

# ZAPIS PODATKOV V RAČUNALNIKU

III. gimnazija Maribor  
Informatika

# Kodiranje

## ⦿ Kaj je kodiranje?

- **predstavitev informacije z vnaprej dogovorjenimi znaki.**

V vsakdanjem življenju nenehno uporabljamo najrazličnejše kodne znake, kot so na primer:

- črke, številke, prometni znaki, dimni signali.

## ⦿ Kako pa komuniciramo z računalnikom?

- Človek komunicira z računalnikom **v jeziku, ki je blizu človeku.**
- Računalnik pa zna **prebrati** le podatke, ki so pretvorjeni v **dvojiška števila.**

# Informacije in podatki v računalniku

- ⦿ Računalnik je **digitalna elektronska** naprava, ki razume le informacijo posredovano v obliki **električnih impulzov**.
- ⦿ Električne impulze pa lahko enostavno predstavimo:
  - **v vodniku je električni tok - predstavimo s številom 1**
  - **v vodniku ni električnega toka - predstavimo s številom 0**
- ⦿ V računalniku so informacije predstavljene z **dvojiškimi števili (0, 1)**.

# Dvojiški ali binarni številski sistem

- Desetiški številski sistem:

cifre: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

desetiška števila: 23, 1176, 25678

- Dvojiški številski sistem:

cifri: 0, 1

dvojiška števila: 1101, 100, 111111

# Kodiranje števil:

- Kakšna števila vnašamo v računalnik ?
  - desetiška
- Kaj naredi računalnik s temi števili ?
  - pretvori v dvojiška

	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
Število:	<b>32</b>	<b>16</b>	<b>8</b>	<b>4</b>	<b>2</b>	<b>1</b>	
<b>2</b>	0	0	0	0	1	0	= 10
<b>9</b>	0	0	1	0	0	1	= 1001
<b>24</b>	0	1	1	0	0	0	= 11000

Desetiško :	0	1	2	3	4	5	6	7	8	9
Dvojiško :	0	1	10	11	100	101	110	111	1000	1001

# Dvojiški (binarni) številski sistem: (0,1)

Pretvorimo število  $55_{(10)}$  v  
dvojiško število  $X_{(2)}$

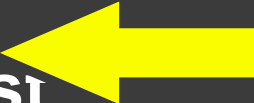
- ⊙  $55 = 2 * 27 + 1$
- ⊙  $27 = 2 * 13 + 1$
- ⊙  $13 = 2 * 6 + 1$
- ⊙  $6 = 2 * 3 + 0$
- ⊙  $3 = 2 * 1 + 1$
- ⊙  $1 = 2 * 0 + 1$



Pretvorimo število  $1567_{(10)}$  v  
dvojiško število  $X_{(2)}$

$$\begin{aligned}1567 &= 2 * 783 + 1 \\783 &= 2 * 391 + 1 \\391 &= 2 * 195 + 1 \\195 &= 2 * 97 + 1 \\97 &= 2 * 48 + 1 \\48 &= 2 * 24 + 0 \\24 &= 2 * 12 + 0 \\12 &= 2 * 6 + 0 \\6 &= 2 * 3 + 0 \\3 &= 2 * 1 + 1 \\1 &= 2 * 0 + 1\end{aligned}$$

## Zapis dvojiškega sistema: $11000011111_{(2)}$

- ⊙ Pretvorimo število  $11000011111_{(2)}$  v desetiški sistem  $X_{(10)}$
- ⊙  $11000011111_{(2)} = X_{(10)}$
- ⊙ 10 9 8 7 6 5 4 3 2 1 0 ..... označimo število mest 
- ⊙ izraz zapišemo v obliki polinoma:
- ⊙  $1*2^{10}+1*2^9+0*2^8+0*2^7+0*2^6+0*2^5+1*2^4+$   
 $+1*2^3+1*2^2 +1*2^1 +1*2^0$
- ⊙  $=1024+512+16+8+4+2+1$
- ⊙  $=1567$

# Kodiranje znakov

- Tudi pisni znaki so v računalniku kodirani z **nizom ničel in enic**.
- Znaki se kodirajo po standardu **ASCII** (*American Standard Code for Information Interchange*)
- Po ASCII standardu kodiramo znake z osem- mestnimi dvojiškimi števili (**8 bitno kodiranje**).



# Tabela ASCII kod

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	\$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	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	Neg. 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	End 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	Substitution	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	□

# ASCII

- Po standardu ASCII kodiramo znake z 8-mestnimi dvojiškimi števili. Na ta način lahko predstavimo **256 različnih znakov**.
- Kod ASCII **predpisuje le 128 znakov**. Neposredno se ta kod lahko uporablja le v **angleško govorečem področju**. Pri nas so **problem šumniki**.

# Kodiranje npr. šumnikov

- Pisave drugih dežel povečini vsebujejo znake, ki jih ASCII ne pozna (npr. šumniki).
- Najprej so to reševali tako, da so:
  - določenim kodam ASCII spremenimo osnovni pomen. Tako smo včasih **za naše šumnike uporabili kode oglatih oklepajev in zavutih oklepajev**. Pomanjkljivost take metode je, **da izgubimo prvotne znake**.
  - zato so se odločili, da osnovni nabor ASCII obdržimo, **manjkajoče znake pa predstavimo s kodami od 128 do 255**.
  - Ker pa je manjkajočih znakov veliko, več kot je prostih mest, je mednarodna organizacija za standarizacijo (International Standard Organisation – ISO) določila standarde **za različne skupine držav**.
  - V Sloveniji uporabljamo standard (imenujemo tudi kodna tabela) **ISO 8859 – 2** (imenovan tudi **Latin 2**) in **UTF-8**, v okolju Windows pa se namesto tega uporablja Microsoftov standard **Windows-1250**.

# Unicode

- Žal pa tudi ASCII z dopolnitvami ne zajema vseh različnih pisav. Zato se v računalništvu uveljavlja nov standard imenovan **Unicode**.
- V njem se uporablja za zapis **niz s 16 biti**. Z njim je možno zapisati  $2^{16}$  znakov to je **65536 znakov**.
- Primerjava:

Unicode

0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

 a

ASCII

0	1	1	0	0	0	0	1
---	---	---	---	---	---	---	---

 a

# Kodiranje slik

- Sliko v računalniku običajno shranimo tako, da jo z mrežo razdelimo na **slikovne točke**. Takšno slikovno točko na zaslonu imenujemo **piksel**. Vsak piksel na zaslonu pa žari v **določeni barvi**. Kako močno žari in v kakšni barvi zapišemo na različne načine.
- Najpogosteje uporabljamo za zapis **24 bitov**. Tako lahko zapišemo preko **16 milijonov barv** ( $2^{24}$ ). Tak način zapisa slik v računalniku imenujemo **točkovni ali bitni zapis**. Dodamo **8 bitov** za **prosojnost**, tako je skupaj **32 bitna globina** (npr. Windows).

# Koliko barv bomo uporabljali?

- ◎ **1 bit** - 1 barva + 1 ozadje
- ◎ 3 biti - 8 barv ( $2^3 = 8$ )
- ◎ 4 biti - 16 barv ( $2^4 = 16$ )
- ◎ 8 bitov - 256 barv ( $2^8 = 256$ )
- ◎ 16 bitov - 65.536 barv ( $2^{16} = 65.536$ )
- ◎ **24 bitov** - 16.777.216 barv ( $2^{24} = 16.777.216$ )
- ◎ **32 bitov** = 24 bitov + 8 bitov za prosojnost

1 bit

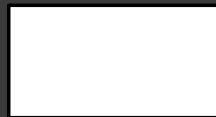


0



1

# 3 biti: 8 barv



**111 = bela**



**110 = rumena**



**101 = svetlom.**



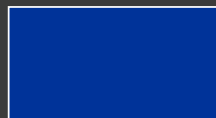
**100 = zelena**



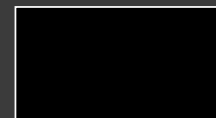
**011 = vijolična**



**010 = rdeča**



**001 = modra**



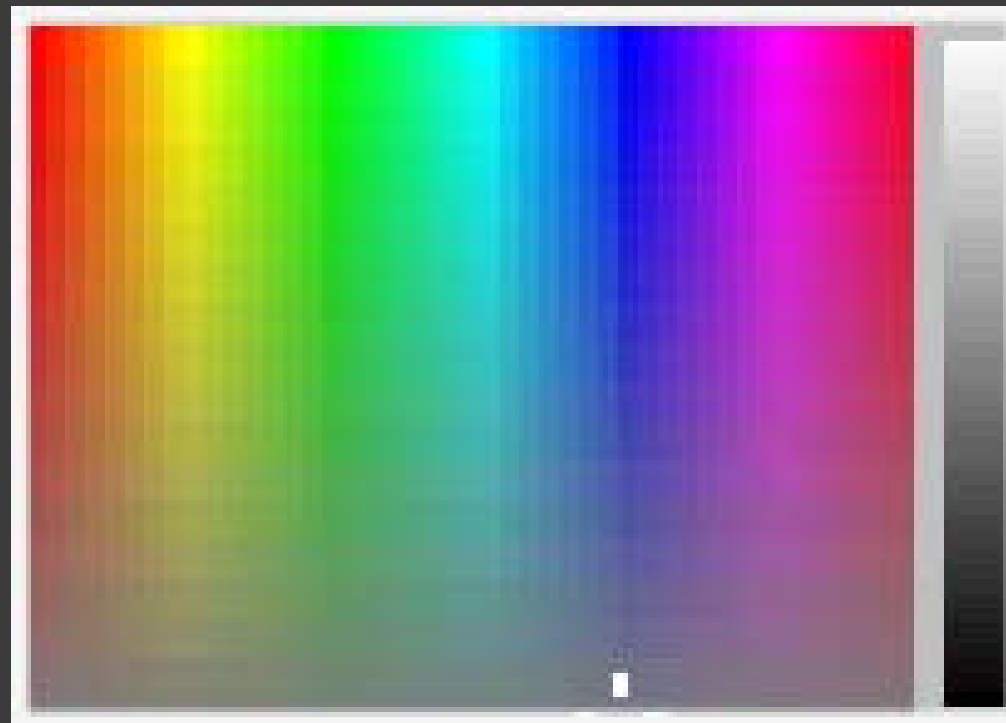
**000 = črna**



**8 bitov = 1 Byte (bajt) = 256 različnih barv**

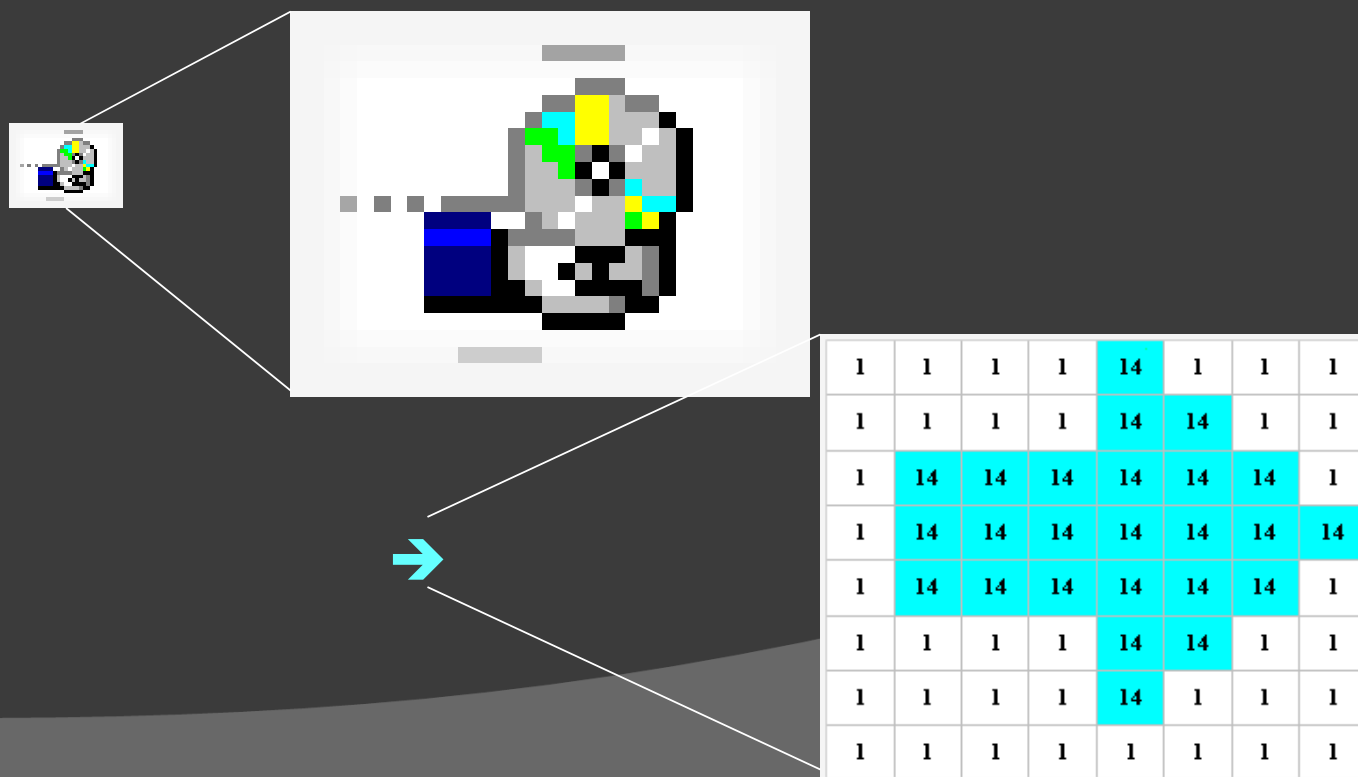


16 bitov - 65.536 barv



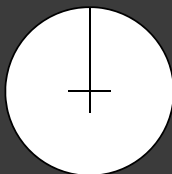
# Bitna slika

- Slika kodirana po pikah.



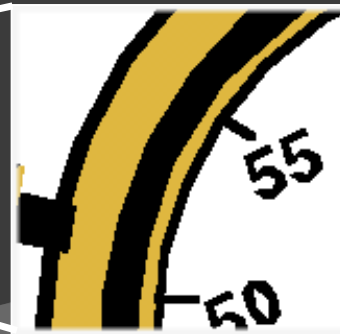
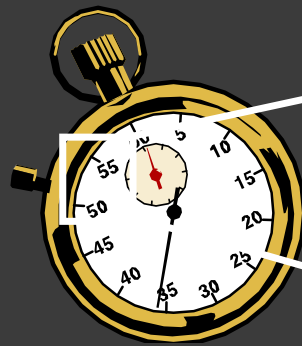
# Vektorska slika

- Poleg točkovnega pa se v računalništvu uporablja še **vektorski zapis slike**. Pri tem načinu je zaslon računalnika zamišljen kot koordinatni sistem, v katerega **matematično zapisujemo elemente likovnega jezika** npr. daljico, krog, elipso, pravokotnik in druge elemente slike. Seveda nam ob risanju ni potrebno matematično določevati predmetov. Mi samo rišemo predmete, vse ostalo opravi računalnik.
- Sliko na ta način lahko podamo tudi z navodili.
- Sliki predstavljeni z obliko in barvo elementov, pravimo **vektorska slika**.



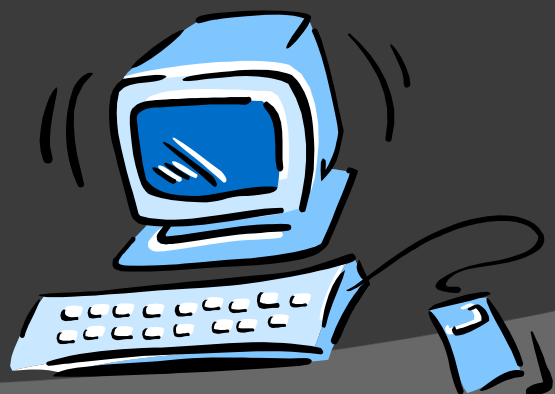
Krog je opisan s štirimi podatki:  $x, y$  – koordinati središča,  $r$  – polmer in  $z$  barvo, ki je v tem primeru bela.

Ob vsaki spremembi mora računalnik sliko seveda na novo izračunati in narisati, kar pa ob čedalje hitrejših računalnikih skoraj ni opazno.



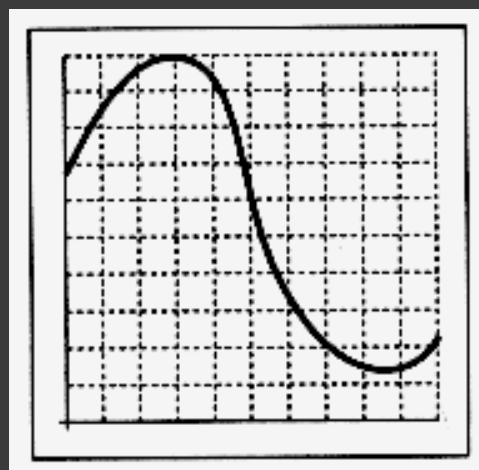
# Prednosti vektorske slike

- ◉ poljubno jo lahko povečamo (zmanjšamo),
- ◉ definiranje pisav.

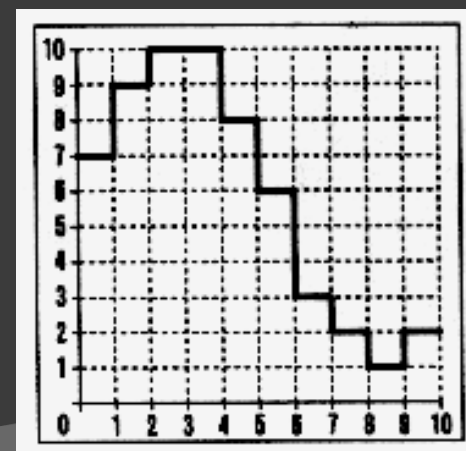


# Kodiranje zvoka

- zvok je analogna informacija,
- zvočna kartica



*Izvorni analogni signal*

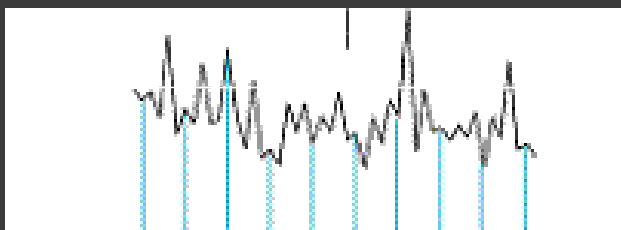


*Digitalni zapis z 10 meritvami*

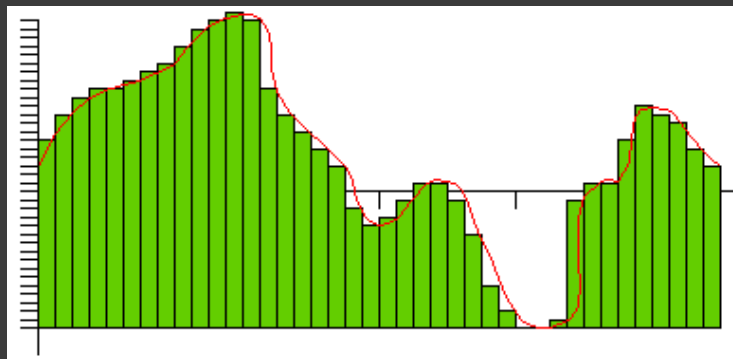
# Kodiranje zvoka

- ⦿ Zvok je **potujoče valovanje, ki ga slišimo**. Od izvira, imenujemo ga **zvočilo** se zvok širi po **zraku**, vodi ali kakšni drugi snovi do sprejemnika.
- ⦿ Zvočno valovanje ima določeno **amplitudo in frekvenco**. **Amplituda določa jakost zvoka, frekvenca pa njegovo višino**.
- ⦿ Časovno spreminjanje jakosti zvoka in njegove frekvence prikažemo s krivuljo, ki jo imenujemo **ovojnica**.
- ⦿ Za zapis zvoka z računalnikom je potrebno shraniti podatke o **časovnem spreminjanju ovojnice**.

# Kodiranje zvoka



- To storimo z enakomernim časovnim odčitavanjem odmika ovojnice od sredine, odčitano vrednost pa zapišemo s celim številom.



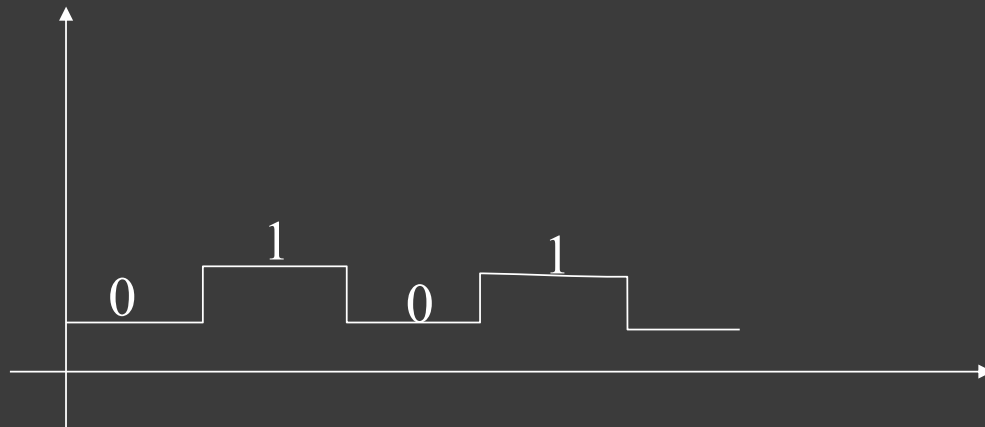


# Kodiranje zvoka

- Za kakovosten zvok moramo odmik ovojnice odčitati čim večkrat in dobljeno vrednost zapisati **s kar največ znaki**.
- Za kakovosten zapis zvoka (High Fidelity – Hi-Fi) glasbenih zgoščenk odčitamo vrednosti **44100 krat** v sekundi (frekvenca vzorčenja je torej 44,1 kHz) in jih zapisujemo s 16 in 24 biti.

# Kodiranje zvoka

- Nezahteven računalnik ima na voljo **1 bit**.



- Zvok tradicionalnih glasbil - znatno več bitov



# Kodiranje glasbe - MIDI

- ⦿ Kodni standard - MIDI  $\Rightarrow$  digitalni vmesnik za glasbo.
- ⦿ Računalniški zapis  $\Rightarrow$  namesto zvoka zapišemo noto, ki ta zvok proizvede.
- ⦿ MIDI koda predstavlja na inštrumentu zaigrano noto. Zapis vsebuje podatek o višini in glasnosti zaigranega tona in načinu, kako je bil zaigran.

