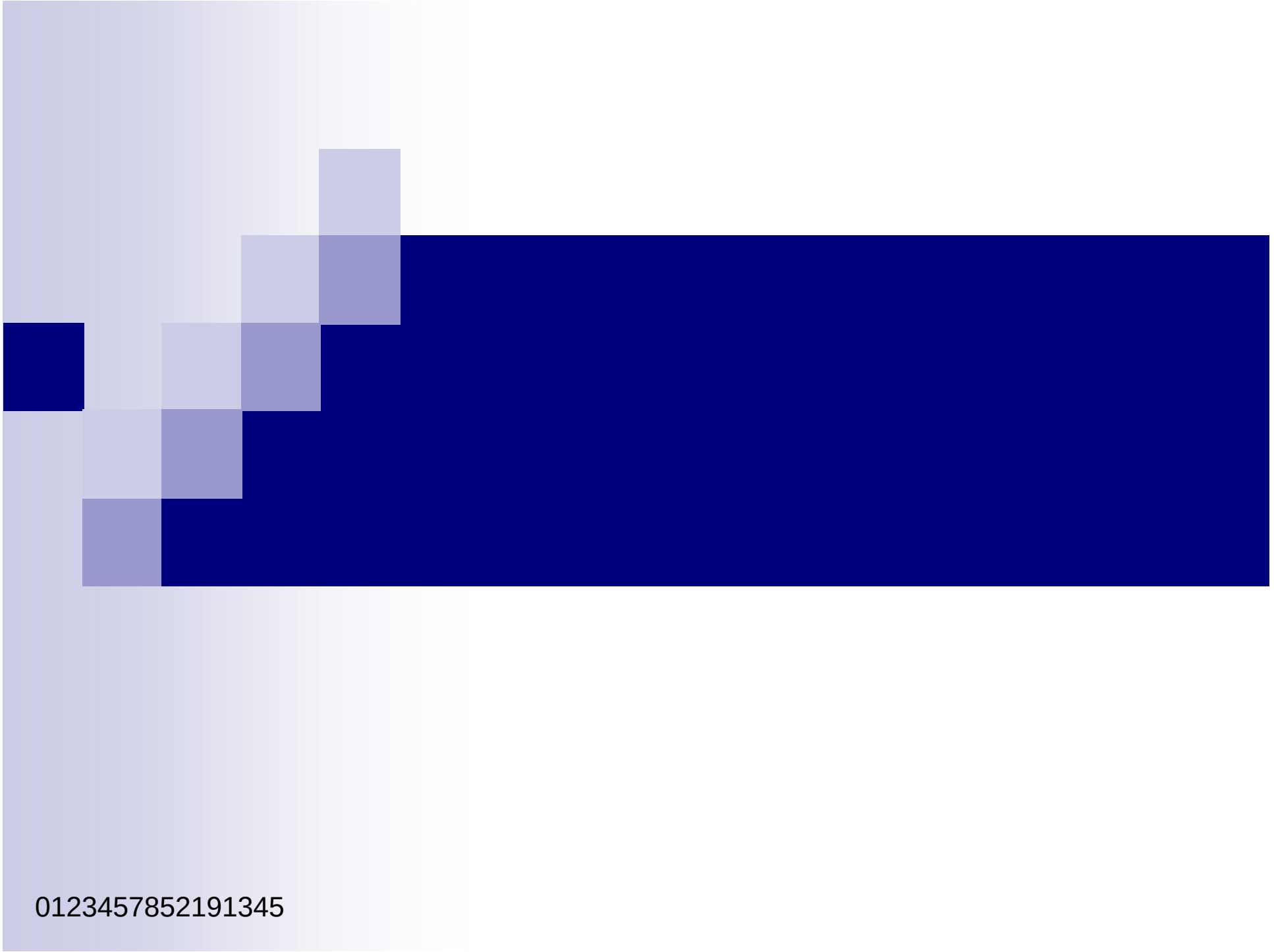
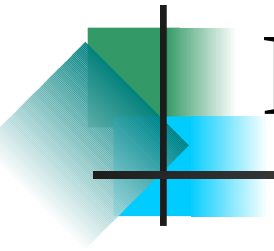

0123457852191345

Vsebina

- Pogled naprej od zadaj
  - Kaj smo želeli narediti
- Kje smo sedaj
  - Kaj smo naredili
- Pogled naprej
  - Kaj lahko še naredimo
- Domača naloga

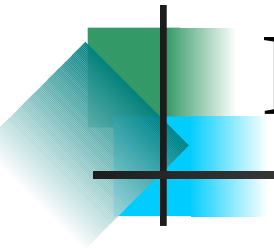


Ponovitev

- AWT (Abstract Window Toolkit)
  - Omogoča izdelavo okenskih programov
  - je standardna zbirka klicev (API) za izdelavo grafičnega uporabniškega vmesnika (GUI)

Ponovitev

- Dogodkovno gnano programiranje
  - Ob interakciji uporabnika z vmesnikom, se komponentam spremeni notranje stanje
  - Na komponente dodamo rokovalnik dogodkov
    - Sproži se dogodek, ki ga rokovalnik prestreže in ustrezno reagira



Ponovitev

- LayoutManager (ravnatelj izgleda)
  - Pomoga pri postavljanju komponent na vsebovalnik (container)
  - Primeri:
    - BorderLayout
    - GridLayout
    - FlowLayout
    - GridbagLayout

Komponenta: TextField

- TextField je enovrstično vnosno polje
- Tvoritelji:
  - `TextField()`
  - `TextField(int st_znakov)`
  - `TextField(String besedilo)`
  - `TextField(String besedilo, int st_znakov)`
- Metode:
  - `String getText()`
  - `Void setText(String besedilo)`
  - `...`

Komponenta: Menu

- Orodna vrstica menujev, menuji
  - MenuBar, Menu, MenuItem
- Tvoritelji:
  - `MenuBar()`
  - `Menu(), Menu(String besedilo)`
  - `MenuItem(), MenuItem(String imeElementa)`
- Skupaj jih zložimo z `add()`

Vsebina

- Polje
  - ponovitev
- Več-dimenzionalna polja
  - osnove
  - matrike
  - uporaba
- Primer
- Domača naloga

Polje – ponovitev

- Polje je skupina spremenljivk enakega tipa
- Do posameznega elementa se dostopa preko njegovega indeksa
- Deklaracija (2 koraka):
  - *tip ime_spremenljivke*[];
  - *ime_spremenljivke* = new *tip*[*velikost*];

  ali krajše (1 korak):
  - *tip ime_spremenljivke*[] = new *tip*[*velikost*];

Polje – primer

```
public static void main(String args[]) {
    String dan_v_tednu[] = new String[7];
    dan_v_tednu[0] = „Ponedeljek“;
    dan_v_tednu[1] = „Torek“;
    dan_v_tednu[2] = „Sreda“;
    dan_v_tednu[3] = „Cetrtek“;
    dan_v_tednu[4] = „Petek“;
    dan_v_tednu[5] = „Sobota“;
    dan_v_tednu[6] = „Nedelja“;
    System.out.println(„Drugi dan tedna: “ +
       dan_v_tednu[1]);
}
```

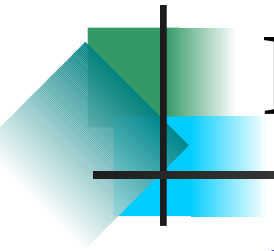
Matrike

- Primer:
  - int matrika[][] = new int[4][4];
    0 0 0 0
    0 0 0 0
    0 0 0 0
    0 0 0 0
  - matrika[1][2] = 1;
    0 0 0 0
    0 0 1 0
    0 0 0 0
    0 0 0 0

Matrike

- Po matrikah se ponavadi sprehajamo z dvojno *for* zanko
- Prvi *for* gre po vrsticah, drugi po elementih v posamezni vrstici

```
for(int i=0; i<4; i++){
	for(int j=0; j<4; j++){
		System.out.print(matrika[i][j] + „ “);
	}
	System.out.println();
}
```

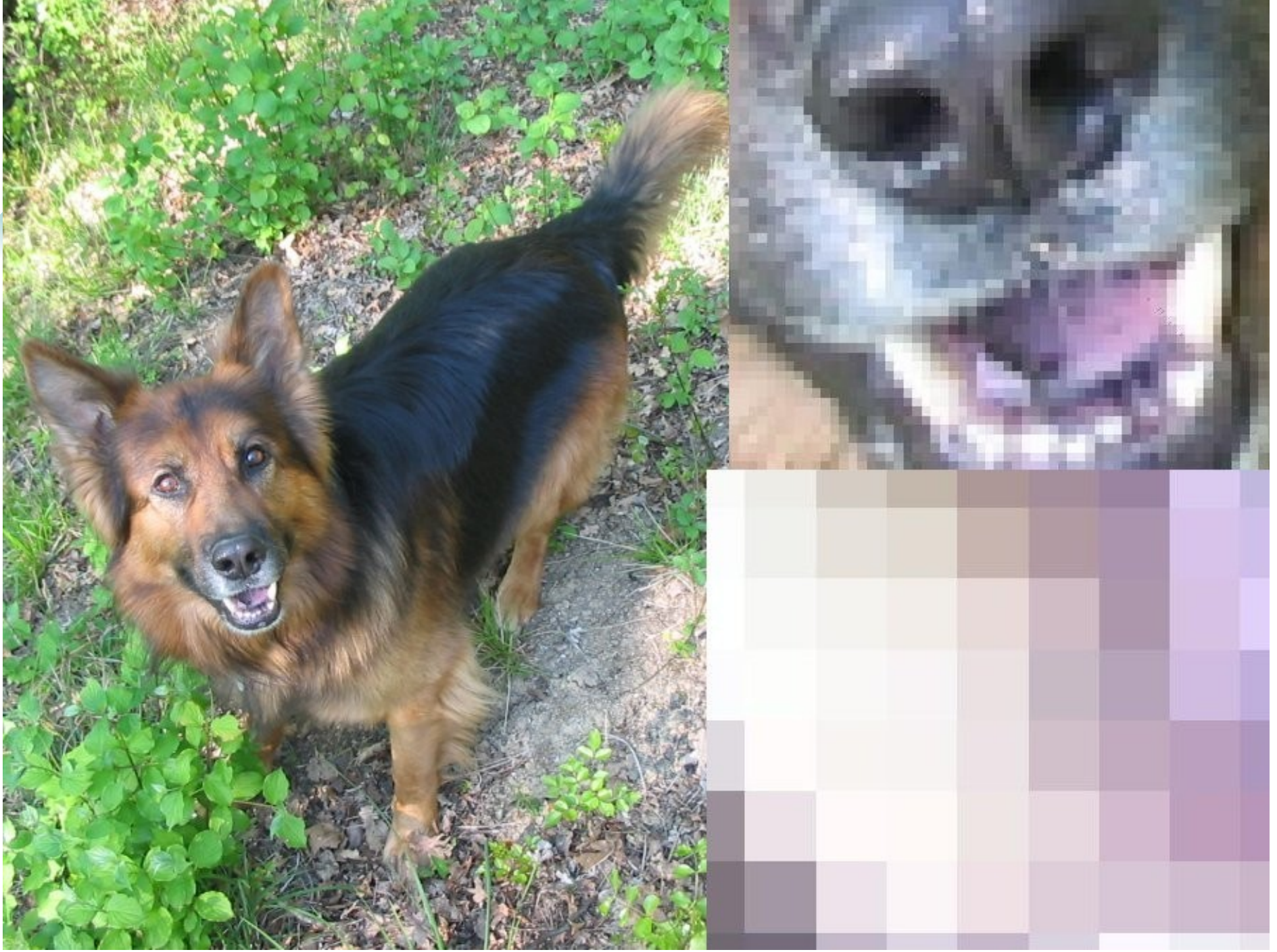


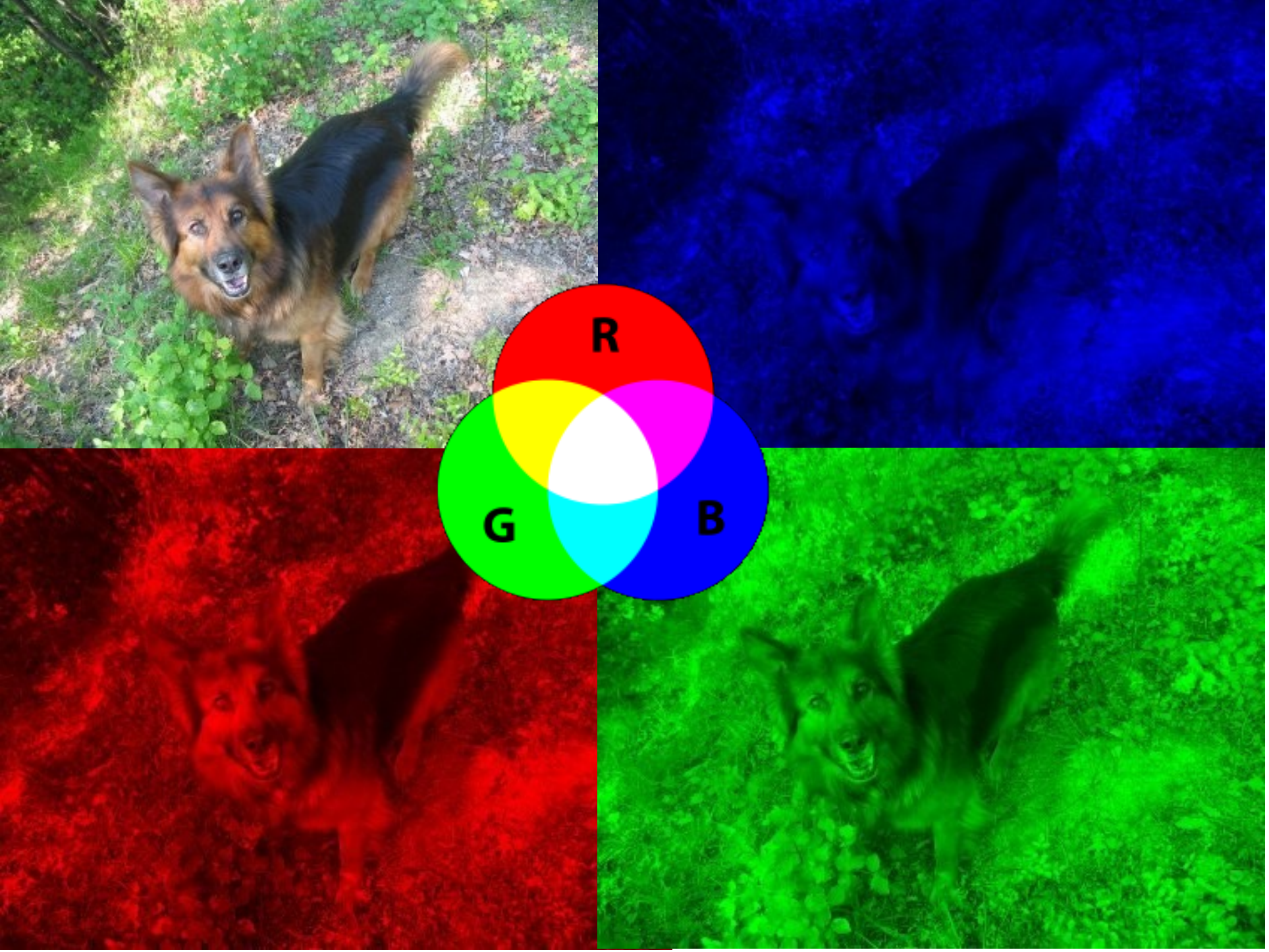
Primer

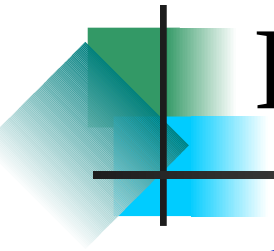
- oglejmo si primer Matrika.java

Uporaba

- 2D polja se lahko uporablja pri delu s slikami
  - vrednost vsake točke v sliki se vpiše na svojo lokacijo v (svojo) matriko
  - nad matriko se izvede določene operacije
  - vrednosti v matriki se nato prepiše nazaj v sliko
- Slika v RGB obliki je predstavljena kot tri dvo-dimenzionalna polja




R
G
B

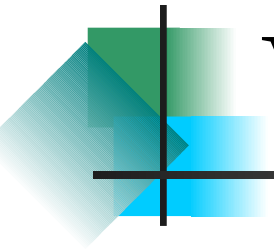


Primer

- Omejili se bomo na ČB slike
  - ČB slike lahko spravimo v 1 matriko. Elementi imajo vrednosti od 0 do 255
    - 0 črna, 255 bela
- Oglejmo si primer Slika2D.java

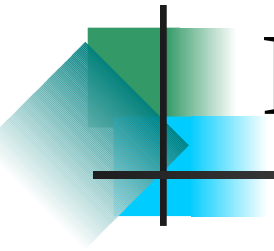
Več-dimenzionalno polje

- Lahko si ga predstavljamo kot polje polj
  - ali kot matriko
- Deklaracija:
  - *tip ime_spremenljivke*[][] = new *tip*[*velikost*][];
- Primer:
  - int matrika[][] = new int[4][4];
- Lahko ima več kot dve dimenziji:
  - int kocka[][][] = new int[5][5][5];



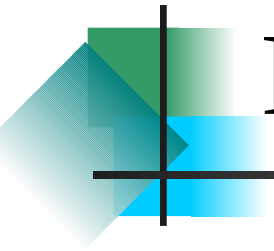
Vsebina

- Ponovitev
  - razredi, dostop, dedovanje
- Abstrakcije
  - primer
- Abstrakcije - slika
  - primer
- Domača naloga



Ponovitev

- Razred:
  - je „predloga“ predmeta
  - sestavljen iz:
    - spremenljivk
    - metod



Ponovitev

- Primer razreda
  - Stol:
    - spremenljivke: št. nog, material, naslonjalo, ...
    - metode: tvoritelj (konstruktor), povej št. nog, material, ali ima naslonjalo, ...

Abstrakcije

- Primer razreda

```
class Stol {
        int st_nog;
        String material;
        boolean naslonjalo;
        ...

        getSt_nog(){
                return st_nog;
        }
        ...
}
```

Abstrakcije

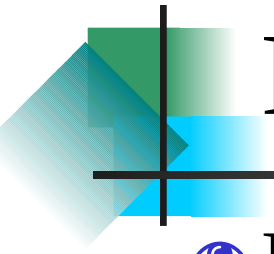
- Primer razreda

```
class Stol {
        int st_nog;
        String material;
        boolean naslonjalo;
        ...

        getSt_nog(){
                return st_nog;
        }
        ...
}
```

Ponovitev

- Predmet (objekt) je realizacija razreda
  - predmet naredimo z ukazom `new`

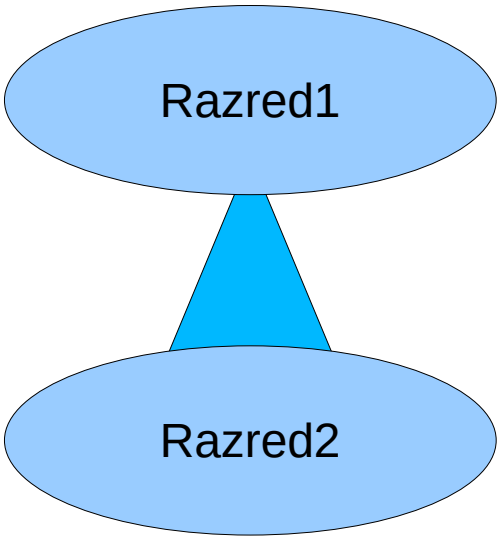


Ponovitev

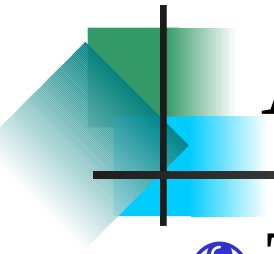
- Dostop razredov oz. metod:
  - public
  - private
  - protected

Ponovitev

- Dedovanje razredov
  - razred ima lahko podrazrede in nadrazrede



  - podrazred od nadrazreda podeduje metode in spremenljivke

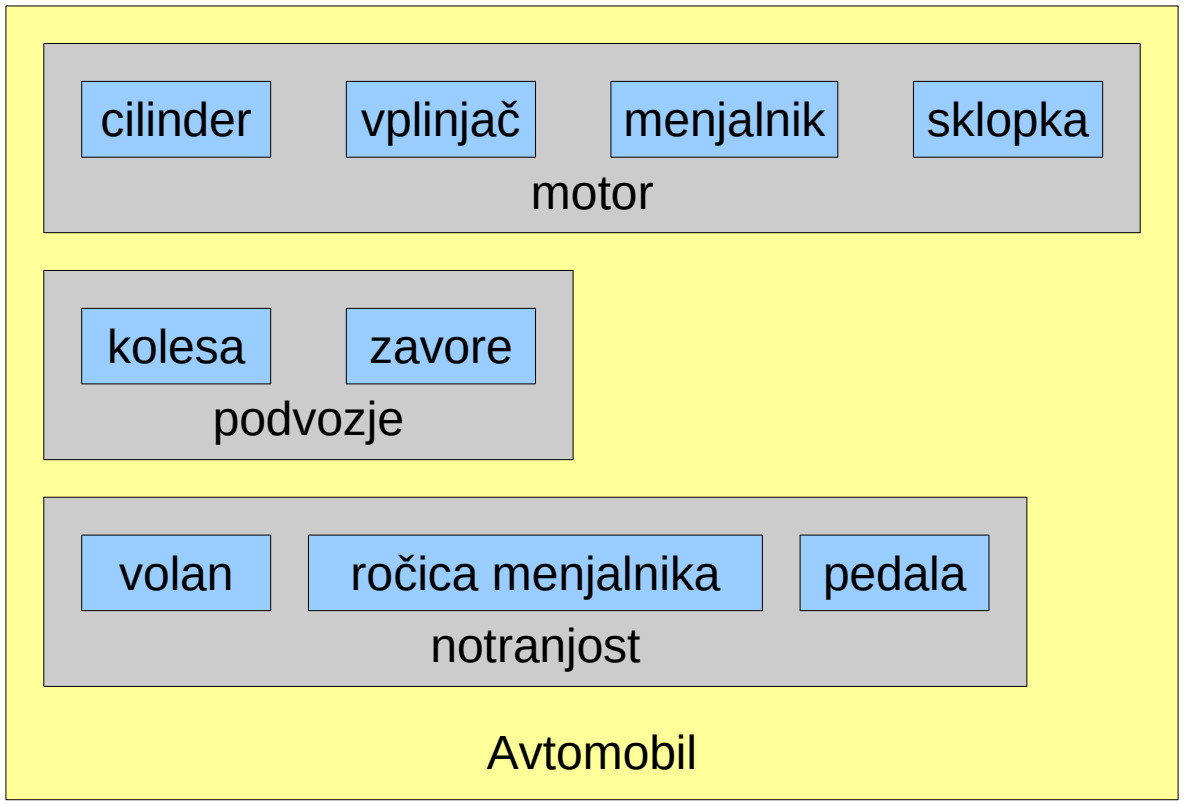


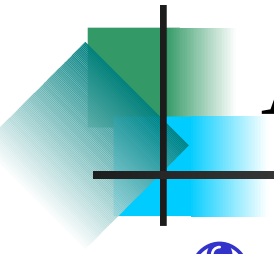
Abstrakcije

- Z abstrakcijo skrijemo kompleksnost
  - v javi lahko z uporabo razredov sestavimo posamezne sestavne dele in jih (hierarhično) povežemo v celoto

Abstrakcije

- Z abstrakcijo skrijemo kompleksnost
  - primer avtomobila:





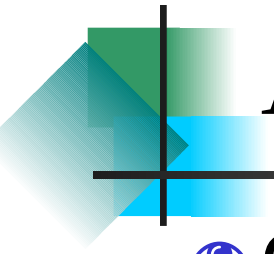
Abstrakcije

- Abstraktne metode in razrede
  - v Javi lahko definiramo abstraktno metodo:
    - zapišemo samo podpis metode
    - metodo mora nato definirati podrazred
  - če razred vsebuje abstraktno metodo, je tudi sam abstrakten

Abstrakcije - slika

- Program iz 7. vaje (Slika2D.java) bomo razširili tako, da bo znal delati z barvnimi slikami
  - pomnimo: RGB barvne slike so sestavljene iz treh barv (rdeče, zelena, modra). Vsaka barvo predstavimo s svojo matriko.
    - vrednost 255 pomeni največjo jakost barve, 0 pomeni da barve ni.
- Uporabili bomo abstrakcije in dedovanje!

Abstrakcije - slika



- Oglejmo si primer Slika2D.java (in vse kar sodi zraven)

Kako naprej?

- Program lahko služi, kot odskočna deska za hobby programiranje v prostem času
  - dodate poljubne efekte
  - dodate poljubno funkcionalnost
  - ....