

# Rinktiniai Java skyriai

Pratybos #1

2007-09-13

Programavimas Java

# Prisiminti Java

- classes, packets
- access modifiers, constructors
- final members, static members, static initializers
- inheritance, abstract methods, interfaces, type casting
- method overloading and overriding
- packets
- exceptions

# Kas naujo Java 5.0 kalboje

- generics
- enhanced for loop
- varargs
- autoboxing
- enums
- static imports
- annotations

# Java best practices

- Rekomendacinio pobūdžio taisyklės:
  - kodo rašymui
  - projektavimui
  - bendrieji patarimai
- Padeda išvengti dažnų klaidų
- [www.javapractices.com](http://www.javapractices.com)
  - gerai organizuotas tokių taisyklių sąvadas

# Java best practices

- Naming conventions

- klasių vardai: PascalCase
- paketų vardai: lowercase
- konstantų vardai: UPPER\_CASE
- visa kita: camelCase

# Java best practices

- Naudokite paketus
  - `companyname.productname.subpackage.ClassName`
- Pvz:
  - `microsoft.vista.services.FancyGraphicsService`
- Mekomajame projekte bus leidžiama naudoti paprastesnius paketus:
  - `manoprojektas.ClassName`
  - `manoprojektas.kitaspaketas.ClassName`

# Java best practices

- Encapsulation & information hiding
  - “maksimalaus privatumo politika”:
    - neviešink nieko, ko nebūtina viešinti
  - visi laukai turi būti privatūs
- Setters and getters
  - `private FieldType fieldName;`
  - `public FieldType getFieldName();`
  - `public void setFieldName(FieldType value)`
    - tik jei reikia galimybės keisti lauko reikšmę
- JavaBeans

# Java best practices

```
class Student {  
  
    private int id;  
    private String name;  
  
    public Student() { }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
}
```

Ši klasė yra **JavaBean klasė**

# Java best practices

- Interfeisai VS abstrakčios klasės

# Java best practices

- Venk NullPointerException
  - String tipo duomenims nenaudok **null**. Naudok “”
  - Stenkis, kad tavo metodai niekada negražintų **null**
    - jei reikia gražinti null, tą būtina dokumentuoti
  - Jei nesi tikras, ar objektas nėra **null**, patikrink:
    - ```
if (obj != null) {  
    obj.doSomething();  
}
```
  - Nesikreipk į masyvo elementą, nepatikrinęs masyvo ilgio:
    - ```
if (array.length > 0) {  
    value = array[0];  
}
```

# Java best practices

- Exception handling

- **niekada gyvenime (!) nedaryk taip**

- ```
try {
    connectToDatabase();
} catch (Exception ex) {
}
```

- **pagavęs exception'ą, mesk jį toliau, ir pridėk papildomos informacijos apie tai, kas įvyko:**

- ```
try {
    connectToDatabase();
} catch (Exception ex) {
    throw new MyException("Could not connect to DB", ex);
}
```

# Java best practices

- Exception handling (2)

- visada atlaisvink resursus:

- ```
OutputStream file = null;
try {
    file = new FileOutputStream("file.txt");
    // write to file
} catch (IOException ex) {
    throw new Exception("Error writing file.txt", ex);
} finally {
    if (file != null) {
        file.close();
    }
}
```

# Java best practices

Uždavinukai