

# Какво са бит, байт, килобайт и т.н.?

- Бит(*binary digit*) - 1 двоична цифра  
1 0 0 0 1<sub>(2)</sub> – число с 5 бита
- Байт – 8 бита  
1 0 0 1 1 0 0 1<sub>(2)</sub> - число с 8 бита или 1 байт
- Килобайт – 1000/1024 байта (спори се)
  - International System of Units – 1kb = 10<sup>3</sup> bytes
  - Microsoft – 1kb = 2<sup>10</sup> bytes

# Типове данни

- Усложняват работата на начинаещите програмисти
- Позволяват управление на паметта
- Помагат за проверка на логика
- Определят множество от допустими стойности
- Делят се на примитивни и съставни

# Примитивни типове данни

- булев (bool)
- целочислен (int)
- символен (char)
- изборен (enum) – в бонус материалите
- числа с плаваща запетая (float, double)

# Размер на примитивните типове данни

\*Информацията е валидна за 64 битова ОС, x64 базиран процесор и VC компилатор

- 1 byte – bool, char
- 2 bytes – short (int)
- 4 bytes – int, long(int), float, enum
- 8 bytes – long long(int), double, long double

# Boolean

- Ключова дума `bool`
- Множество от стойности:  $\{0, 1\}$
- 0 и 1 могат да бъдат заместени с литералите `false` и `true`
- `false` е равностойно на 0
- `true` е равностойно на 1
- Всяка стойност различна от 0 се смята за `true`

# Цели числа

- Ключова дума `int` (или само името на модификатора)
- Множество от стойности:  $[-2^{31}; 2^{31} - 1]$
- Модификатори
  - `short`:  $[-2^{15}; 2^{15} - 1]$
  - `long`:  $[-2^{31}; 2^{31} - 1]$
  - `long long`:  $[-2^{63}; 2^{63} - 1]$
  - `unsigned`:  $[0; 2^x - 1]$ , където ( $x = 16, 32, 64$ )
- По подразбиране целочислените константи са `int`, за да се третира като `long long` трябва да го отбележим:
  - `123414234123523L`

# Числа с плаваща запетая

- Ключова дума `double`
- Множество от стойности:  $\pm 1.7e \pm 308$  (~15 digits)
- Модификатори
  - `long`: също като при `int` нищо не прави под моята архитектура

# Числа с плаваща запетая

- Ключова дума float
- Множество от стойности: +/- 3.4e +/- 38 (~7 digits)
- По подразбиране константите с плаваща запетая са double, за да се отбележи, че искаме да се третира като float трябва да го отбележим:
  - 3.14F

# Character

- Ключова дума `char`
- `' '` – използват се, за да работим директно със символ
- `' '` съдържат само 1 символ
- `\ '`  – специален символ, няколко символа образуват 1 конкретен
  - `'\n', '\t', '\\'`
- Приложения на ASCII таблицата може да разгледате в бонус материалите

# ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

# Константи и литерали

- Константи и литералните стойности в C++ са такива, които не могат да се променят
- Числовите стойности
  - 1 е 1 и това няма как да се промени
- Booleans
  - Истината е истина и лъжата е лъжа
- Символи
  - 'а' не може да се промени

# Променливи

- Ядрото на информатиката
- Обратно на константите, потребителят избира каква стойността на променливата
- Потребителят избира типът
- Потребителят може да променя стойността

# СИНТАКСИС(не важи за enum)

- <тип> <име на променливата>; - декларация
- <тип><име на променливата> = <стойност>; - декларация + инициализация
- **!!!Неинициализираните променливи са потенциален източник на грешки, бъдете много внимателни с тях!!!**

# Именуване на променливи

- Препоръчително от 1 до 31 символа, някои компилатори работят и с по-дълги имена
- Имената трябва да започват с латинска буква, \$ или \_, след първия символ може да има и цифри
- Има разлика между главни и малки букви
- Някои имена са забранени, защото вече са запазени
- След като едно име бъде употребено то също става запазено за съответното поле

# Примери

```
int $php = 5;
```

```
double _$_ = 3;
```

```
char nayqkiqsimvolever = 'M';
```

```
float MnOgOqKo = 2.4;
```

```
bool _ = false;
```

```
long ___ = 5234;
```

```
long long $$ = 42342;
```

```
unsigned gjnueirgeri = 453;
```

# Важно!!!

- Всички примери от предния слайд работят

- **Не правете така!!!**

- Това е много лоша практика и може да ви струва скъпо!

# Именуване на променливи – добри практики

- Когато пишете код и използвате променливи, които имат важна роля е важно да ги именувате така, че да е ясно каква роля имат!
- Не е добре променливите ви да имат много дълги имена
- Не е препоръчително да пишете на шльокавица, защото един ден кодът ви може да се чете от човек, който не говори български

# Примери

```
int age = 5;
```

```
double sum = 3;
```

```
char coolestSymbol = 'M'; //camelCase
```

```
float MyDegree = 5.6; //PascalCase
```

```
long remaining_slides = 5234; //space replacement
```

```
bool isOk= true; //добра практика за именуване на bool
```

```
bool hasBlueEyes = false;
```

# Представяне на константа в бройна система, различна от десетичната

- За да използвате шестнайсетична бройна система, трябва преди числото да напишете `0x`.
  - Например, за да присвоите на променлива `a` стойност 255, използвайки шестнайсетичното и представяне (FF), трябва да напишете `int a = 0xFF`.
- За да използвате двоична бройна система, от C++14 съществува и опцията `0b`
  - Например, за да присвоите на променлива `a` стойност 5, използвайки двоичното и представяне (101), трябва да напишете `int a = 0b101`.

# Константни променливи

- `const <тип><име на променливата> = <стойност>;`
- Стойността се задава при инициализация и не може да се променя след това
- Неинициализирана константа - `undefined behaviour`

# Примери

```
const int number; //неинициализирана константна променлива,  
                //повечето компилатори ще изведат грешка
```

```
//правилно създаване на константа
```

```
//добра практика е имената на променливи да са изцяло с главни букви
```

```
const int MY_NUM = 2;
```

```
const char TAB = '\t';
```

```
const double PI = 3.14;
```