

# Klasės ir Objektai Javoje

Mindaugas Karpinskas  
2017

# U0 - pakartoti...

---

Programa turi paklausti, kokį veiksmą vartotojas nori atlikti:

- A - sugeneruoti atsitiktinį skaičių 1-100
- B - sugeneruoti 10 atsitiktinių skaičių

Programa išspręskite trimis būdais:

1. Naudojant tik **while**
2. Naudojant itk **do-while**
3. Naudojant ik **for**.

# Turinys

---

Java klasės ir objektai

Objektinio programavimo paradigma Javoje

Java klasių savybės

enum tipas

## Įvadas (remember and repeat)

---

- Java yra objektinio programavimo kalba. Todėl jos visos programos kodas yra rašomas klasėse
- Visos Java programos apibrėžia klasės “būseną” ir “elgesį”, būseną apibrėžiama klasės kintamaisiais, o elgesys – jos metodais.
- Klasės tipo egzemplioriai (objektai) realizuoja įvairias tos pačios klasės būsenas ir individualią elgseną.

## Išvadas (remember and repeat)

---

- Objektai naudoja metodus bendraujant tarpusavyje.
- Klasė apjungia savyje grupę kintamųjų.
- Klasių metodai apibrėžia objektų kintamųjų kitimo taisykles.
- Išskirtinė grupė metodų, vadinamų klasių **konstruktoriais**, apibrėžia klasės kintamuosius (“būseną”) naujo klasės egzemplioriaus kūrimo metu.



Svarbu

# Įvadas (remember and repeat)

---

- Bet kuri klasė apibrėžia naują tipą *analogišką* ***Integer, String, Random, Scanner, ...***
- Java kalba apibrėžia klasės laukų ir metodų sintaksės taisykles.
- Klasių laukams ir metodams galioja standartiniai objektinio programavimo elementai:
  - duomenų apgauba,
  - paveldimumas,
  - polimorfizmas,
  - perklojimas.

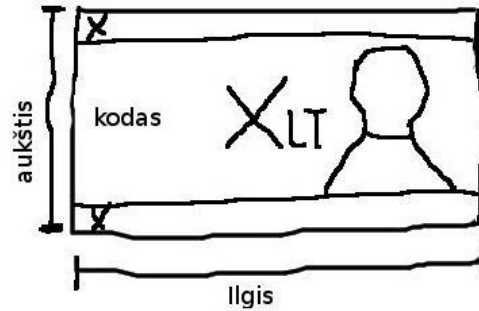
Aptarsim vėliau

# Klasė

Klasė šablonas pagal  
kuria kuriami  
egzemplioriai

Štai klasė, kuri nurodo kokias savybes  
gali turėti kiekvienas gaminamas  
pinigas (aukštis, ilgis, spalva, vertė,  
vaizduojamas žmogus).

Tai yra tarsi šablonas, pagal kurį yra  
kuriami objektai (šiuo atveju - pinigai).

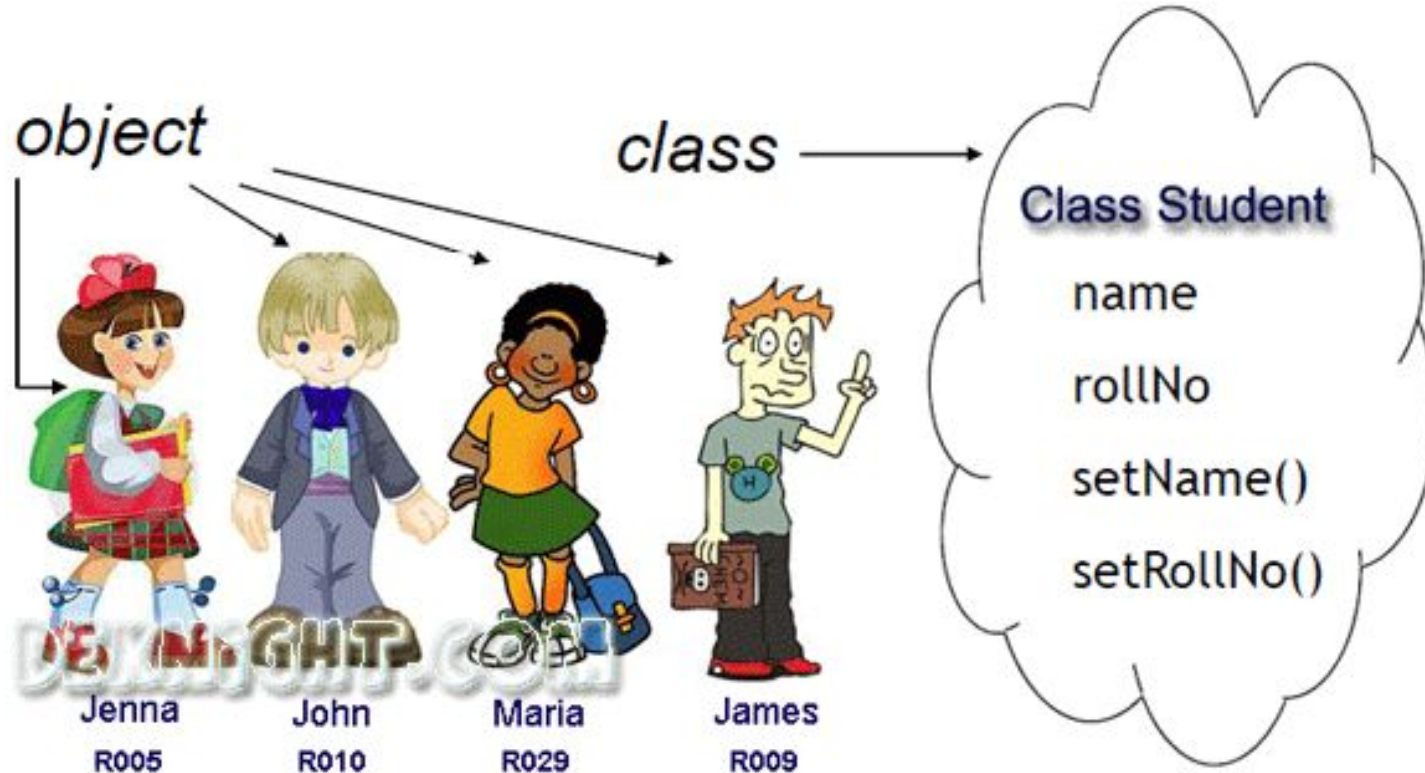


O štai tą klasę atvaizduojantis tikras  
apčiuopiamas objektas (klasės egzempliorius)



# Class vs Object

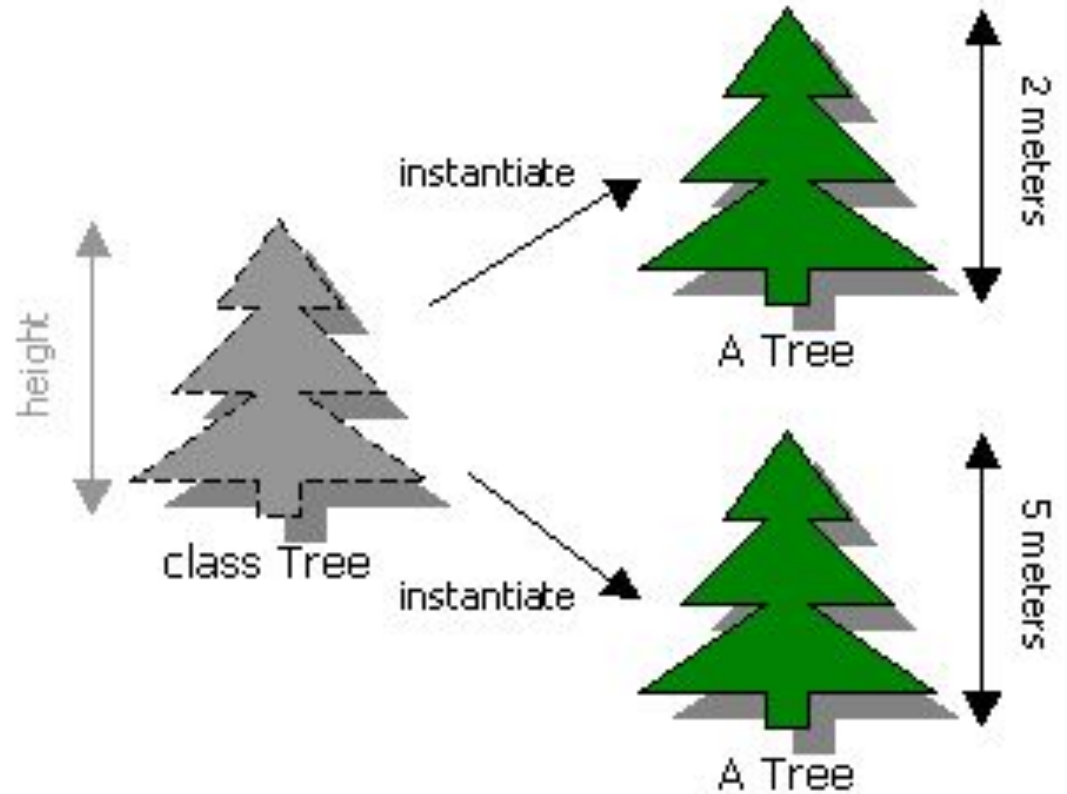
(šaltinis: <http://www.c4learn.com/>)





# Class vs Object

Šaltinis: <http://www.ustudy.in/>



# Klasės

---

- klasė (*class*) yra jos laukų (*fields*) (kintamųjų) ir metodų (*methods*, procedūros arba funkcijos) rinkinys. Metodai operuoja su tos pačios ir kitų klasių laukais

<b>Circle</b>
<b>centre</b> <b>radius</b>
<b>circumference()</b> <b>area()</b>

Klasė

Jos laukai

ir metodai

# Klasė

---

## Klasės apibrėžimo sintaksė:

```
class ClassName [extends SuperClassName] [implements InterfaceName] {  
    [fields declaration]  
    [methods declaration]  
}
```

## Primityvi klasė (be laukų ir metodų)

```
public class Circle {  
    // my circle class  
}
```

Pridedame laukus (kintamuosius): class Circle su laukais

---

Pridedame klasei *laukus*

```
public class Circle {  
    public double x, y; // centre coordinate  
    public double r;    // radius of the circle  
  
}
```

Klasės laukai (kintamieji) kartais vadinami  
*egzemplioriais (arba objekto)* kintamaisiais.

# Constructors

---

To create an instance of a class, all you have to do is write **new** before it. For example:

```
Random r = new Random();
```

IŠ:

Certified Associate Java Study Guide

# PVZ

```
public class ConfigSvcProperty {  
    String component;  
    String context;  
    String key;  
    String value;  
    String type;  
    boolean hasOverrides;  
}
```

```
public static void main(String[] a) {  
    ConfigSvcProperty property = new  
    ConfigSvcProperty();  
  
    property.component = "ABC Komponentas";  
    property.key = "Raktas";  
    property.value = "reikšmė";  
  
    System.out.println("Komp: " + property.component  
        + ", raktas: " + property.value  
        + " reikšmė: " +  
    property.value);  
}
```

# PVZ

---

```
public static void main(String[] a) {  
    ConfigSvcProperty property = new ConfigSvcProperty();  
  
    property.component = "ABC Komponentas";  
    property.key = "raktasX";  
    property.value = "B";  
  
    parodyk(property);  
}  
private static void parodyk(ConfigSvcProperty property) {  
    System.out.println("Komp: " + property.component  
        + ", raktas: " + property.value  
        + " reikšmė: " + property.value);  
}
```

# Uzduotis1: Asmuo

---

- Sukurti naują tipą/klasę - Asmuo su kintamaisiais(vardas, metai)
  - Asmuo (vardas, metai)

Klasės laukai be **static** žodelio!

- main metode sukurti 5 egzempliorius, objektus Asmuo. Kiekvienam iš egzempliorių priskirti kitą vardą ir metus
- Atspausdinti vardus ir metus

*PVZ*

```
public class Circle {  
    public double x, y;  
    public double r;  
}
```



# Uzduotis1

```
public class Asmuo {  
    String vardas;  
    int metai;  
}
```

```
class Uzduotis1 {  
    public static void main(String[] args) {
```

```
        Asmuo a1 = new Asmuo();
```

```
    }
```

# Uzduotis1

```
public class Asmuo {  
    String vardas;  
    int metai;  
}
```

```
class Uzduotis1 {  
    public static void main(String[] args) {  
        Asmuo a1 = new Asmuo();  
        a1.vardas = "Jonas";  
        a1.metai = 20;  
  
        Asmuo a2 = new Asmuo();  
        a2.vardas = "Simonas";  
        a2.metai = 40;  
  
        <...>  
  
        System.out.println("Pirmas vardas " + a1.vardas  
                           + " ir metai " + a1.metai);  
    }  
}
```



5 kintamieji

# Papildome klasę jos metodais

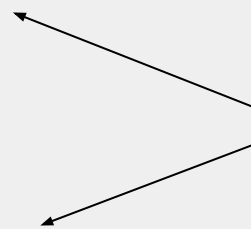
---

- Klasė tik su savo laukais yra „*negyva*“, jų reikšmės gali keisti tik **kitų** klasių metodai, kuriems keblu palaikyti vieningą klasės elgesį
- Apibrėžyi toje pačioje klasėje metodai nurodo vieningą klasės laukų kitimo elgseną. Metodai rašomi po visų klasės laukų apibrėžimo, nors tai nėra privaloma taisyklė.
- Klasės metodų apibrėžimo sintaksė:

```
type MethodName (parameter-list){  
    Method-body;  
}
```

# Papildome klasę Circle jos metoda is

```
public class Circle {  
    public double x, y; // centre of the circle  
    public double r;    // radius of circle  
  
    //Methods to return circumference and area  
    public double circumference() {  
        return 2*3.14*r;  
    }  
    public double area() {  
        return 3.14 * r * r;  
    }  
}
```



Method  
Body

# Uzduotis1a: Asmuo (papildom metodu)

---

- Sukurti naują tipą/klasę - Asmuo su kintamaisiais(vardas, metai)
  - Asmuo (vardas, metai)

Klasės laukai be **static** žodelio!

- main metode sukurti 5 egzempliorius, objektus Asmuo. Kiekvienam iš egzempliorių priskirti kitą vardą ir metus
- Atspausdinti vardus ir metus
- **Klasė turi turėti metodą spausdink(); kuris išvestu į konsole laukų reikšmes**

Klasės metodai be **static** žodelio!

# Kintamųjų tipas

---

- Bet kuri klasė apibrėžia savo vardo naują tipą. Klasė Circle apibrėžia naują Java duomenų tipą vardu Circle
- Circle ir kitos klasės gali naudoti šį duomenų tipą:
  - *Circle aCircle;*
  - *Circle bCircle;*

# Naujus “tipus” galima panaudoti apibrėžiant masyvus

---

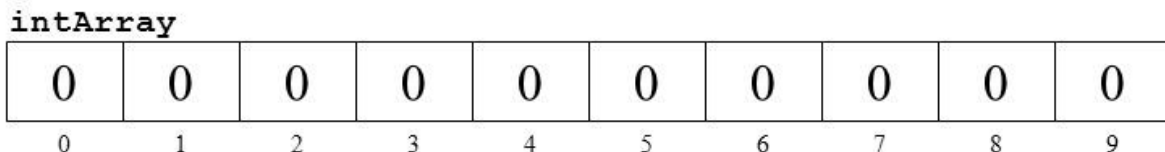
- Bet kuri klasė apibrėžia savo vardo naują tipą. Klasė Circle apibrėžia naują Java duomenų tipą vardu Circle
- Circle ir kitos klasės gali naudoti šį duomenų tipą:
  - *Circle aCircle;*
  - *Circle bCircle;*
- Bet kuri sukurta klasė gali būti panauduota apsibriažiant masyvus
- Jei galime susikurti Klasės kintamąjį, tai galima susikurti ir masyvo kintamąjį panaudodami naują klasę:
  - `Circle[] aCircles;`
  - `Circle[] bCircles;`

# An Example of Array Declaration

- The following declaration creates an array called **intArray** consisting of 10 values of type **int**:

```
int[] intArray = new int[10];
```

- This easiest way to visualize arrays is to think of them as a linear collection of boxes, each of which is marked with its index number. You might therefore diagram the **intArray** variable by drawing something like this:



- Java automatically initializes each element of a newly created array to its **default value**, which is zero for numeric types, **false** for values of type **boolean**, and **null** for objects.

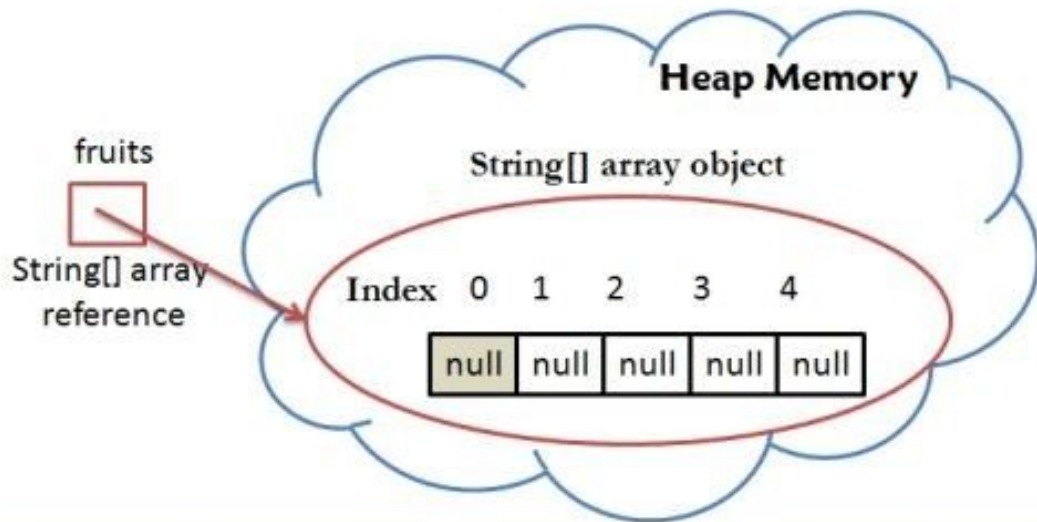


## Declaring and Constructing a String Array

© theopentutorials.com

### Statement:

```
private String[] fruits = new String[5];
```



Once array is declared and constructed, elements gets default value based on their type. This is true regardless of where (local, instance or static) the array itself is declared and constructed.

Default value for String is **null**

# Nuorodos tipo masyvai

```
Circle[] aCircles;
```

```
aCircles = new Circle[5];  
aCircles[1] = new Circle();  
aCircles[2] = new Circle();  
aCircles[3] = new Circle();  
aCircles[4] = new Circle();  
aCircles[5] = new Circle();
```

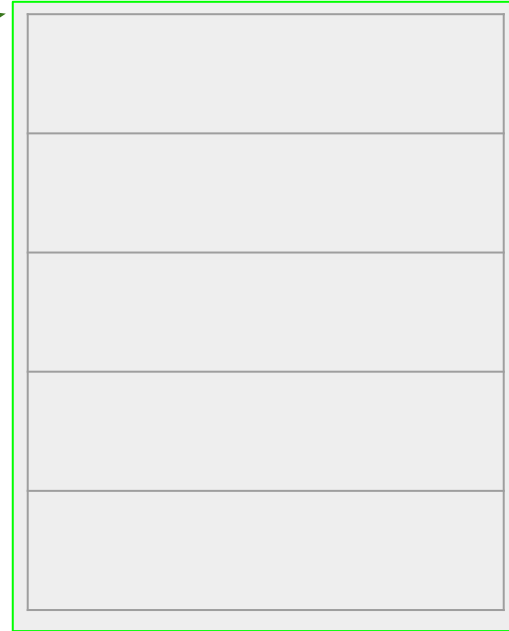


# Nuorodos tipo masyvai

```
Circle[] aCircles;
```

```
aCircles = new Circle[5];
```

```
aCircles[0].x = 7;
```



# Nuorodos tipo masyvai

```
Circle[] aCircles;
```

```
aCircles = new Circle[5];
```

```
aCircles[0].x = 7;
```



# Nuorodos tipo masyvai

```
Circle[] aCircles;
```

```
aCircles = new Circle[5];
```

```
aCircles[0] = new Circle();
```

```
aCircles[1] = new Circle();
```

```
aCircles[2] = new Circle();
```

```
aCircles[3] = new Circle();
```

```
aCircles[4] = new Circle();
```

```
aCircles[0].x = 7;
```

[0] x=7, y=0, z=0

[1] x=0, y=0, z=0

[2] x=0, y=0, z=0

[3] x=0, y=0, z=0

[4] x=0, y=0, z=0

# Uzduotis1b: Asmuo (papildom masyvu)

---

- Sukurti naują tipą/klasę - Asmuo su kintamaisiais(vardas, metai)
  - Asmuo (vardas, metai)

Klasės laukai be **static** žodelio!

- main metode sukurti 5 egzempliorius, objektus Asmuo. Kiekvienam iš egzempliorių priskirti kitą vardą ir metus
- Atspausdinti vardus ir metus
- Klasė turi turėti metodą spausdink(); kuris išvestu į konsolę laukų reikšmes

Klasės metodai be **static** žodelio!

- **Panaudoti Asmenų masyva**

GC

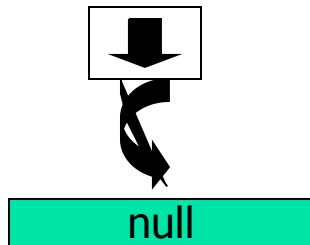
---

# Klasės Circle turinys

---

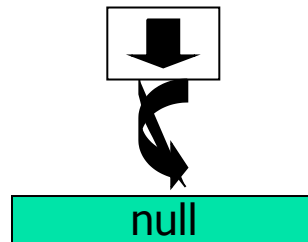
- aCircle, bCircle yra nuorodos į atminties adresą nuo kurio fiziškai yra talpinamas to tipo klasės egzemplioriaus kintamųjų ir metodų turinys. Jei egzempliorius neapibrėžtas, talpinama nuoroda į *null*

aCircle



Abstrakti nuoroda (Null Reference)

bCircle



Abstrakti nuoroda (Null Reference)

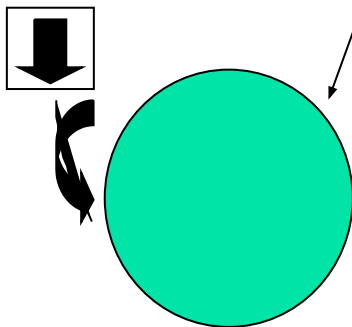


# Klasės objekto (egzemplioriaus) sukūrimas

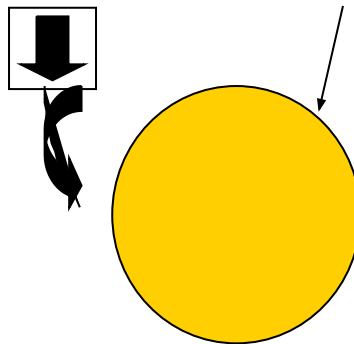
---

- Objektai kuriami dinamiškai (vykdymo metu) naudojant raktinį Java žodį *new*.
- aCircle ir bCircle tampa nuorodomis į Circle objektus

■ `aCircle = new Circle() ;`



`bCircle = new Circle() ;`



# Uzduotis2 Klasės tipo objekto sukūrimas

---

- Sukurti naują Tipą/Klasę Circle su kintamaisiais:
  1. public double x, y; // centre of the circle
  2. public double r; // radius of circle
- Ir su dviem metodais:
  1. public double perimetras() (angl. circumference)
  2. public double plotas() (angl. area)

Panaudojimas PVZ:

```
Circle aCircle = new Circle();
```

```
Circle bCircle = new Circle() ;
```

```
bCircle = aCircle;
```

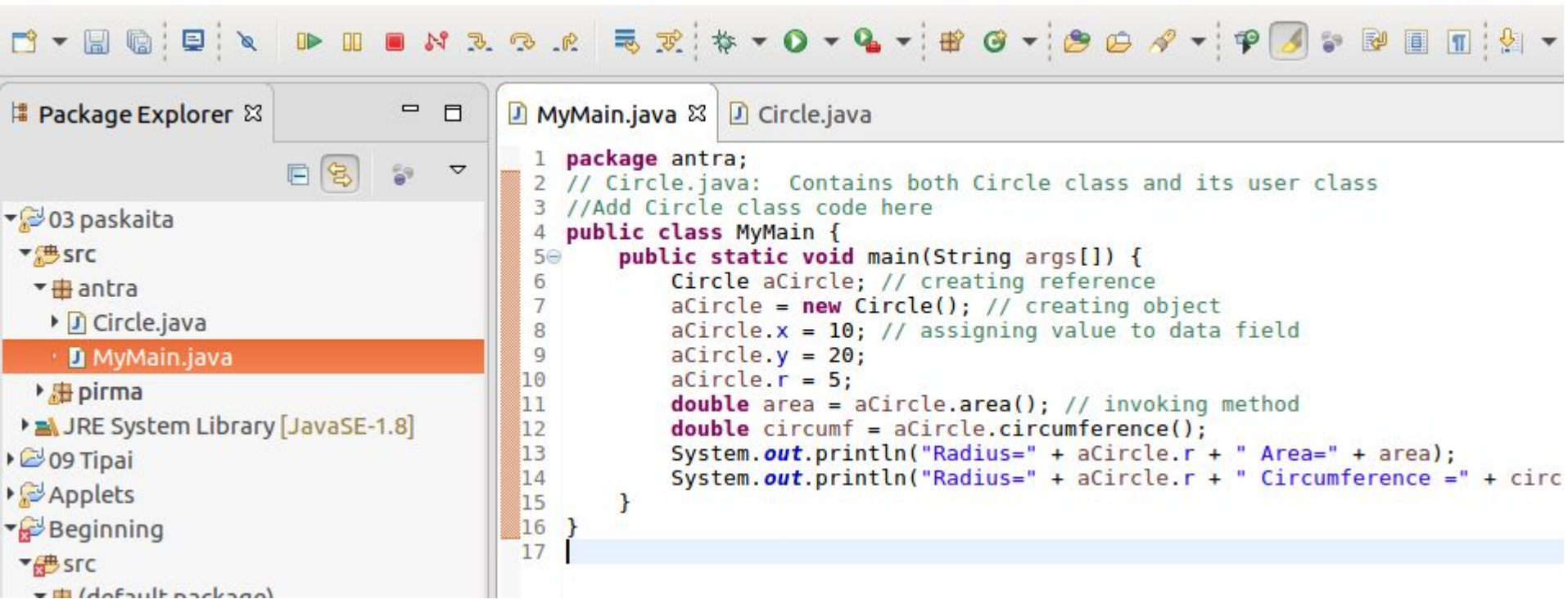
```
bCircle.x = 55.6;
```

```
bCircle.y = 100.6;
```

```
bCircle.r = 1000.0;
```

# Klasēs tipo objekto sukūrimas Uzduotis1

---



# class Circle

---

```
class Circle {  
    public double x, y; // centre of the circle  
    public double r; // radius of circle
```



static?

```
<...>  
    // Methods to return circumference and area  
    public double circumference() {  
        return 2 * 3.14 * r;  
    }  
    public double area() {  
        return 3.14 * r * r;  
    }  
}
```

# Uzduotis2

---

```
public class Uzduotis2 {  
    public static void main(String args[]) {  
        Circle aCircle; // creating reference  
        aCircle = new Circle(); // creating object  
        aCircle.x = 10; // assigning value to data field  
        aCircle.y = 20;  
        aCircle.r = 5;  
        double area = aCircle.area(); // invoking method  
        double circumf = aCircle.circumference();  
        System.out.println("Radius=" + aCircle.r + " Area=" + area);  
        System.out.println("Radius=" + aCircle.r + " Circumference =" + circumf);  
    }  
}
```

# Klasēs tipo objekto sukūrimas

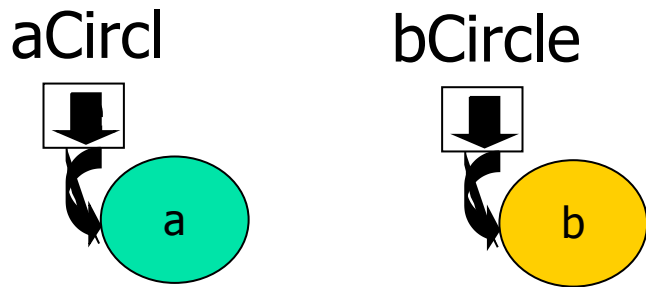
---

```
Circle aCircle = new Circle();
```

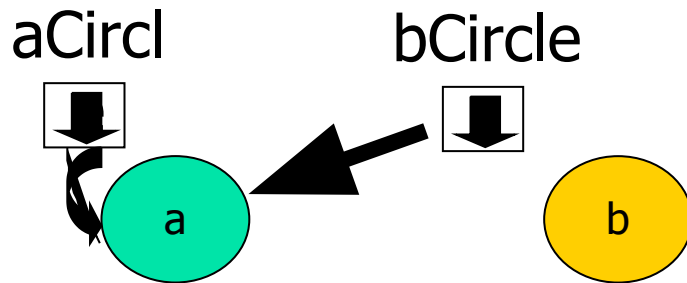
```
Circle bCircle = new Circle() ;
```

```
bCircle = aCircle;
```

## Prieš priskyrimā

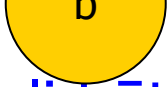


## Po priskyrimo



# Automatinis atminties atlaisvinimas (Automatic garbage collection) GC

---

- Tarkime šis  objektas neturi į save nuorodų ir todėl ateityje negali būti panaudotas.
- Tokie objektai tampa kandidatais automatiniam naudojamam atminties atlaisvinimui (**garbage collection**).
- Java periodiškai automatiškai suranda tokius objektus ir atlaisvina atmintį, kurią užima objektai be nuorodų.

# Prieiga prie objekto Circle kintamųjų

---

*\*\*\*Prieiga panaši į C kalboje naudojamų struktūrų sintaksę.*

*ObjectName.VariableName*

*ObjectName.MethodName(parameter-list)*

```
Circle aCircle = new Circle();
```

```
aCircle.x = 2.0 // initialize center and radius
```

```
aCircle.y = 2.0
```

```
aCircle.r = 1.0
```



# Objekto Circle metodų iškvietimas

---

Objekto metodų panaudojimas:

kreipiamės į aCircle metodą

```
Circle aCircle = new Circle();
```

```
double area;
```

```
aCircle.r = 1.0;
```

```
area = aCircle.area();
```

A black arrow originates from the `area()` part of the `area = aCircle.area();` line in the code block below and points diagonally upwards and to the right towards the text "kreipiamės į aCircle metodą" in the box above.

# Klasės Circle panaudojimas

---

*// Circle.java: Contains both Circle class and its user class*

*//Add Circle class code here*

```
class MyMain {  
    public static void main(String args[]) {  
        Circle aCircle; // creating reference  
        aCircle = new Circle(); // creating object  
        aCircle.x = 10; // assigning value to data field  
        aCircle.y = 20;  
        aCircle.r = 5;  
        double area = aCircle.area(); // invoking method  
        double circumf = aCircle.circumference();  
        System.out.println("Radius="+aCircle.r+" Area="+area);  
        System.out.println("Radius="+aCircle.r+" Circumference =" +circumf);  
    }  
}
```

# Rezultatas

```
// Circle.java: Contains both Circle class and its user class
//Add Circle class code here
class MyMain
{
    public static void main(String args[])
    {
        Circle aCircle; // creating reference
        aCircle = new Circle(); // creating object
        aCircle.x = 10; // assigning value to data field
        aCircle.y = 20;
        aCircle.r = 5;
        double area = aCircle.area(); // invoking method
        double circumf = aCircle.circumference();
        System.out.println("Radius="+aCircle.r+" Area="+area);
        System.out.println("Radius="+aCircle.r+" Circumference =" +circumf);
    }
}
```

```
mindaugas@024-PC: ~/prg/ghx/main/all/adaptor-framework/src/main/java
mindaugas@024-PC:~/prg/ghx/main/all/adaptor-framework/src/main/java$ javac MyMain.java
mindaugas@024-PC:~/prg/ghx/main/all/adaptor-framework/src/main/java$ java MyMain
Radius=5.0 Area=78.5
Radius=5.0 Circumference =31.400000000000002
mindaugas@024-PC:~/prg/ghx/main/all/adaptor-framework/src/main/java$
```

## Uzduotis3:

---

Sukurti klasę Prisiminimas, kurioje turime išsaugoti (private String **informacija**;) kintamajame pasinaudoje metodu:

**issaugok**(String **manoUzrasai**);      *(Klasė prisiminimas turi metodą issaugok)*

Klasė turi turėti metodą: **void** spausdink() - kur išspausdina išsaugotą informaciją

# Prisimename2

```
package lt.codeacademy.sdudy;
public class Prisimename2 {
    public static void main(String[] args) {
        String manoUzasai = "Aš studentas.";
        Prisiminimas priminimas = new Prisiminimas();
        priminimas.issaugok(manoUzasai);
        manoUzasai = "Aš dėstytojas.";
        priminimas.spausdink();
        priminimas.issaugok(manoUzasai);
        priminimas.spausdink(); }}
```

```
// sukuriamas kintamasis
// sukuriamas kintamasis 2
// iškviečiamas metodas
// priskirama nauja reikšmė kintamajam
// kviečiamas metodas
// iškviečiamas metodas su nauja reikšme
// iškviečiamas metodas
```

```
class Prisiminimas {
    private String informacija;
    void issaugok(String naujaInformacija) {
        this.informacija = naujaInformacija;
    }
    void spausdink() {
        System.out.println("Saugoma informacija: "
                           + informacija);
    }
}
```

Saugoma informacija: Aš studentas.  
Saugoma informacija: Aš dėstytojas.

## Uzduotis3a:

---

Sukurti klasę Prisininimas, kurioje turime išsaugoti (String **informacija**;) informaciją pasinaudoję metodu:

issaugok(String **manoUzrasai**);      *(Klasė prisininimas turi metodą issaugok)*

Klasė turi turėti metodą: **void** spausdink() - kur išspausdina išsaugotą informaciją

- **Atsiranda papildomas reikalavimas, kad galima išsaugoti informacija tik tuo atveju jei eilutėje nėra žodžio “bananas”**

# Užduotis4: Adresas

ND!

- 
1. Klasė Adresas visi laukai turi būti **private**
    - a. Šalis
    - b. Miestas
    - c. Gatvė
    - d. Namų numeris
    - e. Buto numeris
  2. Adreso metodas išsaugo informaciją
  3. Išspausdina informaciją *parodykAdresa()*;

# Užduotis: Adresas

---





# Užduotis: Adresas

```
package lt.codeacademy.sdudy;  
public class AdresoUzduotis {  
    public static void main(String[] args) {  
        Adresas manoAdresas = new Adresas();  
        manoAdresas.issaugome("Lietuva", "Vilnius", "Antakalnis",  
            "17a", "5");  
        manoAdresas.parodykAdresa();  
    }  
}
```

Šalis: Lietuva  
Miestas: Vilnius  
Gatvė: Antakalnis  
Namų numeris: 17a  
Buto numeris: 5



```
        butoNumeris);  
    }  
}
```

# Užduotis: Adresas

```
package lt.codeacademy.sdudy;  
public class AdresoUzduotis {  
  public static void main(String[] args) {  
    Adresas manoAdresas = new Adresas();  
    manoAdresas.issaugome("Lietuva", "Vilnius"  
                          , "Antakalnio", "17a");  
    manoAdresas.parodykAdresa();  
  }  
}
```

Šalis: Lietuva  
Miestas: Vilnius  
Gatvė: Antakalnio  
Namo numeris: 17a  
Buto numeris: 5

**class** Adresas {





# enum tipas

---

*// Diena.java: apibrėžia savaitės dienų pavadinimus*

```
public enum Diena {  
    PIRMADIENIS, ANTRADIENIS, TREČIADIENIS, KETVIRTADIENIS,  
    PENKTADIENIS, ŠEŠTADIENIS, SEKMADIENIS  
}
```

*// DienaDemo.java: enum Diena panaudojimo pvz*

```
public class DienaDemo {  
    Diena diena;  
    public DienaDemo(Diena diena) {  
        this.diena = diena;  
    }  
    public void apibudink() {  
        switch (diena) {  
            case PIRMADIENIS:  
                System.out.println("Pirmadienis sunki diena.");  
                break;  
            case PENKTADIENIS:  
                System.out.println("Penktadienis gerai.");  
                break;  
            case ŠEŠTADIENIS:  
            case SEKMADIENIS:  
                System.out.println("Išėiginė puiku.");  
                break;  
            default:  
                System.out.println("Elinė darbo diena.");  
        }  
    }  
}
```

# enum tipas. enum Diena panaudojimo pvz

---

```
public static void main(String[] args) {  
    DienaDemo pirmaDiena = new DienaDemo(Diena.PIRMADIENIS);  
    pirmaDiena.apibudink();  
    DienaDemo treciaDiena = new DienaDemo(Diena.TREČIADIENIS);  
    treciaDiena.apibudink();  
    DienaDemo penktaDiena = new DienaDemo(Diena.PENKTADIENIS);  
    penktaDiena.apibudink();  
    DienaDemo sestaDiena = new DienaDemo(Diena.ŠEŠTADIENIS);  
    sestaDiena.apibudink();  
    DienaDemo septintaDiena = new DienaDemo(Diena.SEKMADIENIS);  
    septintaDiena.apibudink();  
}
```

Pirmadienis sunki  
diena.  
Elinė darbo diena.  
Penktadienis gerai.  
Išsiginė puiku.  
Išsiginė puiku.

```
mindaugas@024-PC: ~/prg/ghx/main/all/adapter-framework/src/main/java
mindaugas@024-PC:~/prg/ghx/main/all/adapter-framework/src/main/java$ javac DienaDemo.java
^[^[A^[[Amindaugas@024-PC:~/prg/ghx/main/all/adapter-framework/src/main/java$ java DienaDemo
Pirmadienis sunki diena.
Elinė darbo diena.
Penktadienis gerai.
Išėiginė puiku.
Išėiginė puiku.
mindaugas@024-PC:~/prg/ghx/main/all/adapter-framework/src/main/java$
```

# Uzduotis5: Enum

---

ND

## 1. Tipas:

1. Alga
2. NT nuoma
3. Stipendija

**enum** Pajamos

---

**public enum** Pajamos {

*Alga*(1 "Darbo užmokestis atskaičius mokesčius")

*Uz*

*St*

*stipendija*");

**in**

*St*

*Pa*

}

}



**enum** Pajamos

---

**public enum** Pajamos {

***Alga***(1, "Darbo užmokestis atskaičius mokesčius"),

***UzNuoma***(2, "NT ar kito turto nuomos pajamos"),

***Stipendija***(3, "Universiteto, kolegijos ar profesinio rengimo centro mokama stipendija");

**int** indeksas;

String aprasymas;

Pajamos(**int** i, String s) {

    indeksas = i;

    aprasymas = s;

}

}

# Uzduotis7:

ND pabaigti!

- 
1. Reikalinga klasė *DienoraščioĮrašas*, kuri turi:
    - a. Privačius laukus
      - i. Kada atliktas įrašas:
        1. diena - skaičius
        2. Mėnuo - enum
        3. Metai - skaičius
        4. *Savaitės diena* - enum
      - ii. Aprašą - tekstas
    - b. Public metodus/elgseną
      - i. Galimybę nustatyti duomenis
      - ii. Galimybę atspausdinti duomenis
  2. Pademonstruoti kaip veikia: 2-3 įrašai
  3. *Panaudokime masyvą ir leiskime vartotojui pridėti įrašus*

# Uzduotis7: Mano biudžetas - Finansai

ND

---

Klasė, kuri apjungtu (pajamų/išlaidų kategorijos indeksą ir sumą [ir data, ir kt...])

Apsirašyti naujos klasės masyvą ir pabandyti priskirti reikšmes...

# Prisiminkime

---

- Klasės, objektai ir metodai yra java kalbos objektinio programavimo pagrindas.
- Mes aptarėme:
  - Kaip parašyti klasę
  - Kaip sukurti tos klasės tipo objektus
  - Kaip apibrėžti klasės laukus ir metodus
  - Kaip naudotis klasės laukais ir metodais
  - **GC**
  - ~~○ Kaip sukurti ir naudotis vidinėmis klasėmis~~
  - Kaip sukurti ir panaudoti išvardinamąjį (enum) tipą

# Klausimai

---