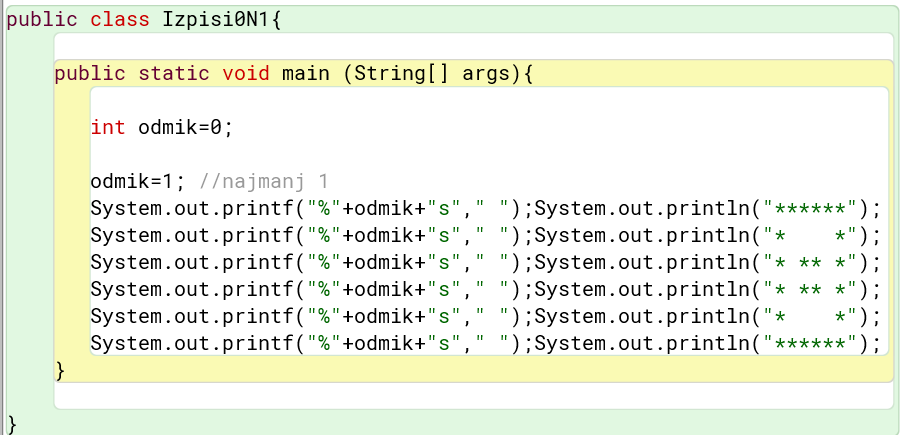
|  |
| --- |
| **Teme**   * Razredni člani * Razredne metode   + Sintaksa, struktura, argumenti :   + Identifikacija, ponovljivost, splošnost   Opomba:  *---.* |

**Naloga 1**

Dan je Javanski program z naslednjo vsebino:



1. Zaženite ga in ugotovite kaj program v dani obliki izpiše
2. Vrednost spremenljivke odmik nastavite na vrednost 6 in opredelite (zabeležite) spremembo v rezultatu izvajanja in s tem vlogo spremenljivke odmik v danem programu.
3. Z replikacijo dane kode ('copy&paste') izvedite 3 vertikalne izpise originalnega lika. Med seboj naj bodo ločeni s po eno prazno vrstico, vsi naj bodo izrisani na enakem (istem) odmiku od levega roba zaslona (npr. 20).

**Naloga 2**

V primeru, da ste sledili navodilom prve nalog, je del c) rezultiral v sekvenci stavkov, kjer se po osem (recimo) stavkov 3 x ponovi. Torej: Izpisi0N1 pretvorite v Izpisi0N2 tako, da replikacij kode ne bo več. Iz ponovitve ustvarite razredno metodo, program naj nato to metodo 3 krat kliče kot je podano na pridanem delčku kode:

public class Izpisi0N2 {

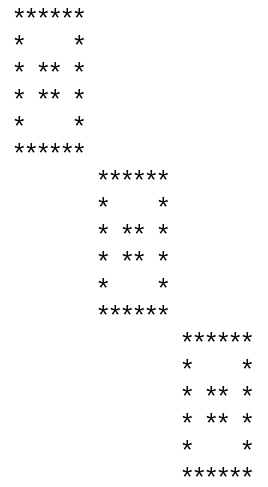
public static void main(String[] args){

izpisiK6();

izpisiK6();

izpisiK6();

}

**Naloga 3**

Metoda izpisiK6/0 kot je opredeljena v predhodni nalogi izpisuje zahtevano obliko na odmiku (recimo 6) od levega roba zaslona. Dodajte razredu še dve metodi: izpisiK13/0 in izpisiK20/0. Kot je iz zapisa razvidno, sta obe metodi brez argumentov. Če izpisiK6/0 izriše obliko na odmiku 6 od levega roba, naj K13 pomeni odmik 13 od levega roba in K20 odmik 20. S končnim programom morate doseči nekaj takega, kot je na sliki z levo od tega besedila.

**Naloga 4**

Z metodami izpisiK6, izpisiK13, izpisiK20 rešujemo 3 različne probleme (postavljanja na različne lokacije). Metode so v principu pa so enake, vsebujejo enako kodo. Razlikujejo se zgolj po vrednosti podatka, ki določa odmik. To dejansko pomeni, da so metode zelo ozko specializirane, hkrati pa tudi to, da jim lahko posplošimo (če jim naredimo neodvisne od podatka). Spišite metodo izpisiKxy/1 (en argument) z identifikacijo(prototipom):

static void izpisiKxy(int odmik);

vrednost odmika boste prenašali v metodo s klicem; torek deklaracija in inicializacija enako poimenovane spremenljivke v telesu metode ni več potrebna in je niti ne sme več biti.

public class Izpisi0N4 {

public static void main(String[] args){

izpisiKxy(6);

izpisiKxy(13);

izpisiKxy(20);

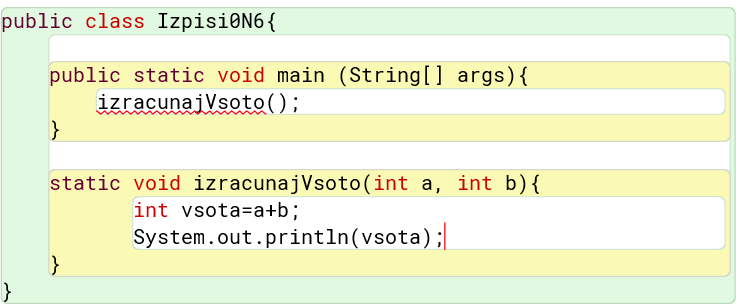
}

Namesto klicev treh različnim metod v izvedi programa Izpisi0N4 izvedite 3 zaporedne klice iste metode z različno vrednostjo argumentov kot

**Naloga 5**

Po logiki izpisovanja oblike iz predhodnih nalog bi na zaslon želeli izrisati 35 likov, drugega nad drugim in sicer tako, da se 'natisnejo' na naključnih odmikih od levega roba zaslona. Ta naključnost naj bo v obsegu od 6 do 90. V ta namen spišite program Izpisi0N5 z vsemi potrebnimi metodami (hm, dvema?).

**Naloga 6**

Dan je program in kot je podan, vsebuje napako. Programa ni mogoče prevesti. Iz obvestila o napaki v postopku prevajanja ugotovite, v čem je problem, in nato problem odpravite. Rezultat izvedbe te naloge naj bo koda popravljena tako, da se bo dalo metodo izracunajVsoto/2 v nespremenjeni obliki izvesti, in pa elementi iz obvestila o napaki, ki omogočajo identifikacijo napake.

**Naloga 7**

Ugotovite, poročilo pa naj vsebuje neovržljiv dokaz o vaši zaključku vaše ugotovitve:

1. Ali je metodo izracunajVsoto v dani obliki mogoče klicati z manj kot dvema argumentoma ali več kot dvemi argumenti. Poročilo naj vsebuje dokaz za vašo ugotovitev.
2. Ali lahko metodo kličemo z argumentoma vrste float, double, long ?
3. Katere vrste oz. kakšne vrste pa so lahko argumenti z katerimi kličete dano metodo ?

**Naloga 8**

Metodo izracunajVsoto(2 iz naloge 6 lahko kličemo kot izracunajVsoto(1,2) ali izracunajVsoto('1','2') ali celo izracunajVsoto(1,'2'). Kakšne vrednosti dobivate kot posledico posameznih klicev in kakšno vlogo igra pri izračunu vsote številka/vrednost 48? (mogoče je pri tem vredno pogledati mesto znaka 0 v (recimo) ascii tabeli znakov).

**Naloga 9**

Recimo, da v originalnem programu naloge 6 (Izpisi0N6), kličete metode izračunaj vsoto kot izracunajVsoto('2','8'). Rezultat, ki ga dobite je logično za 2\*48 prevelik. Zato:

Iz programa naredite program Izpisi0N9 in v razred dodajte metodo, ki bo izvedla korekcijo izračuna. Preverite delovanje in beležite odziv. Pomnite, da imate v programu še vedno obe metodi in da se je izbira izvršila avtomatično. Preverite še klice oblik ('2',4) in (2,'4') ….. ( zelo 'ambigious' ! )

static void izracunajVsoto(char a, char b){

int vsota=a+b-2\*48;

System.out.println(vsota);

}

**Naloga 10**

Naš seštevalnik celih števil iz predhodnih nalog je omejen na sesštevanja dveh argumentov. Vemo pa, da v aritmetiki velja tudi to da če operandu nič ne prišteješ, se njegova vrednost ne spremeni, prav tako, če operanda niti nimate. In da, seštejemo lahko tudi 3 operande, pa štiri, pa pet in šest njih. Recimo, da nam šest zadošča. Spišite torej mehanizem izracunajVsoto, ki bo to omogočal ! Razred poimenujte Izpisi0N10.