Poročilo:

O izvedeni vaji:

07\_lav03\_vaja

Laboratorijske vaje

Vaja 7

Maj Mohar, G3A

Ljubljana, november 2024

Naloga 1

a) Napišite sled izvajanja danega programa.

b) Ugotovite, kaj program izpiše .

c) Program prepišite tako, da zanko do-while zamenjate z while.

public class N01\_03{

public static void main(String[] args){

char x='O'; // je črka O

final char y='A'; // to je konstanta

do{

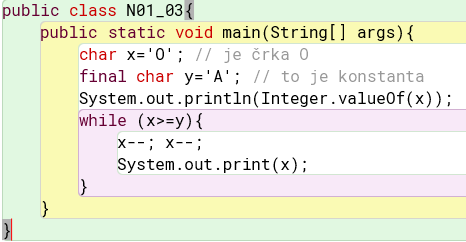
x--; x--;

System.out.print(x);

} while (x>=y);

}

}

1. Program char x, ki je na začetku 0 vsakič zmanjša za dve vrednosti, nato pa ga izpiše. Torej so vrednosti x: 0, M, K, I, G, E ,C ,A ,?, ker je v tebeli vrednost pripisana črki. Torej začetna vrednost x-a je 79 in se potem po dve zmanjšuje, kar pa program napiše kot črko, ker je definiran kot char.
2. MKIGECA?
3. .

Naloga 2

Ugotovite, kaj izpiše naslednji program (dokaz=sled+izpis).

public class Naloga3401{

public static void main(String[] args){

int i=4;

do {

System.out.print("To je ");

System.out.print("repeat oz. do-while zanka ");

System.out.printf("številka %4d",i);

System.out.println();

i = i + 2;

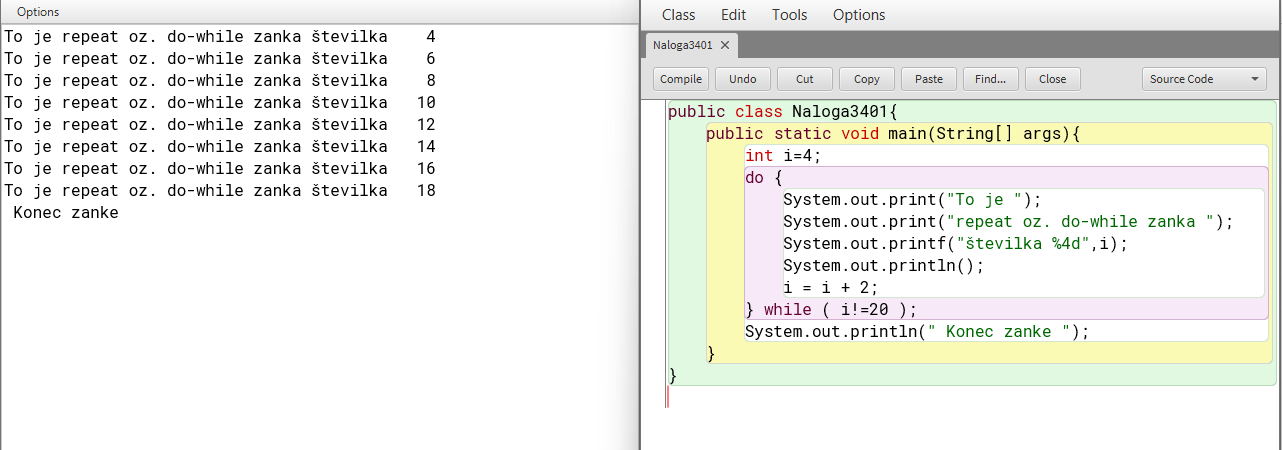
} while ( i!=20 );

System.out.println(" Konec zanke ");

}

}

Ta program napiše »To je repeat oz. do-while zanka številka (število)« dokler i ni enak 20, kar se zgodi po 8 ponovitvah. Program izpiše 8-krat zgornji stavek in števila: 4, 6, 8, 10, 12, 14, 16, 18 nato pa izpiše konec zanke.



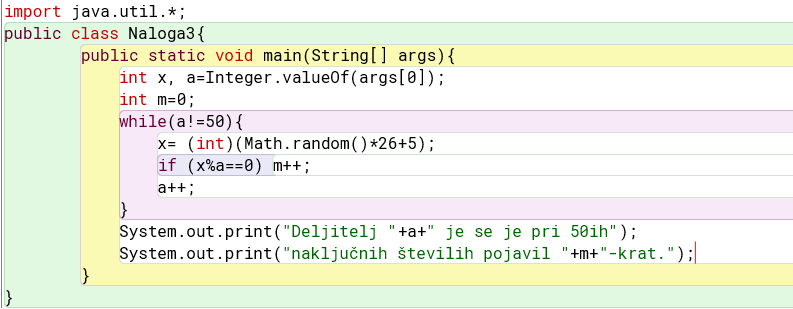
Naloga 3

Napišite program, ki generira 50 naključnih števil iz intervala [5..30] in izpiše, kolikokrat se je pojavil večkratnik

prebranega celega števila n. Število n naj bo argument pri zagonu programa.

Postopek je podan opisno s (pseudo)programom:

* preberi n
* inicializiraj števec (nastavi začetno vrednost števca) večkratnikov
* (inicializiraj generator naključnih števil, če je to potrebno)
* ponovi petdesetkrat
* ustvari naključno število iz intervala [5..30]
* če je dobljeno število večkratnik števila n, povečaj števec večkratnikov
* izpiši števec večkratnikov



Naloga 4

a) Napišite sled izvajanja spodnjega programa.

b) Ugotovite, kaj izpiše.

c) Kaj se spremeni, če v zanki pred if dodamo ukaz za negacijo spremeljivke pogoj? Razložite odgovor.

d) Kaj se spremeni, če v zanki pred if spremenimo vrednost spremeljivke pogoj v

(!pogoj)?!pogoj:pogoj;? Razložite odgovor.

public class N01\_05 {

public static void main(String[] args){

int i;

boolean pogoj;

for(i=1; i<=15; i++){

pogoj = (i%4==0) || (i%3==0) && (i%5==0) ;

//c// pogoj = !pogoj;

//d// pogoj = (!pogoj)?!pogoj:pogoj;

if (pogoj)

System.out.print('\*');

else

System.out.print('!');

}

}

}

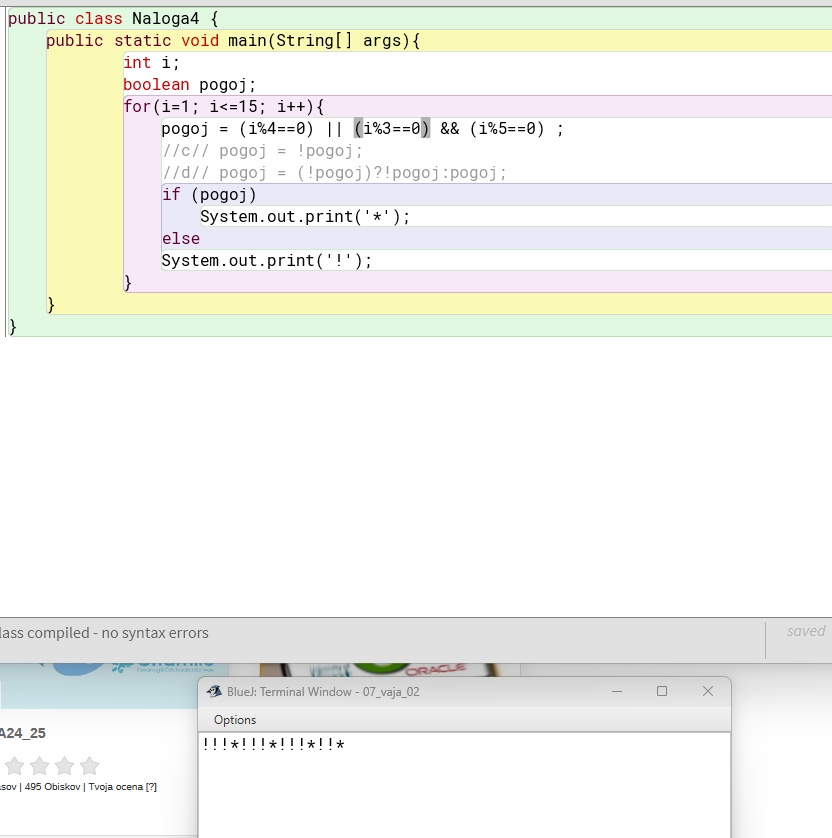
Najprej se izpiše 3x !, ker dokler en pogoj v enačbi ni true, je oboje false.

Potem pa, ko je i=4 je deljiv z 4 in je pogoj true, zato se izpiše !.

Prednost ima && in nato se šele izvede \*.

c)obrnejo se vrednosti

d) vse postanejo zvezdice, ker je pogoj vedno enak



Naloga 6

Napišite program N01\_06, ki za podano celo število n izračuna in izpiše:

a. Vsoto števk (npr. vsota števk števila 144 je 9).

b. Število števk (npr. število 144 ima 3 števke).

c. Največjo števko.

d. Števko, ki največkrat nastopa v številu. (če znate, sicer neobvezno)

Upoštevajte dejstvo, da lahko podano tudi negativno število in da ima vsako prebrano število vsaj eno števko. Pri številu lahko zanemarite le vodilne ničle, ki ne vplivajo na vrednost števila. Npr. število števk števila 0017 je 2.

**import** java.lang.\*;

/\*\*

\* **@author** Maj

\* **@version** 12. 11. 2024

\*/

public class Naloga6 {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String[] args) {

final int število = Integer.valueOf(args[0]);

int vsota=0,x, koliko\_števk=0;

//vsota števk

**for** (int i = število; i>0;){

x = i%10;

vsota += x;

i/=10;

}

System.out.println("Vsota števk tega števila je: "+vsota+".");

//število števk

**for** (int i = število; i>0;){

x = i%10;

koliko\_števk++;

i/=10;

}

System.out.println("Število števk je: "+koliko\_števk+".");

//največja števka

int max=0;

**for** (int i = število; i>0;){

x = i%10;

**if** (x>max) max=x;

i/=10;

}

System.out.println("Največja števka je: "+max+".");

//ponovitev števk

int nič=0, ena=0, dva=0, tri=0, štiri=0, pet=0, šest=0, sedem=0, osem=0, devet=0, največkrat\_števka=0;

**for** (int i = število; i>0;){

x = i%10;

**switch**(x){

**case** 0**:** nič++; **break**;

**case** 1**:** ena++; **break**;

**case** 2**:** dva++; **break**;

**case** 3**:** tri++; **break**;

**case** 4**:** štiri++; **break**;

**case** 5**:** pet++; **break**;

**case** 6**:** šest++; **break**;

**case** 7**:** sedem++; **break**;

**case** 8**:** osem++; **break**;

**case** 9**:** devet++; **break**;

}

i/=10;

}

//koliko\_števk

int števke=0;

**if** (nič>števke){

števke=nič;

največkrat\_števka=0;

}

**if** (ena>števke){

števke=ena;

največkrat\_števka=1;

}

**if** (dva>števke){

števke=dva;

največkrat\_števka=2;

}

**if** (tri>števke){

števke=tri;

največkrat\_števka=3;

}

**if** (štiri>števke){

števke=štiri;

največkrat\_števka=4;

}

**if** (pet>števke){

števke=pet;

največkrat\_števka=5;

}

**if** (šest>števke){

števke=šest;

največkrat\_števka=6;

}

**if** (sedem>števke){

števke=sedem;

največkrat\_števka=7;

}

**if** (osem>števke){

števke=osem;

največkrat\_števka=8;

}

**if** (devet>števke){

števke=devet;

največkrat\_števka=9;

}

System.out.println("Največkrat ponovljena števka je bila: "+največkrat\_števka+".");

}

}

Naloga 7

Spodnji program naj bi vneseno število tipa 'long' razcepil na posamezne števke in jih izpisal na prikazan način:

> java Nal3307 1487503

1\*1

4\*\*\*\*4

8\*\*\*\*\*\*\*\*8

7\*\*\*\*\*\*\*7

5\*\*\*\*\*5

00

3\*\*\*3

public class Nal3307{

public static void main(String[] args){

long stevilo=0,pom,i;

int prvo=0;

long mnozitelj;

stevilo=Long.valueOf(args[0]);

while(stevilo>0){

/\* poiščemo prvo števko in določimo potenco kofaktorja 10 \*/

pom=stevilo;

mnozitelj = 1;

while (pom != 0 ){

prvo = (int)(pom % 10);

mnozitelj = mnozitelj \* 10;

pom = pom / 10 ;

}

mnozitelj = mnozitelj / 10;

System.out.print(prvo); /\* izpis števke \*/

/\* izpis zvezdic, kot jih določa velikost števke \*/

for (pom = 1 ; pom<= prvo ;pom++)

System.out.print('\*');

System.out.println(prvo); /\* izpis števke \*/

stevilo = stevilo % mnozitelj; /\* odrežemo št. prvo števko \*/

}

}

}

Program vsebuje algoritmično napako. Poiščite jo s pomočjo sledi izvajanja programa!

Program ima napako, ki je z manjšimi popravki ne moremo popraviti. Da se jo popraviti z preureditvijo logike.

Naloga 8

Program V02\_05 ugotavlja, koliko naključnih števil je potrebno generirati, da z njihovo vsebino lahko zapolnimo vseh spodnjih 16 bitov. Recimo: št. 5 pokrije bita 0 in 2, število 10 bita 3 in 1. Ti števili skupaj zapolnita spodnje 4 bite... Naslednje število 7 bi bilo nekoristno, ker so njegove enice že vsebovane v številih 5 in 10. Program med izvajanjem izpiše vsa tista števila, ki 'prispevajo' svoje bite oz. so koristna, nekoristnih pa ne. Ob koncu izpiše še , koliko števil je bilo potrebno generirati, koliko od tega je bilo koristnih (ustreznih) in kolikšen je bil procent koristnih. Naključna števila izbirajte iz intervala:

a. [0-33000]

b. [0-Integer.MAX\_VALUE]

Primer izpisa pri enem izmed zagonov programa:

18430

4860

11157

32907

(65535 == vsi biti) --> 65535 vseh generiranih: 17, ustreznih: 4, = 23.52[%]

**import** java.lang.\*;

/\*\*

\* Program to generate random numbers and count how many contribute to a cumulative bitwise OR operation.

\*

\* **@author** Maj

\* **@version** 13. 11. 2024

\*/

public class Naloga8 {

/\*\*

\* Main application method.

\*

\* **@param** args - Command line arguments (not used in this program)

\*/

public static void main(String[] args) {

int konec = 0, random = 0, uspešno = 0, število = 0;

**while** ((konec & 0b1111111111111111) != 65535) {

število++;

random = (int)(Math.random()\*Integer.MAX\_VALUE);

//random = (int)(Math.random()\*33000);

int temp1 = konec;

konec = konec | random;

**if** (konec != temp1) {

uspešno++; // število uspešnih

System.out.println(random);

}

}

System.out.printf("Vseh generiranih števil: %d, ustreznih: %d, = %.2f%%%n",

število, uspešno, ((double) uspešno / število) \* 100);

}

}

Naloga 9

Program V02\_06 predstavlja simulacijo streljanja serije desetih strelov v tarčo. Odstopanje strela od centra tarče znaša največ 60% oziroma +-6 v x in v y smeri (6 pomeni razdaljo od centra tarče oz. 10, kar pomeni, da je strelec pač zadel 4). Programsko izvedite simulacijo 10-tih strelov in določite rezultat glede na doseženo vrednost strela. Točnost strela se določa na ena decimalko natančno.

Primer izpisa simulacije streljanja:

serija 1: Vrednost strela: 10.0 akumulacija: 10.0

serija 2: Vrednost strela: 7.3 akumulacija: 17.3

serija 3: Vrednost strela: 4.6 akumulacija: 21.9

serija 4: Vrednost strela: 8.2 akumulacija: 30.1

serija 5: Vrednost strela: 5.9 akumulacija: 36.0

serija 6: Vrednost strela: 5.2 akumulacija: 41.2

serija 7: Vrednost strela: 5.4 akumulacija: 46.6

serija 8: Vrednost strela: 8.9 akumulacija: 55.5

serija 9: Vrednost strela: 6.9 akumulacija: 62.4

serija 10: Vrednost strela: 6.2 akumulacija: 68.6

Rezultat: 68.6

Program naj dodatno izpiše še :

¬ vrednost najboljšega strela in v kateri seriji je bil dosežen,

**import** java.lang.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 17. 11. 2024

\*/

public class Naloga {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String[] args) {

double x, y, razdalja, točke=0, vsota=0, max=0;

**for** (int i = 1; i<=10; i++){

x = Math.round((double)(Math.random()\*12-6)\*10.0)/10.0;

y = Math.round((double)(Math.random()\*12-6)\*10.0)/10.0;

razdalja = 10-Math.round(Math.sqrt(x\*x+y\*y)\*10.0)/10.0;

točke = točke + 10-razdalja;

max=(razdalja>max**?**razdalja**:**max);

System.out.print("serija "+ i+ ": ");

System.out.printf("Vrednost strela: %.1f akumulacija %.1f", razdalja, točke);

System.out.println("");

}

System.out.printf("Rezultat: %.1f \n", točke);

System.out.printf("Najboljši strel: %.1f", max);

}

}

Naloga 11

Program V01\_01c izriše okvir po zunanjem robu navideznega zaslona velikosti 80x25. Vogali so predstavljeni z znaki +, vertikale z vertikalnimi črtami, horizontale s horizontalnimi črtami. Slika izpisa se nahaja levo od tega besedila.

**import** java.lang.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 18. 11. 2024

\*/

public class Naloga11 {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String[] args) {

int višina, širina=80;

**for** (višina=25; višina>0; višina--){

**if** (višina==25||višina==1){

System.out.print("+");

**for**(int i=širina-2; i>0; i--)System.out.print("-");

System.out.print("+");

}

**else** {

String a = "|";

System.out.printf("| %"+(širina-2)+"s", a);

}

System.out.println("");

}

}

}

Naloga 10

Napišite program N01\_01, ki na 15 naključnih legah na zaslonu prikaže 15 naključnih malih črk. Predpostavite, da je tekstovna resolucija (dimenzije) zaslona omejena na 15 vrstic s po 80 stolpcev. V tekstovnem načinu imate torej na voljo lege od (1,1) do (80,25) oziroma (0,0) do (79,24). Rezultat naj bo recimo nekaj takega:

**import** java.lang.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 18. 11. 2024

\*/

public class Naloga10 {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String args[]){

int crke=0;

**for**(int i=0; i<25;i++){

**if**(crke>14)**break**;

int n=(int)(Math.random() \* 100); //zato, da dobimo neko število

**for**(int j=0; j<80;j++){ //da je šansa 80%

**if**(n==j){

char randomChar = (char) ('a' + Math.random() \* 26);

System.out.print(randomChar );

crke++;

}

**else**{

System.out.print(" ");

}

}

System.out.println();

}

}

}

A white background with black text

Description automatically generated

**import** java.lang.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 18. 11. 2024

\*/

public class Naloga14 {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String[] args) {

char znak = args[0].charAt(0);

int število = Integer.valueOf(args[1]);

**if** (število - 2\*(število/2)==0)System.out.println("Romb z sodim številom vrstic ni možen!");

**else**{

int polovica\_romba= število/2;

int vrstica = (število+2)\*3;

**for**(int i=vrstica; i>0;i--) System.out.print(znak);

System.out.println("");

int romb=1;

boolean čez\_pol=**false**;

**for** (int i = 0; i<število;i++){

int leva=(vrstica-romb)/2;

**for**(int j=leva;j>0;j--)System.out.print(znak);

**for**(int j=romb;j>0;j--)System.out.print(" ");

**for**(int j=leva;j>0;j--)System.out.print(znak);

System.out.println("");

**if**(i>=polovica\_romba|| čez\_pol==**true**){

čez\_pol=**true**;

romb=romb-2;

}

**else** romb=romb+2;

}

**for**(int i=vrstica; i>0;i--) System.out.print(znak);

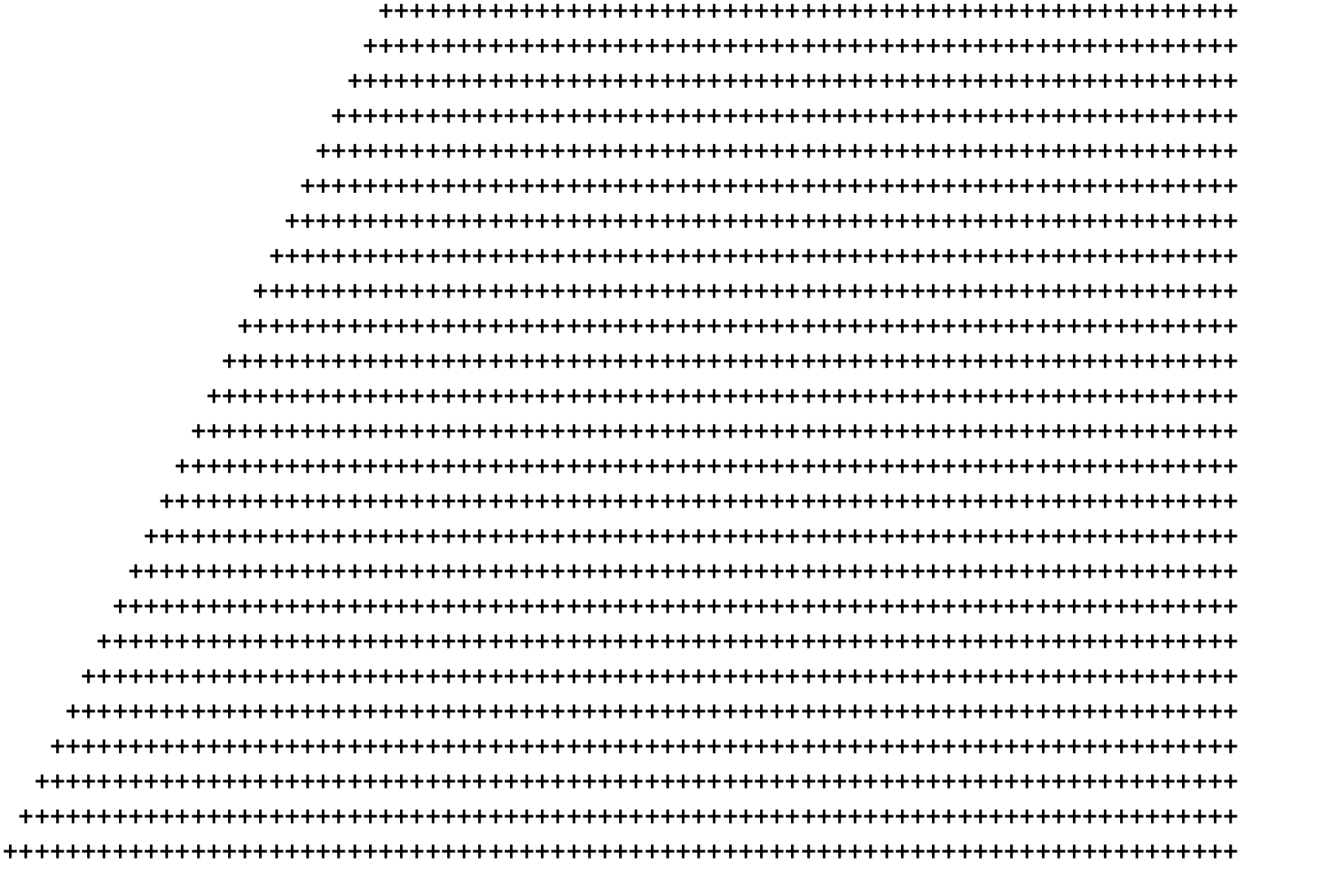
}

}

}

A computer code with text

Description automatically generated with medium confidence

A black and white triangle with small dots

Description automatically generated with medium confidence

A line of dots with a star

Description automatically generatedA row of black dots

Description automatically generatedA pyramid of dots

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated

**import** java.lang.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 18. 11. 2024

\*/

public class Naloga15 {

/\*\*

\* Glavna metoda aplikacije - spisite komentar po svoji meri

\*

\* **@param** arg[0] - primer parametra glavne funkcije

\*/

public static void main(String[] args) {

int vrstica = 80, stolpec = 16;

int count = 0, odštevanje=0;

boolean čez\_sred=**false**;

int progres;

**for** (int i = stolpec; i>1; i--){

progres=count;

**if** (count==0){

System.out.print("+");

**for** (int x=vrstica-2; x>0;x--) System.out.print("-");

System.out.print("+");

}

**else**{

**for** (int j = count; j>0;j--){//do zaćetka kvadrata

System.out.print("| ");

}

System.out.print("+---");

**for** (int j = vrstica-count\*4\*2-8; j>0;j--){//od sredine do konca

System.out.print("-");

}

System.out.print("---+");

**for** (int j = count; j>0; j--){

System.out.print(" |");

}

}

**if** (count == stolpec/2-1|| čez\_sred==**true**){

čez\_sred=**true**;

count--;

}

**else** count++;

System.out.println("");

}

}

}