|  |
| --- |
| **Utrjevanje:**   * **ponavljanja.** |

|  |
| --- |
| Metode reševanja in sredstva javanskega jezika omejite zgolj na sredstva, ki so bila predstavljena pri pouku ali v okviru vaj *(algoritmično se vse dane probleme da rešiti brez rabe pretvorbe v niz ali tabelo. Vprašanje je ali ste to sposobni narediti; res pa je tudi, da je lahko dobra rešitev tudi z omenjeno pretvorbo).* |

**Naloga 1**

a) Napišite sled izvajanja danega programa. Na podlagi sledi ugotovite, kaj program izpiše .   
b) 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);

}

}

**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 ");

}

}

**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('!');

}

}

}

**Naloga 5**

Napišite program R3305, ki podano desetiško vrednost izpiše v obliki binarnega števila. (dodajte še izpis v osmiški in šestnajstiški obliki: ta dva dosežete z oblikovanjem načina izpisa). Referenca : Java,Class Formatter, <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Formatter.html>, oktober 2022, poglejte možnosti o-osmiško in x-šestnajstiško, in uporabite v primeru izpisa desetiške vrednosti 18 : System.out.printf(" %d %o %x ",18,18,18)

**Naloga 6**

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

1. Vsoto števk (npr. vsota števk števila 144 je 9).
2. Število števk (npr. število 144 ima 3 števke).
3. Največjo števko.
4. 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.

**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!

**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:

1. [0-33000]
2. [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[%]

**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,
* vrednost najslabšega strela ter serija, v kateri je bil dosežen.

|  |
| --- |
| Opomba:  *Naloge, ki so vezane na 'risanje' v terminal uporabljajo linijski način izrisovanja; torej je smer izrisa vedno 'desno in dol' oz. najprej izrišete vsebino prve vrstice, nato druge, nato tretje, … vračanja nazaj ni !* |

**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:



**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.

**Naloga 12**

Program V02\_N01 je namenjen risanju funkcij v sistemih z linijskih prikazom. Velikost prikaza je omejena na okno velikosti 80x25. Ugotovite, katere funkcije program v dani obliki lahko prikazuje in v poročilo dodajte zaslonske slike rezultatov izvajanja vseh posameznih 'vgrajenih' funkcij. Zaradi 'lepšega prikaza' program ne izrisuje zgolj ovojnice funkcije, temveč 'pobarva' celotno površino med ovojnice in ordinato koordinatnega sistema. (Pripravljenih je 5 funkcij, ena je aktivna, 4 so v kodi komentirane.)

public class V02\_N01 {

public static void main(String[] args ) {

int y=0, x=0;

int lines=24;

for(int line=0;line<25;line++){

for (int col=0;col<80;col++){

y = 25-line; x=col;

if (y<=x) // y=x

//if (y<= 0.25\*x) // y=x/4 : položnejša

//if (y<= Math.log(x) ) // y=log(x) :logaritem

//if (y<= Math.log(x\*4) ) // y=log(x\*4) :strmina

//if (y<=24\*Math.sin(x\*(Math.PI/80))) // sinus z A=24

// in raztegom PI/80

System.out.print('+');

else

System.out.print(' ');

}

System.out.println();

}

}

}

**Naloga 13\***

S programom V02\_04 v ravnini velikosti v ravnini velikosti 80x25 izrisujemo pravokotnike. Vhodni prametri programa so pozicija (x,y) in velikost (a,b) pravokotnika. Program 'izriše' podan pravokotnik na podanih pozicijah kot prazen prostor na ravnini znakov +. Pravokotnik se mora vedno izrisati v celoti, dokler njegova velikost ne presega velikosti dane ravnine. V tem primeru izrišete pravokotnik v velikosti 80x25(hm, prazne zaslon ?). Izris mora biti trpežen; neustrezen vnos uporabnika moderiramo (!), da se pravokotnik vedno izriše v celoti. Dimenzijsko neustrezni pravokotniki se transformirajo v obseg dimenzij [(1,1),(80,25)] oz. [(0,0),(79,24)]. Za primer podajam nekaj naborov testnih podatkov:

1. -2,-2,4,4
2. 89,-2,4,4
3. 89,60,4,4
4. 89,6,4,400
5. 6,6,400,-20

Pri poljubnem se mora pravokotnih v celoti izrisati na zaslonu.

**Naloga 14**

Napišite program, s katerim zapolnite zaslonsko sliko s poljubnim znakom. Sredi zaslonske slike naj bo prazen prostor, ki predstavlja romb višine n. Program naj se kliče kot : **N02\_06 *znak* *visina***, oziroma za spodnji primer:

* java N02\_06 c 5

ccccccccccccccccccccc pri generiranju leve slike je bilo uporabljeno:

cccccccccc cccccccccc visina = 5 + 2 = 7

ccccccccc ccccccccc širina = (5+2) \* 3 = 21

cccccccc cccccccc

ccccccccc ccccccccc romb je centiran v polju velikosti 21x7

cccccccccc cccccccccc

ccccccccccccccccccccc

**Naloga 15\***

Spišite program V02\_N3 (kot varianto programa V01\_01c iz predhodnega besedila), ki na zaslon velikosti 80x25 znakov izrišite lik kot je prikazan spodaj.

+------------------------------------------------------------------------------+

| +----------------------------------------------------------------------+ |

| | +--------------------------------------------------------------+ | |

| | | +------------------------------------------------------+ | | |

| | | | +----------------------------------------------+ | | | |

| | | | | +--------------------------------------+ | | | | |

| | | | | | +------------------------------+ | | | | | |

| | | | | | | +----------------------+ | | | | | | |

| | | | | | | +----------------------+ | | | | | | |

| | | | | | +------------------------------+ | | | | | |

| | | | | +--------------------------------------+ | | | | |

| | | | +----------------------------------------------+ | | | |

| | | +------------------------------------------------------+ | | |

| | +--------------------------------------------------------------+ | |

| +----------------------------------------------------------------------+ |

+------------------------------------------------------------------------------+

Prikazan primer je izrisan v velikosti 80x16, vsak pravokotnik znotraj prejšnjega je dimenzij (a-8, b-2) lahko pa se odločite tudi drugače, dokler je pravokotnik v pravokotniku in glede na večjega centriran v njegovem središču.

**Naloga 16\***

Program V02\_N9 je (težja) varianta programa V02\_N3, predstavlja eno samo 'črto' , ki se ciklično razvije iz centra slike. Spišite ga.

+------------------------------------------------------------------------------+

| +----------------------------------------------------------------------+ |

| | +--------------------------------------------------------------+ | |

| | | +------------------------------------------------------+ | | |

| | | | +----------------------------------------------+ | | | |

| | | | | +--------------------------------------+ | | | | |

| | | | | | +------------------------------+ | | | | | |

| | | | | | | +----------------------+ | | | | | | |

| | | | | | | | -------------------+ | | | | | | |

| | | | | | | +--------------------------+ | | | | | |

| | | | | | +----------------------------------+ | | | | |

| | | | | +------------------------------------------+ | | | |

| | | | +--------------------------------------------------+ | | |

| | | +----------------------------------------------------------+ | |

| | +------------------------------------------------------------------+ |

| +--------------------------------------------------------------------------+