Входные данные содержат одно число k ($1 \leqslant k \leqslant 10^9$).

Формат выходных данных

Выведите одно число — красоту числа k по мнению Саши.

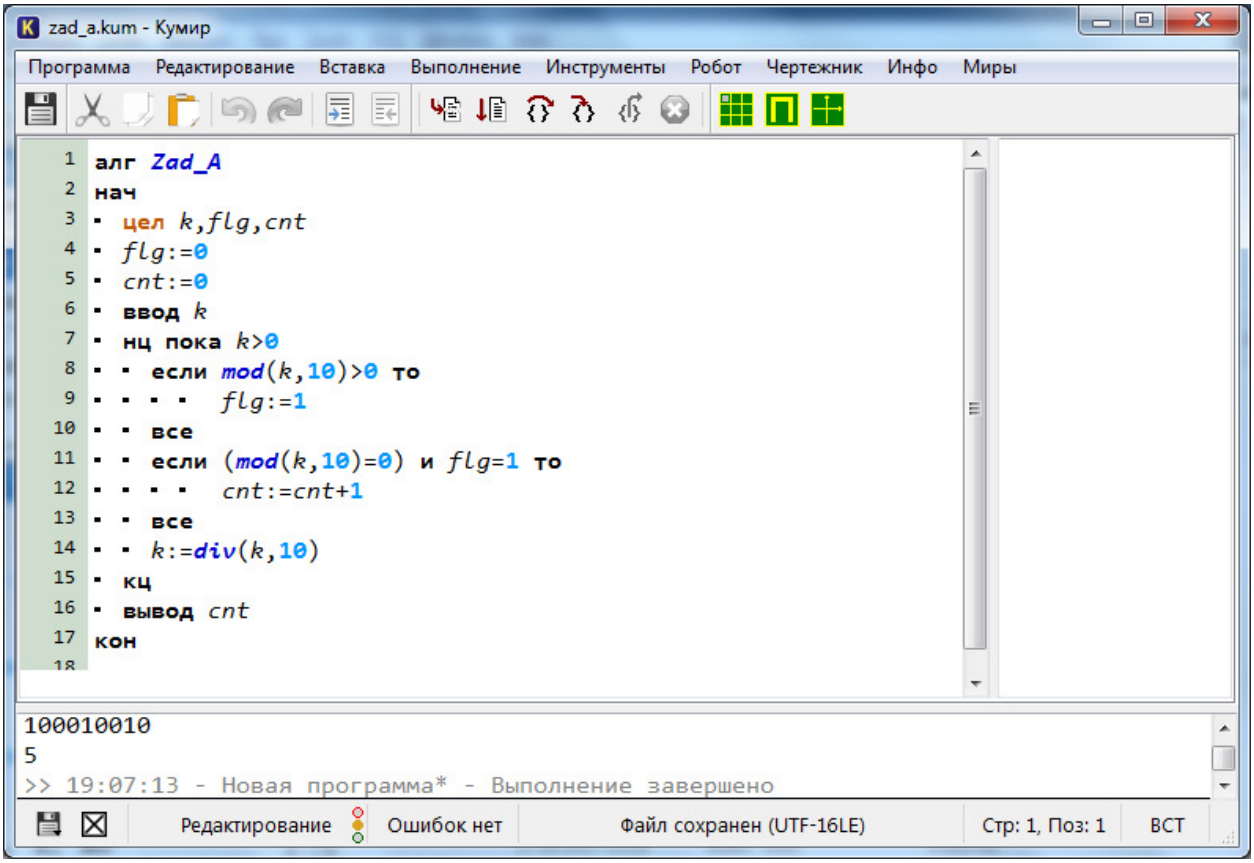
Пример

стандартный ввод	стандартный вывод
100500	2

Задача достаточно тривиальна и ниже приведены тексты программ и скриншоты сред программирования для различных алгоритмических языков, которые на наш взгляд заслуживают внимания при изучении информатики.

КУМИР

КУМИР (Комплект Учебных МИРов) — язык и система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе. Основана на методике, разработанной во второй половине 1980-х годов в СССР, под руководством академика А. П. Ершова. В системе КуМир используется придуманный А. П. Ершовым школьный алгоритмический язык — простой алгол-паскаль-подобный язык с русской лексикой и встроенными командами управления программными исполнителями (Робот, Чертёжник).



Текст программы:

```
алг Zad_A
нач
. цел k,flg,cnt
. flg:=0
. cnt:=0
. ввод k
. нц пока k>0
. . если mod(k,10)>0 то
. . . . flg:=1
. . все
. . если (mod(k,10)=0) и flg=1 то
. . . . cnt:=cnt+1
. . все
. . k:=div(k,10)
. кц
. вывод cnt
кон
```

FORTRAN

FORTRAN – это первый "настоящий" язык программирования высокого уровня. До его появления, все программы писались на "машинных" языках, учитывающих архитектурные особенности центральных процессоров ЭВМ различных производителей (фирм). Название Fortran является аббревиатурой от FORmula TRANslator, то есть, переводчик формул. Фортран широко используется в первую очередь для научных и инженерных вычислений. Одно из

преимуществ современного Фортрана — огромное количество написанных на нём программ и библиотек подпрограмм.
Современный Фортран (Fortran 95 и Fortran 2003) приобрёл черты, необходимые для эффективного программирования для новых вычислительных архитектур и позволяет применять современные технологии программирования, в частности, объектно-ориентированного программирования (ООП).
Фортран — жёстко стандартизированный язык, именно поэтому он легко переносился на различные платформы.
Существует несколько международных стандартов языка:

- FORTRAN IV (он же — FORTRAN 66) (1966)
- FORTRAN 77 (1978)

Множество улучшений: строковый тип данных и функции для его обработки, блочные операторы IF, ELSE IF, ELSE, END IF, оператор включения фрагмента программы INCLUDE и т. д.

- Fortran 90 (1991)

Значительно переработан стандарт языка. Введён свободный формат написания кода. Появились дополнтельные описания IMPLICIT NONE, TYPE, ALLOCATABLE, POINTER, TARGET, NAMELIST; управляющие конструкции DO … END DO, DO WHILE, CYCLE, SELECT CASE, WHERE; работа с динамической памятью (ALLOCATE, DEALLOCATE, NULLIFY); программные компоненты MODULE, PRIVATE, PUBLIC, CONTAINS, INTERFACE, USE, INTENT. Появились новые встроенные функции, в первую очередь, для работы с массивами.
В языке появились элементы ООП.
Отдельно объявлен список устаревших черт языка, предназначенных для удаления в будущем.

- Fortran 95 (1997)

Коррекция предыдущего стандарта.

- Fortran 2003 (2004)

Дальнейшее развитие поддержки ООП в языке. Взаимодействие с операционной системой.
До 1997 основным производителем компиляторов Фортрана для IBM PC совместимых компьютеров была корпорация "Майкрософт". Впоследствии она отказалась от их разработки в связи с низкой прибыльностью. Также коммерческие компиляторы поставляла фирма "Digital Equipment Corporation" (DEC),1958-1998 г.г. DEC была куплена компанией "Compaq" и вместе с последней в 2002 г. слилась с "HP". Компания разработала компилятор, интегрированный в среду разработки Digital Visual Fortran, основанную на Microsoft Visual Studio. Известными продуктами этой линейки являлись FPS 4.0 (Microsoft Fortran Power Station), DVF 5.0 и 6.0. Каждый компилятор мог поддерживать несколько стандартов

Фортрана. Поглощения (слияния) компаний DEC и Compaq явились причиной того, что последующие продукты появлялись на рынке под торговыми марками Compaq и HP. HP разработала среду разработки для Intel процессоров. Поддержка Фортрана реализована также для всех высокопроизводительных серверных платформ HP. Другим крупным поставщиком систем разработки на Фортране являлась фирма "Lahey", предлагавшая интегрированные решения для Windows и Linux. Долгое время лучшим компилятором Фортрана считался компилятор фирмы "Watcom", который был выделен в отдельный проект Opan Watcom и развивавший компилятор на открытой основе. Широко известен и активно развивается также компилятор фирмы Intel - Intel Fortran Compiler, который позволяет оптимизировать код под платформу Intel ia32 и ia64. Фонд свободного программного обеспечения GNU выпускает открытый компилятор Фортрана-77 g77, доступный практически для любой платформы и полностью совместимый с GCC, но не поддерживающий всех языковых конструкций современных стандартов Фортрана. Также существует проект g95 по созданию на основе GCC компилятора Fortran-95.

Структура программ изначально была ориентирована на ввод с перфокарт и имела ряд удобных именно для этого случая свойств. Так, 1-я колонка служила для маркировки текста как комментария (символом C), с 1-й по 5-ю располагалась область меток, а с 7-й по 72-ю располагался собственно текст оператора или комментария. Колонки с 73-й по 80-ю могли служить для нумерации карт (чтобы восстановить случайно рассыпавшуюся колоду) или для краткого комментария, транслятором они игнорировались. Если текст оператора не вписывался в отведённое пространство (с 7-й по 72-ю колонку), в 6-ой колонке следующей карты ставился признак продолжения, и затем оператор продолжался на ней. Расположить два или более оператора в одной строке (карте) было нельзя. Когда перфокарты ушли в историю, эти достоинства превратились в серьёзные неудобства. Именно поэтому в стандарт Фортрана, начиная с Fortran 90, в добавление к фиксированному формату исходного текста появился свободный формат, который не регламентирует позиции строки, а также позволяет записывать более одного оператора на строку. Введение свободного формата позволило создавать код, читабельность и ясность которого не уступает коду, созданному при помощи других современных языков программирования, таких как C или Java.

Своего рода "визитной карточкой" старого Фортрана является огромное количество меток, которые

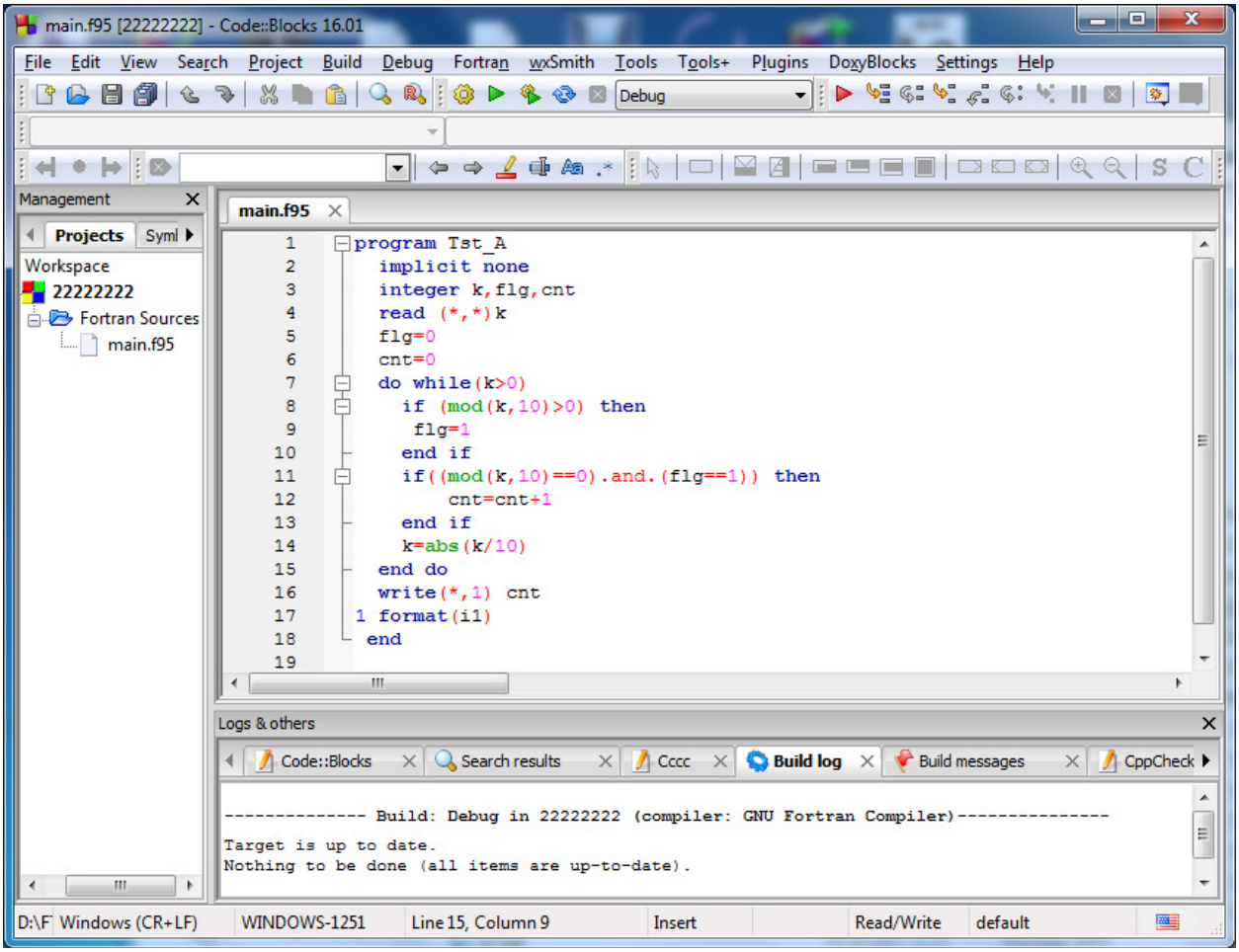
использовались как в операторах безусловного перехода GOTO , так и в операторах циклов, и в операторах описания форматного ввода/вывода **FORMAT.** Большое количества меток и операторов **GOTO** часто делало программы на Фортране трудными для понимания.

Именно этот негативный опыт стал причиной, по которой в ряде современных языков программирования (например, **Java**) метки и связанные с ними операторы безусловного перехода (**go to label**) вообще отсутствуют.

Однако современный Фортран избавлен от избытка меток за счет введения таких операторов, как **DO … END DO, DO WHILE, SELECT CASE**

Также к положительным чертам современного Фортрана стоит отнести большое количество встроенных операций с массивами и гибкую поддержку массивов с необычной индексацией.

```
!Программа на Fortran-95
program Tst_A
  implicit none
  integer k,flg,cnt
  read (*,*)k
  flg=0
  cnt=0
  do while(k>0)
    if (mod(k,10)>0) then
     flg=1
    end if
    if((mod(k,10)==0).and.(flg==1)) then
        cnt=cnt+1
    end if
    k=abs(k/10)
  end do
  write(*,1) cnt
1 format(i1)
 end
```

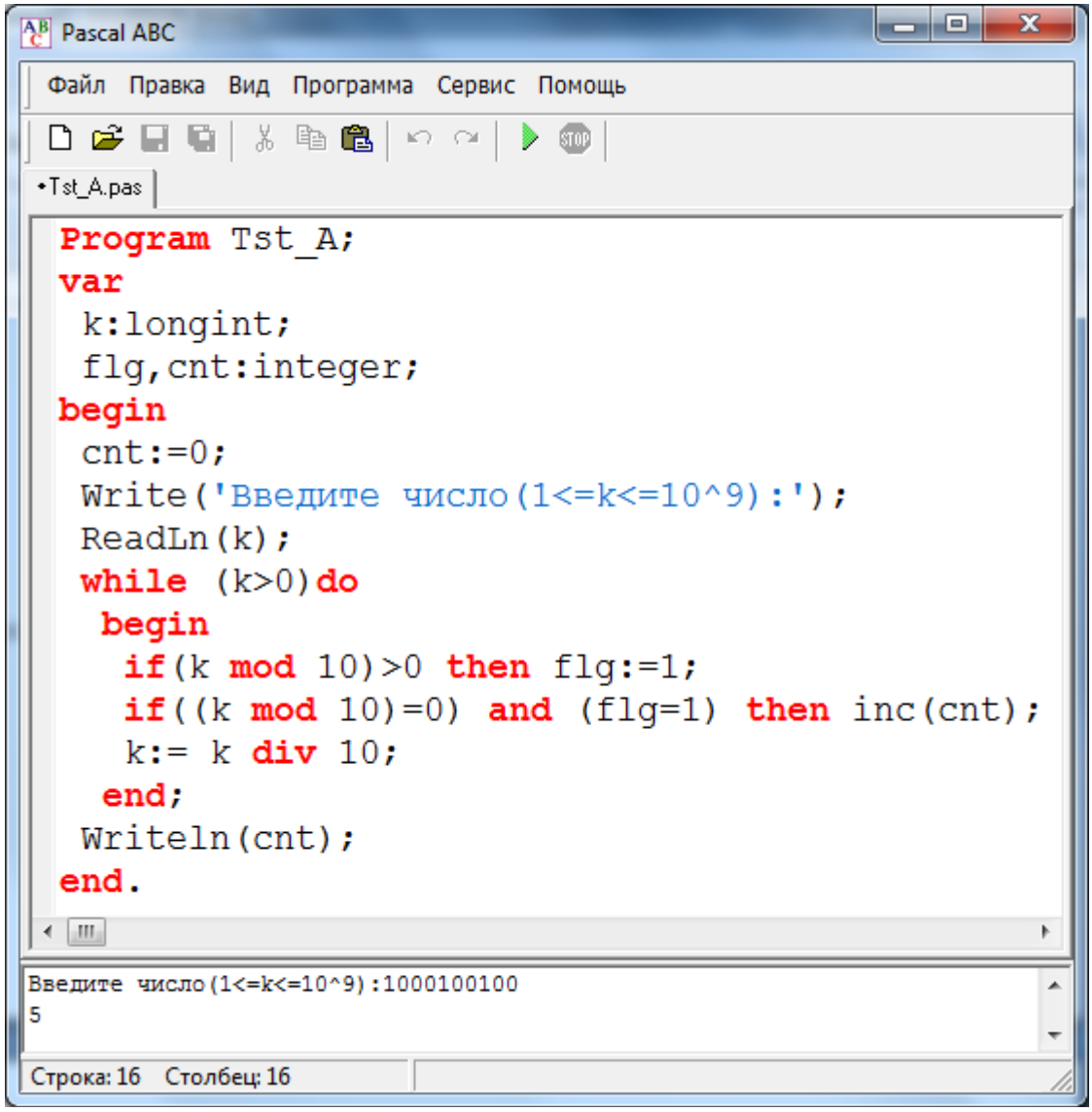


Pascal

Язык Паскаль (Pascal) был создан профессором Никлаусом Виртом в 1968—1969 годах после его участия в работе комитета по разработке стандартов других алгоритмических языков (Алгол-68). Язык назван в честь выдающегося французского математика, физика, литератора и философа Блеза Паскаля, который создал первую в мире суммирующую механическую машину (Паскалина). Изобретённый Паскалем принцип связанных зубчатых колёс почти на три столетия стал основой создания всех механических арифмометров. Первая публикация Вирта о языке датирована 1970 годом; представляя язык, автор в качестве цели его создания указывал построение небольшого и эффективного языка, способствующего хорошему стилю программирования, использующему структурное программирование и структурированные данные. Реализации систем программирования на языке Pascal: **Pascal ABC**, **Free Pascal**, **Embarcadero Delphi** и др.

```
Program Tst_A;
var
 k:longint;
 flg,cnt:integer;
begin
```

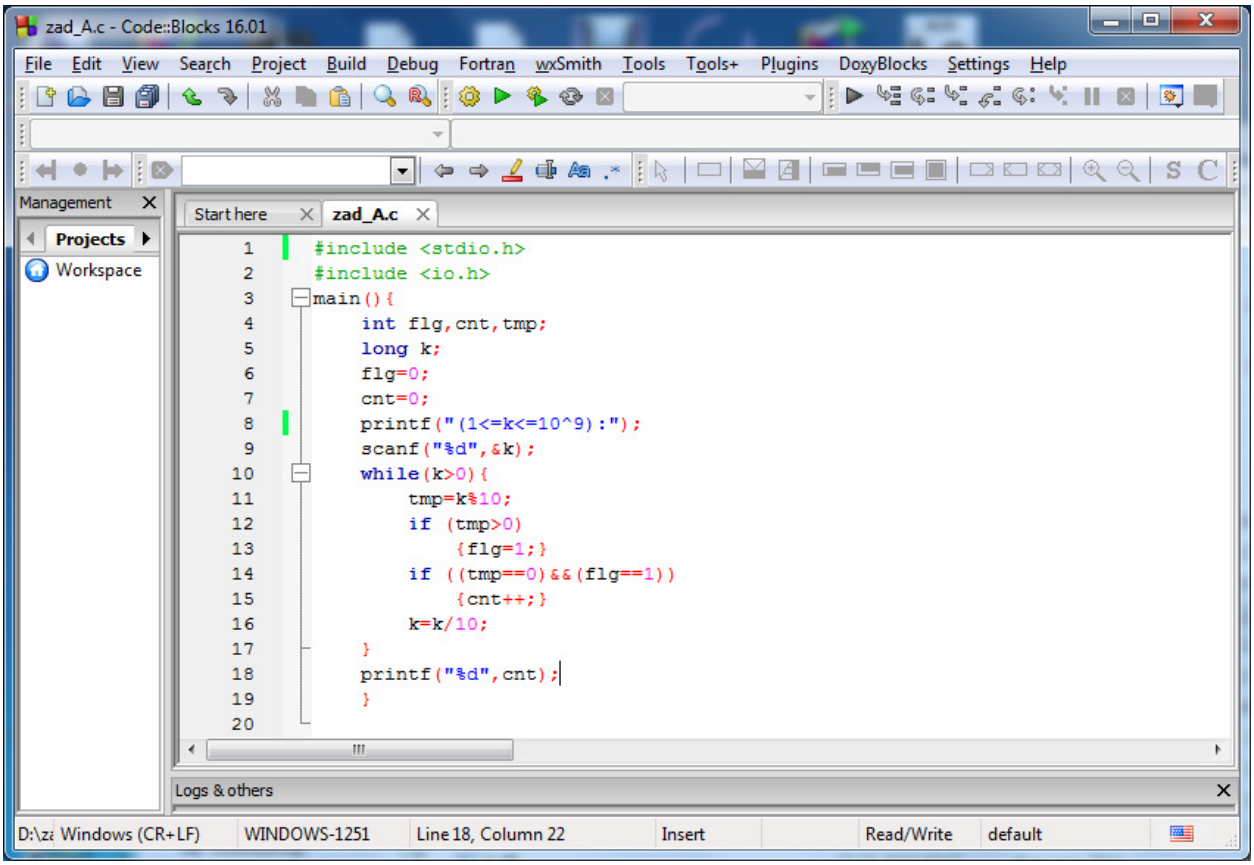
```
 cnt:=0;
 Write('Введите число(1<=k<=10^9):');
 ReadLn(k);
 while (k>0)do
  begin
   if(k mod 10)>0 then flg:=1;
   if((k mod 10)=0) and (flg=1) then inc(cnt);
   k:= k div 10;
  end;
 Writeln(cnt);
end.
```



C

C (Си) — компилируемый статически типизированный язык программирования общего назначения, разработанный в 1969—1973 годах сотрудниками Bell Labs Деннисом Ритчи и Кеном Томпсоном. Изобретение языка Си и его роль в разработке операционной системы UNIX (Д.Ритчи и К.Томпсон), сделали его пионером современной вычислительной техники. Язык Си по сей день широко используется для написания приложений и операционных

систем, и его влияние наблюдается во многих современных языках программирования. ОС UNIX также оказал колоссальное влияние, основав идеи и принципы, которые сейчас являются прочно устоявшимися в вычислительной технике.
Первоначально Си был разработан для реализации операционной системы UNIX, но впоследствии был перенесён на множество других платформ. Согласно дизайну языка Си, его конструкции близко сопоставляются типичным машинным инструкциям, благодаря чему он нашёл применение в проектах, для которых был свойственен язык ассемблера, в том числе как в операционных системах, так и в различном прикладном ПО для множества устройств — от суперкомпьютеров до встраиваемых систем. Язык программирования Си оказал существенное влияние на развитие индустрии программного обеспечения, а его синтаксис стал основой для таких языков программирования, как **C++**, **C#**, **Java** и **Objective-C (Apple)**.




```
#include <stdio.h>
#include <io.h>
main(){
    int flg,cnt,tmp;
    long k;
    flg=0;
```

```
cnt=0;
printf("(1<=k<=10^9):");
scanf("%d",&k);
while(k>0){
    tmp=k%10;
    if (tmp>0)
        {flg=1;}
    if ((tmp==0)&&(flg==1))
        {cnt++;}
    k=k/10;
}
printf("%d",cnt);
}
```

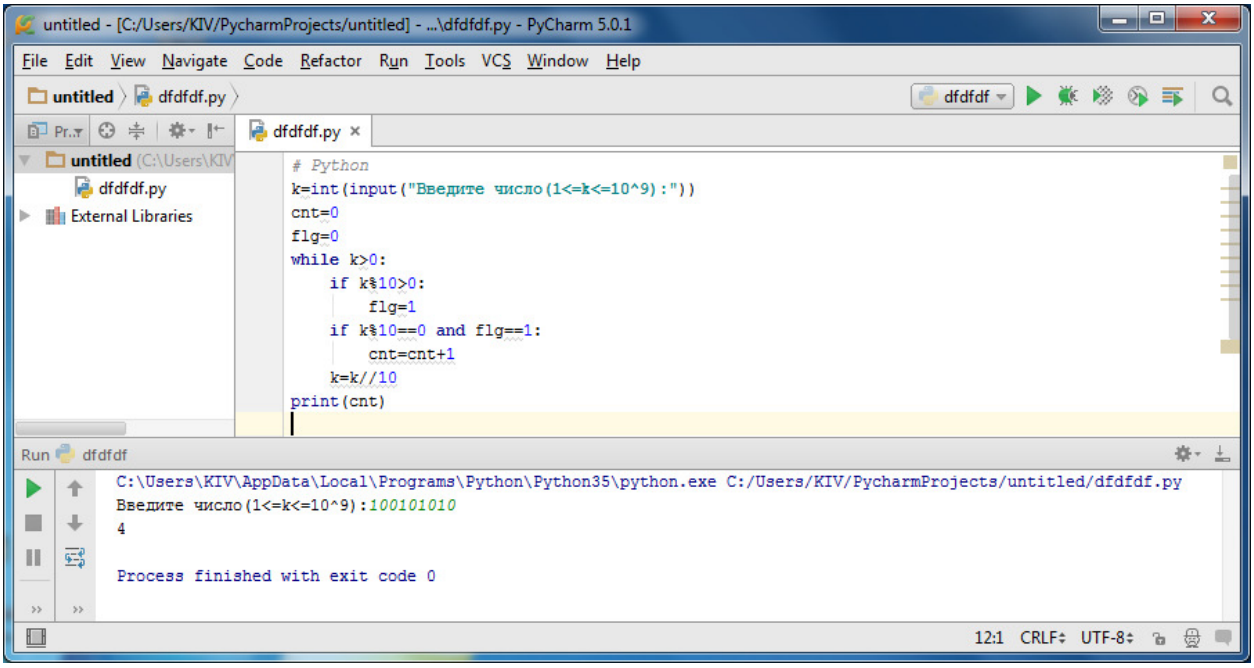
Python

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Python организовывается в функции и классы, которые могут объединяться в модули (они в свою очередь могут быть объединены в пакеты).

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая коммерческие. Есть реализации множества интерпретаторов для JVM (с возможностью компиляции) и других.

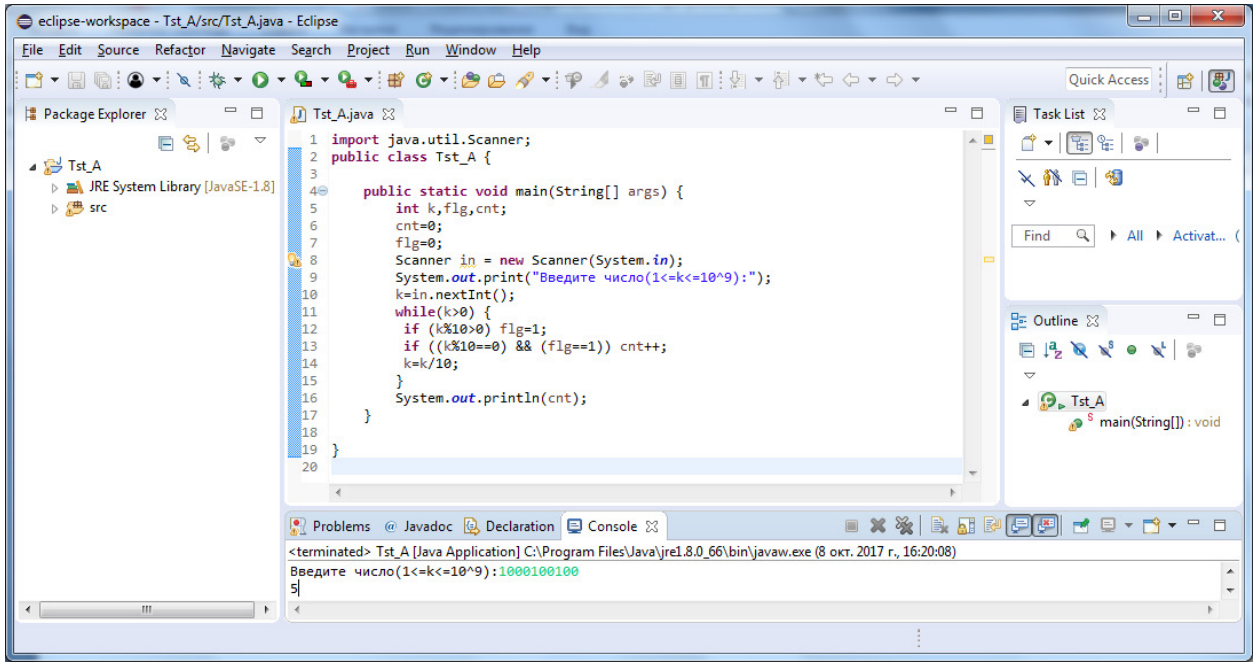
Python — активно развивающийся язык программирования, новые версии (с добавлением/изменением языковых свойств) выходят примерно раз в два с половиной года. Вследствие этого и некоторых других причин на Python отсутствуют стандарт ANSI, ISO или другие официальные стандарты.




```
# Python
k=int(input("Введите число(1<=k<=10^9):"))
cnt=0
flg=0
while k>0:
    if k%10>0:
        flg=1
    if k%10==0 and flg==1:
        cnt=cnt+1
    k=k//10
print(cnt)
```

Java

Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

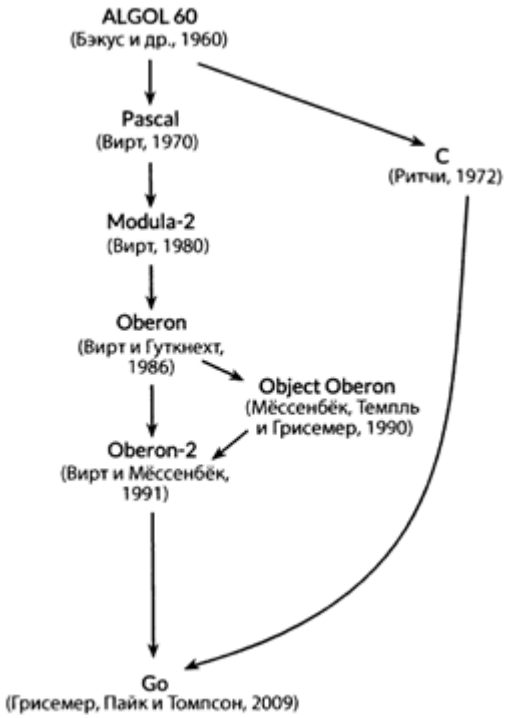



```
import java.util.Scanner;
public class Tst_A {

    public static void main(String[] args) {
        int k,flg,cnt;
        cnt=0;
        flg=0;
        Scanner in = new Scanner(System.in);
        System.out.print("Введите число(1<=k<=10^9):");
        k=in.nextInt();
        while(k>0) {
         if (k%10>0) flg=1;
         if ((k%10==0) && (flg==1)) cnt++;
         k=k/10;
        }
        System.out.println(cnt);
    }

}
```

Go

Подобно биологическим видам, успешные языки порождают потомство, которое наследует наилучшие особенности своих предков. Скрещивание при этом иногда приводит к удивительным результатам. Аналогом мутаций служит появление радикально новых идей. Как и в случае с живыми существами, глядя на такое влияние предков, можно многое сказать о том, почему язык получился именно таким, какой он есть, и для каких условий работы он приспособлен более всего.

На рисунке ниже показано, какие языки повлияли на дизайн языка программирования Go.

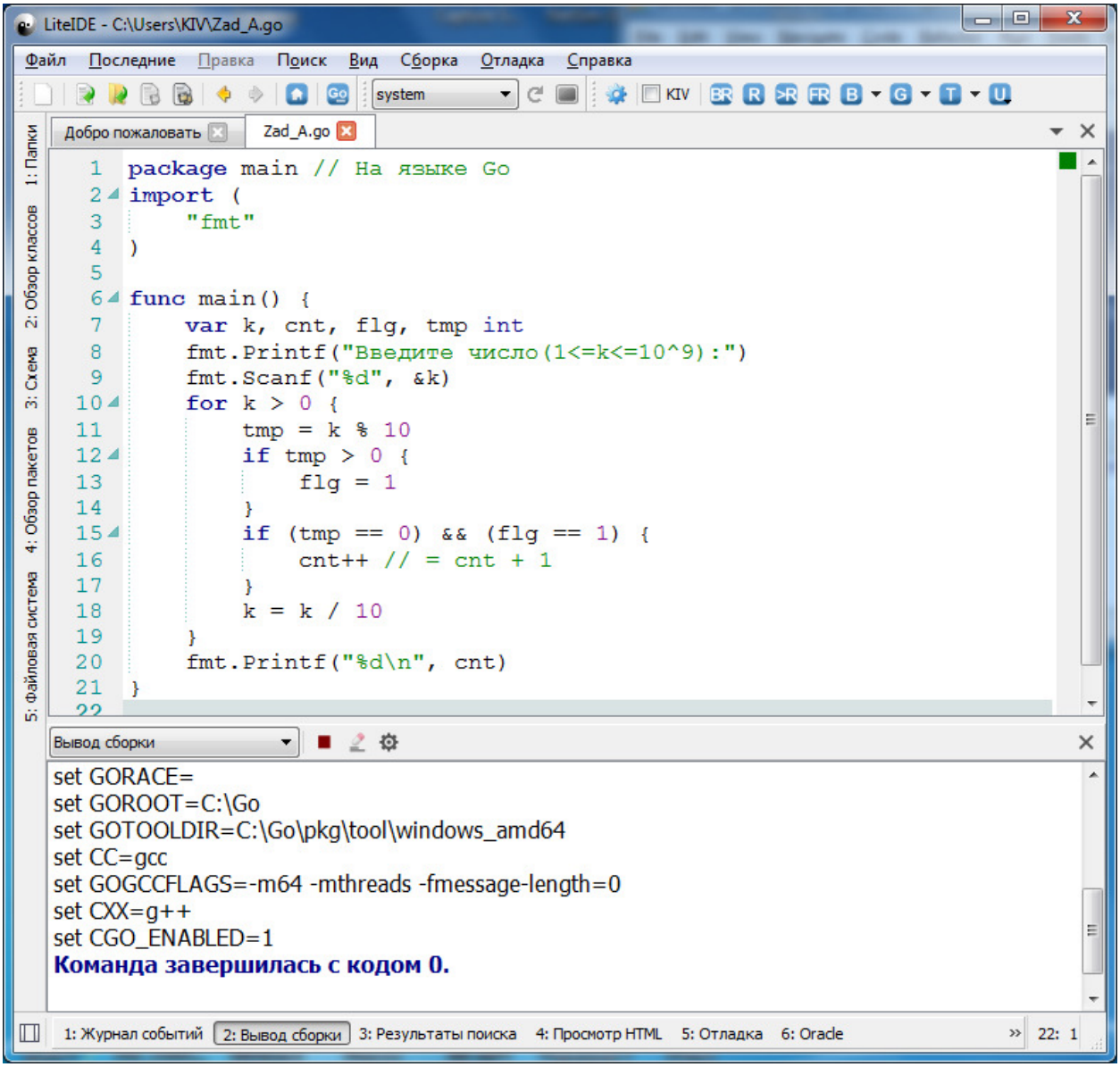


Язык Go часто описывают как "C-подобный язык" или "язык C XXI века". От языка C, Go унаследовал синтаксис выражений, конструкции управления потоком, базовые типы данных, передачу параметров в функции по значению, понятие указателей и, что важнее всего, направленность C на получение при компиляции эффективного машинного кода и естественное взаимодействие с абстракциями современных операционных систем.

Однако в генеалогическом древе Go есть и другие предки. Одно из сильнейших влияний на Go оказали языки программирования профессора Вирта, начиная с Pascal. Modula-2 привнесла концепцию пакетов. Oberon использует один файл для определения модуля и его реализации. Oberon-2 явился источником синтаксиса пакетов, импорта и объявлений (прежде всего, объявлений методов), которые он, в свою очередь, унаследовал от языка Object Oberon.

Все языки программирования тем или иным образом отражают философию их создателей, что часто приводит к включению в язык программирования их реакции на слабости и недостатки более ранних языков. Go не является исключением. Проект Go родился из разочарования специалистов корпорации Google несколькими программными системами, страдающими от "взрыва сложности".

Только с помощью простоты дизайна система может в процессе роста оставаться устойчивой, безопасной и последовательной. Проект Go включает в себя сам язык, его инструментарий, стандартные библиотеки и последнее (по списку, но не по значению) - культуру радикальной простоты. Будучи одним из самых современных высокоуровневых языков, Go обладает преимуществом ретроспективного анализа других языков, и это преимущество использовано в полной мере.




```go
// На языке Go алгоритм решения
package main
import (
    "fmt"
)

func main() {
    var k, cnt, flg, tmp int
    fmt.Printf("Введите число(1<=k<=10^9):")
    fmt.Scanf("%d", &k)
    for k > 0 {
        tmp = k % 10
        if tmp > 0 {
            flg = 1
        }
        if (tmp == 0) && (flg == 1) {
            cnt++ // = cnt + 1
        }
        k = k / 10
    }
    fmt.Printf("%d\n", cnt)
}
```

Генеалогическое дерево рассмотренных выше языков программирования