Nalogo rešite brez uporabe razredov za delo z nizi (kot so npr. String, StringBuilder in podobni)!
Razred String in njegove metode lahko uporabite le pri točkah 4) in 8).

Napišite razred `Znaki` in v njem statične metode za izpis znakov, definiranih na mreži 4 x 4 ter 8 x 8.

1) V razredu `Znaki` deklarirajte naslednje spremenljivke in tabele:

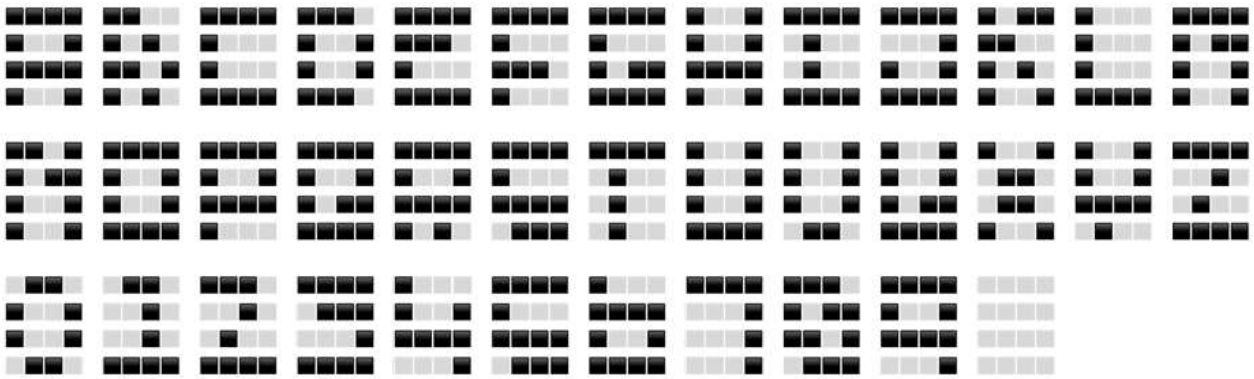
```
private static final char crnaPika = '\u2B1B'; // črn kvadratek
private static final char belaPika = '\u2B1C'; // prazen (bel) kvadratek

private static final char[] abeceda = {
    'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M',
    'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z',
    '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ' '};

  private static final short[] kodeZnakov16bit = {
      (short) 0b1111100111111001, // A
      (short) 0b1100101011011010, // B
      (short) 0b1111100010001111, // C
      (short) 0b1110100110011110, // D
      (short) 0b1111111010001111, // E
      (short) 0b1111100011101000, // F
      (short) 0b1111100010111111, // G
      (short) 0b1001100111111001, // H
      (short) 0b1111010001001111, // I
      (short) 0b1111000110011111, // J
      (short) 0b1011110010101001, // K
      (short) 0b1000100010001111, // L
      (short) 0b1111101110011001, // M
      (short) 0b1101101110011001, // N
      (short) 0b1111100110011111, // O
      (short) 0b0,                // TODO: dodajte znak P
      (short) 0b1111100110111111, // Q
      (short) 0b1111100111111010, // R
      (short) 0b1111100011110111, // S
      (short) 0b1111010001000100, // T
      (short) 0b1001100110011111, // U
      (short) 0b1001100110010110, // V
      (short) 0b1001100110111111, // W
      (short) 0b1001011001101001, // X
      (short) 0b1001100111110100, // Y
      (short) 0b1111001001001111, // Z
      (short) 0b0110100110010110, // 0
      (short) 0b0110001000101111, // 1
      (short) 0b0,                // TODO: dodajte znak 2
      (short) 0b1111011100011111, // 3
      (short) 0b1000100111110001, // 4
      (short) 0b1111100011110111, // 5
      (short) 0b1000111110011111, // 6
      (short) 0b1111000100010001, // 7
      (short) 0b1110101111010111, // 8
      (short) 0b1111100111110001, // 9
      0                           // presledek
  };
```

?

kjer tabela `kodeZnakov16bit` vsebuje znake abecede, predstavljene s 16 biti. Zapis `0b1111100111111001` je primer binarnega zapisa vrednosti spremenljivke, kjer `0b` napove, da gre za binarni zapis, nato pa sledijo biti od tistega z največjo težo proti tistemu z najmanjšo (0b [bit-15] [bit-14] ... [bit-2] [bit-1] [bit-0]). Spodnja shema prikazuje razporeditev bitov v znaku, kjer je 0 najnižji bit, 15 pa najvišji bit:

15	14	13	12
11	10	9	8
7	6	5	4
3	2	1	0

Znaki 16-bitne abecede so definirani, kot je prikazano na spodnji sliki:



Primer zapisa črke E:

```
(short)0b1111111010001111;
```

Tabelo `kodeZnakov16bit` dopolnite še z manjkajočima kodama znakov P in 2.

2) Definirajte metodo `izpisi16bit(short kodaZnaka)`, ki kot vhodni parameter sprejme 16-bitno kodo znaka in izpiše znak na zaslon.

Primer klica:

```
izpisi16bit((short)0b1111100111111001);
```

izriše znak A:

Delovanje metode preverite še na znakih P in 2.

Namig: *Za lažjo implementacijo metode uporabite bitne operatorje ter s pomočjo bitnega maskiranja preverite vrednost bita na določeni poziciji.*
Vrednost bita na določenem mestu lahko (neodvisno od ostalih bitov) preverimo z bitnim maskiranjem tako, da s pomočjo bitnega operatorja IN (&) ugasnemo vse bite, razen iskanega, ter preverimo rezultat. Pri tem si pomagamo z masko, to je številom, ki ima v bitni predstavitvi na iskanem mestu bit postavljen na 1, vsi ostali biti pa so 0. Če je rezultat (število & maska) enak 0, je iskani bit 0, če pa je rezultat enak maski, je iskani bit 1.

Poglejmo primer: poiščimo vrednost 4. bita pri številu 157 in pri številu 149 (bita sta zapisana krepko).

$157_{10} = 1001\mathbf{1}101_2$
$149_{10} = 1001\mathbf{0}101_2$
maska = $0000\mathbf{1}000_2$ (zanima nas 4. bit, zato je ta postavljen na 1)

```
   1001 1101 (število)                1001 0101 (število)
 & 0000 1000 (maska)                & 0000 1000 (maska)
  -----------                        -----------
   0000 1000                          0000 0000
```

3) Definirajte metodo `izpisi16bit(short[] nizZnakov)`, ki kot vhodni parameter sprejme tabelo 16-bitnih kod znakov in jih izpiše na zaslon. Pri tem naj dva zaporedna znaka razmakne za en kvadratek (za vsakim izrisanim znakom je en stolpec belih kvadratkov).

Primer klica:

```
izpisi16bit(new short[] {(short)0b1111100011101000, (short)0b1111100111111010,
(short)0b1111010001001111});
```

izpiše:

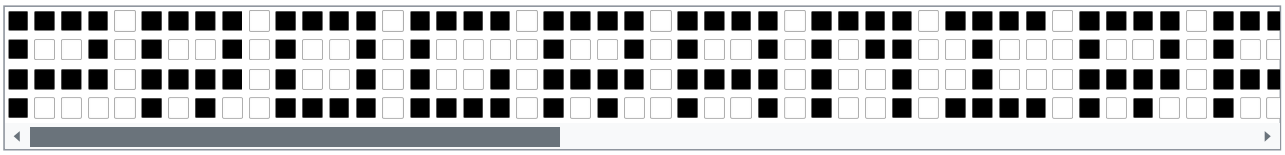
Delovanje metode preverite tudi z izpisom celotne abecede.

4) Definirajte metodo `izpisi16bit(String niz)`, ki kot vhodni parameter sprejme niz (`String`) in ga izpiše na zaslon. Pri tem znake v nizu preslika v črke definirane 16-bitne abecede. Če znaka ni v abecedi, namesto njega izpiše znak za presledek. Znake pred izpisom pretvorite v velike črke (kot so zapisane v tabeli `abeceda`). Pri izpisu predpostavite, da je širina konzole zadostna.

Primer klica:

```
izpisi16bit("Programiranje je zakon");
```

izpiše:



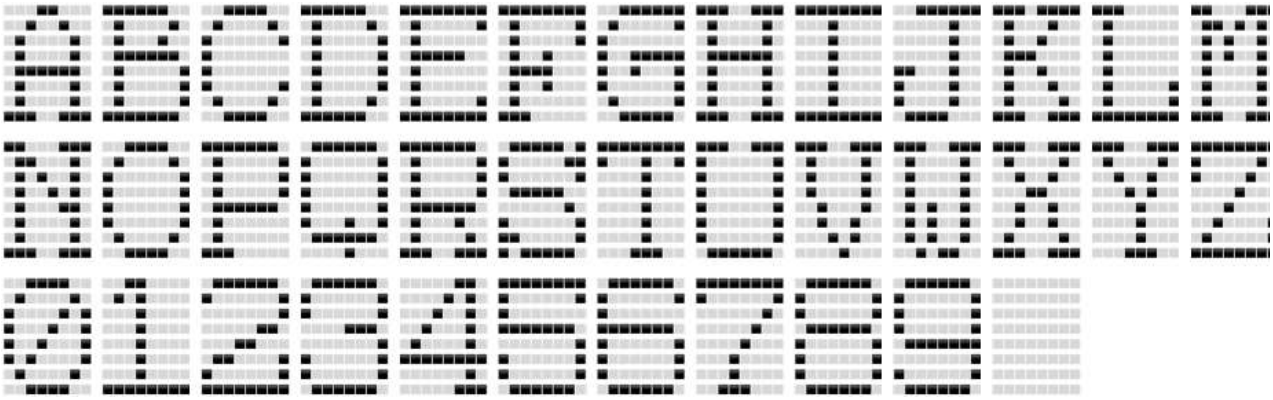
5) Med deklaracije razreda dodajte še tabelo:

```
 private static final long[] kodeZnakov64bit = {
       0b0001100000100100010000100100001001111110010000100100001011100111L,
       0b1111110001000010010001000111111001000001010000010100000111111110L,
       0b0011110001000010100000011000000010000000100000001010000100011100L,
       0b1111110001000010010000010100000101000001010000010100001011111100L,
       0b1111111101000001010000000111110001000000010000000100000111111111L,
       0b1111111101000001010000001010010000111100001001000010000000111000000L,
       0b0011111101000001100000001001111110010001100000010100000100111110L,
       0b1110011101000010010000100111111001000010010000100100001011100111L,
       0b1111111100010000000100000001000000010000000100000001000011111111L,
       0b0011111100000100000001000000010011000100100001001000010001111000L,
       0b1110111101000100010010000011100000100100001000100010000101110011L,
       0b1110000001000000010000000100000001000000010000010100000111111111L,
       0b1100011101101010010100100101001001000010010000100100001011100111L,
       0b1100011101100010010100100100101001000011001000010010000101110011L,
       0b0011110001000010100000011000000110000001100000010100001000111100L,
       0b0,                                                                  // TODO: dodaj znak P
       0b0111111010000001100000011000000110000001100010010111111000001000L,
       0b1111111001000001010000010100000101111110010001000100001011100111L,
       0b0111110110000011100000010111110000000010100000011100000110111110L,
       0b1111111110001001000010000000100000001000000010000000100000011100L,
       0b1110011110000001100000011000000110000001100000011000000101111110L,
       0b1110011101000010010000100100001000100010010000001010000001000L,
       0b1110011101000010010000100100001001010010010100100101001000101100L,
       0b1110011101000010001001000001100000100100001001000100001011100111L,
       0b1110011101000010001001000001010000001000000010000000100000011100L,
       0b1111111110000010100001000000100000010000001000001010000001111111L,
       0b0011110001000010100001011000100110010001101000010100001000111100L,
       0b0011000001010000000100000001000000010000000100000001000011111111L,
       0b0,                                                                 // TODO: dodaj znak 2
       0b0111111010000001100000010000111000000001100000011000000101111110L,
       0b0000011000001010000100100010001001000010111111110000001000000111L,
       0b1111111110000001100000001111111000000001100000011000000101111110L,
       0b0111111010000001100000001111111010000001100000011000000101111110L,
       0b1111111110000001000000010000001000000010000000100000001000000111000L,
       0b0111111010000001100000010111111010000001100000011000000101111110L,
       0b0111111010000001100000011000000101111111000000011000000101111110L,
       0
   };
```

ki vsebuje znake abecede, predstavljene s 64 biti (enako kot zgoraj predstavljen 16-bitni zapis; "L" na koncu pove, da je zapis v obliki tipa long), kot prikazuje spodnja shema, kjer je 0 najnižji bit, 63 pa najvišji bit:

63	62	61	60	59	58	57	56
55	54	53	52	51	50	49	48
47	46	45	44	43	42	41	40
39	38	37	36	35	34	33	32
31	30	29	28	27	26	25	24
23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8
7	6	5	4	3	2	1	0

Znaki 64-bitne abecede so definirani, kot je prikazano na spodnji sliki:



Primer zapisa črke A:

```
0b0001100000100100010000100100001001111110010000100100001011100111L
```

V tabeli `kodeZnakov64bit` definirajte tudi manjkajoča znaka P in 2.

6) Definirajte metodo `izpisi64bit(long kodaZnaka)`, ki kot vhodni parameter sprejme 64-bitno kodo znaka in izpiše znak na zaslon.

Primer klica:

```
izpisi64bit(0b0001100000100100010000100100001001111110010000100100001011100111L);
```

izpiše črko A:

S to metodo izpišite tudi znaka P in 2.

7) Definirajte metodo `izpisi64bit(long[] nizZnakov)`, ki kot vhodni parameter sprejme tabelo 64-bitnih kod znakov in jih izpiše na zaslon.

Primer klica:

```
izpisi64bit(new long[] {0b1111111101000001010000010100100001111000010010000100000011100000L,
0b0011000001010000000100000001000000010000000100000001000011111111L});
```

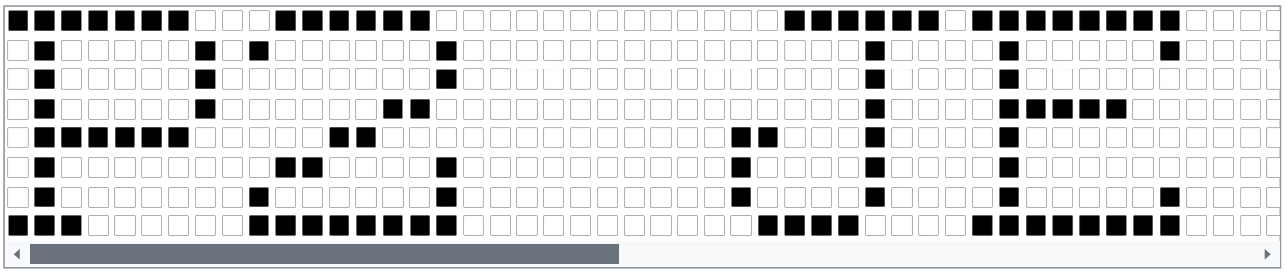
izpis:

8) Definirajte metodo `izpisi64bit(String niz)`, ki kot vhodni parameter sprejme niz (`String`) in ga izpiše na zaslon. Pri tem znake v nizu preslika v črke definirane 64-bitne abecede. Če znaka ni v abecedi, namesto njega izpiše znak za presledek. Znake pred izpisom pretvorite v velike črke. Pri izpisu predpostavite, da je širina konzole zadostna.

Primer klica:

```
izpisi64bit("P2 je super");
```

izpis:



9) Preverite delovanje programa še za spodnja dva klica:

```
izpisi64bit(4342219536296657468L);
```

```
izpisi64bit(1746410238858002085L);
```

za izpis naslednjih dveh znakov:

in

Kaj pa izpiše spodnji klic metode?

```
izpisi64bit(-36525672788885761L);
```

DODATNI IZZIVI

A) Abecedo dopolnite še s šumniki in ločili (uporabite vsaj vejico, piko, klicaj in vprašaj) ter ustrezno dopolnite tabelo 64-bitnih kod.

Nato izpišite stavek "Če programiram, še bolj uživam!" ter vaše polno ime in priimek.

B) Program dopolnite tako, da mu kot argument lahko podamo velikost znakov (4 oz. 8) ter poljuben niz, ki ga program nato izpiše z znaki 16-bitne oz. 64-bitne abecede.

Add submission

Submission status

Submission status	No submissions have been made yet
Grading status	Not graded

◄ Rešitev naloge

Jump to...

Rešitev naloge ►