Poročilo:

O izvedeni vaji:

12\_lav03\_vaja

Laboratorijske vaje

Vaja 12

Maj Mohar, G3A

Ljubljana, december 2024

**Naloga 1**

Dana je razredna metoda s kodnim/razvojnim imenom x\_32\_v4.

static int x\_32\_v4(long abc){

int ll=1;

for(;(abc/=10)!=0;ll++);

return Math.max(1,ll);

}

* Analizirajte njeno delovanje in ugotovite, kaj počne
  + Program prešteje število mest v long številki, ki jo podamo kot argument
* Glede na ugotovitev dokažite pravilnost (ugotovljenega) delovanja (dokazi bazirajo na osnovi sledi izvajanj).Dokazi morajo pokrivati/dokazovati pravilnost/ vseh vstopnih možnosti za dani postopek (oziroma vsako vrednost argumenta).
  + Najprej inicializira spremenjivko ll, ki šteje ponovitve for zanke
  + Potem pa ponavlja deljenje z 10 in inkrementira ll, dokler rezultat deljenja z 10 ni 0.
  + Nato vrne maksimalno vrednost od vrednosti 1 in ll
* Spišite demonstracijski program, ki bo pokazal pričakovano delovanje metode pri klicu s poljubno celoštevilsko vrednostjo.
  + A computer screen with text and numbers

    Description automatically generated
* Obrazložite, zakaj iz metode vračamo večjo od specificiranih vrednosti.
  + Da dobimo realen rezultat te zanke, in ne dobimo vrednosti 1
* Predlagajte ime metode, ki bo dovolj opisno za funkcijo, ki jo opravlja.
  + prestejStevilskaMesta(long abc){}

**Naloga 2**

V podjetju imamo 100 zaposlenih delavcev. Podatki o plačah so shranjeni v tabeli 'place'. Identiteta(ID) oz. šifra zaposlenega je predstavljena z indeksom v tabeli. V tabelo place vpišite naključne vrednosti iz intervala [500..15000].

1. Programsko izpišite ugotovitev, ali je bila izplačana najvišja plača (15000).
2. Spišite metodo, ki izračuna in vrne vsoto vseh plač ('maso' plač). Vrnjeno izpišite.
3. Razredna metoda izpisite\_IDje(tabela\_plac) naj izpiše plače vseh zaposlenih, ki imajo plačo 680. Spišite jo.

**Naloga 3**

Dano je zaporedje: 2,4,5,11,23,25,33,55,88,89,91,93,98,103,110,130

1. Napišite sled izvajanja binarnega iskanja za podatek 98
   * Najprej preveri ali je iskano število večje ali manjše od sredinske vrednosti
   * Nato gre iskati v zgornji polovici. Tam spet preveri ali je iskana vrednost manjša ali večje od sredinske vrednosti, zato gre iskati od 89 do sredine-1
   * V naslednjem koraku najde vrednost
2. Napišite sled izvajanja binarnega iskanja za podatek 22. Na vsakem koraku mora biti razvidno kaj iščemo, kateri je sredinski element, kateri del tabele pregledujemo, kateri del tabele se na vsakem koraku zavrže.

|  |  |  |  |
| --- | --- | --- | --- |
| **zac** | **kon** | **sred** | **tab[sred]** |
| 0 | 15 | 7 | 55 |
| 0 | 7-1 = 6 | 6/2 = 3 | 11 |
| 3+1 = 4 | 6 | 5 | 25 |
| 4 | 5-1 = 4 | 4 | 23 |

Potem konča program, ker je zac enak koncu, kar pomeni, da nismo našli iskanega števila.

**Naloga 4**

Napišite program, s katerim v tabelo t vpišete 12 naključnih velikih črk angleške abecede. Pri vnosu podatkov zagotovite, da bo vsebina tabele urejena naraščajoče. Nato izpišite ugotovitev, ali je v tabeli črka x. Črko x dobite z generatorjem naključnih števil.

1. Uporabite linearno iskanje
2. Uporabite binarno iskanje

Po izvedbi vsakega izmed postopkov iskanja izpišite tudi število izvedenih primerjav.

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

**import** java.util.\*;

/\*\*

\* Opis:

\*

\* **@author** Maj

\* **@version** 18. 12. 2024

\*/

public class Naloga4 {

public static void main(String[] args) {

char[] t = {'B','D','F','G','L','M','P','S','T','V','Y','Z'};

char crka = generirajCrko();

int index = poisciCrkoLinearno(crka, t);

**if**(index==-1)System.out.println("Črke "+crka+" nismo našli v tabeli.");

**else** System.out.println("Črko "+crka+" smo v tebeli našli na indexu "+index+".");

}

public static char generirajCrko(){

char[] crke = {'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'};

char a = crke[(int)(Math.random()\*26)];

**return** a;

}

public static int poisciCrkoBinarno(char iskano, char[] tab){

int zac = 0, kon = tab.length-1, sre,a=0, ret=-1;

**do** {

sre = (zac+kon)/2;

**if** (tab[sre] == iskano){

ret=sre;

**break**;

}

**else** **if** (tab[sre]<iskano){

zac = sre+1;

}

**else** kon = sre-1;

a++;

}**while**(zac<=kon);

System.out.println("primerjali smo "+a+"-krat");

**return** ret;

}

public static int poisciCrkoLinearno(char crka, char[] tab){

int a =0, ret=-1;

**for** (int i =0; i<tab.length; i++){

a++;

**if** (tab[i] == crka){

ret= i;

**break**;

}

}

System.out.println("primerjali smo "+a+"-krat");

**return** ret;

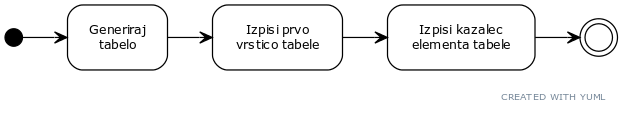
}

}

Opomba pred naslednjimi nalogami: *če vam izračun pozicij označevalnikov v izrisu dela težave, lahko predpostavite, da se vsa števila vedno izpisujejo v polju dolžine 5, ter da je označevalnik vedno na sredini polja.*

**Naloga 5**

Dan javanski program izvaja sekvenco aktivnosti, kot je prikazano na diagramu:

Dopolnite dano kodo z realizacijo ne-implementiranih metod. Pri tem predpostavite:

public class Vizualizacija00 {

static int dolzinaTabele=12;

public static void main(String[] args){

int[] tab=generirajTabelo(dolzinaTabele);

izpisiPrvoVrsticoTabele(tab);

izpisiKazelecElemTabele(tab,6);

}

}

* tabela lahko vsebuje zgolj do 5 mestna pseudo-naključna pozitivna števila z vključeno vrednostjo 0;
* izpis prve vrstice tabele izpiše števila v eni sami vrstici, pri tem se med posameznimi števili izpiše en sam presledek
* izpis kazalca se izvrši pod izpisom tabele na sredini pod število, na katerega kaže. Npr.

A number on a white background

Description automatically generatedkaže na element z zaporedno številko (indeksom) 6, kot je dano s primerom delne realizacije programa, danega v nadaljevanju:

**Naloga 6**

Med obe izpisni vrstici v razredu Vizualizacija00 vrinite naslednje zaporedje stavkov:

try{

Thread.sleep(2000);

} catch(InterruptedException e){}

Obrazložite, kaj se je med izvajanjem zgodilo. Povečajte še vrednost argumenta metode sleep na npr. 7000 in opazujte razliko.

**Naloga 7**

Z zaporednimi klici izpisovalnih metod iz glavne metode razreda Vizualizacija00 je mogoče izvesti vizualizacijo postopka zaporednega iskanja, korak po korak. Izvedite jo. Pri tem podajte primer za iskanja za obstoječo vrednost in za neobstoječo vrednost. Naj bo zakasnitev med posameznimi koraki izvajanja vsaj 1.5s .

**Naloga 8**

Na podoben način, kot je izvedena animacija vizualizacije postopka v nalogi 4, izvedite tudi za binarno iskanje. Za to:

* Dodajte metodo generirajUrejenoTabelo/1
* Dodajte kot prvi klic izpisnih metod še klic metode izpisiPredPrvoVrsticoTabele(tab,spod,zgor); , ki bo izrisal meje iskalnega zaporedja v postopku binarnega iskanja v vrstici nad vsebino elementov tabele. Npr.:

A number with a red line

Description automatically generated

Razmislite tudi o spremembi znaka, ko sta obe meji postavljeni na isti element tabele.

**import** java.util.Arrays;

public class Naloga8 {

public static void main(String[] args) {

int[] tabela = generirajUrejenoTab(15);

int stevilo = (int)(Math.random()\*90);

System.out.println("Iscemo stevilo "+stevilo);

poisciBinarno(tabela, 23);

}

public static void izpisiTab(int[] tab){

System.out.println(Arrays.toString(tab).replace("[", "").replace(",","").replace("]",""));

}

public static int[] generirajUrejenoTab(int dolzina) {

int[] tabela = **new** int[dolzina];

tabela[0] = (int) (Math.random() \* 10);

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

tabela[i] = tabela[i - 1] + (int) (Math.random() \* 10) + 1;

}

**return** tabela;

}

public static void izpisiPredPrvoVrsticoTabele(int spod, int zgor, int dolzina) {

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

**if** (i == spod && spod == zgor) {

System.out.print(" v ");

} **else** **if** (i == spod) {

System.out.print(" < ");

} **else** **if** (i == zgor) {

System.out.print(" > ");

} **else** {

System.out.print(" ");

}

}

System.out.println();

}

public static void poisciBinarno(int[] tabela, int iskano) {

int zac = 0, kon = tabela.length - 1, sred, p = 0;

**while** (zac <= kon) {

izpisiPredPrvoVrsticoTabele(zac, kon, tabela.length);

izpisiTab(tabela);

sred = (zac +kon)/2;

p++;

System.out.println("Sredinski element (" + sred + "): " + tabela[sred]);

**if** (tabela[sred] == iskano) {

System.out.println("Najdeno na indeksu "+sred);

System.out.println("Primerjave: "+p);

**return**;

} **else** **if** (tabela[sred] < iskano) {

zac = sred + 1;

System.out.println("Iskanje v zgornji polovici.");

} **else** {

kon = sred - 1;

System.out.println("Iskanje v spodnji polovici.");

}

**try** {

Thread.sleep(1500);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

System.out.println("Število ni v tabeli.");

System.out.println("Število primerjav: " + p);

}

}