

Specifikacija i modeliranje softvera

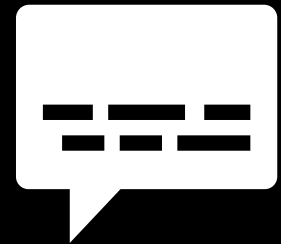
Čist dizajn
koda II

Nikola Luburić
nikola.luburic@uns.ac.rs

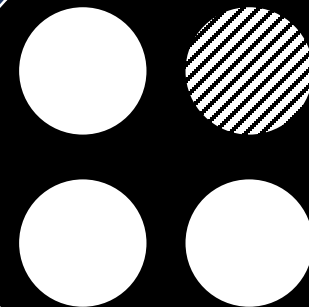
```
message =  
    if not hasattr(self, '_headers_buffer'):  
        self._headers_buffer = []  
    self._headers_buffer.append(("s %d %s\r\n" %  
                                (self.protocol_version, code, message))  
                                ('latin-1', 'strict'))  
  
    send_header(self, keyword, value):  
        """Send a MIME header to the headers buffer."""  
        if self.request_version != 'HTTP/0.9':  
            if not hasattr(self, '_headers_buffer'):  
                self._headers_buffer = []  
            self._headers_buffer.append(  
                ("%s: %s\r\n" % (keyword, value)).encode('latin  
        keyword.lower() == 'connection':  
            if value.lower() == 'close':  
                self.close_connection = True  
            elif value.lower() == 'keep-alive':  
                self.close_connection = False
```

Čist dizajn koda II

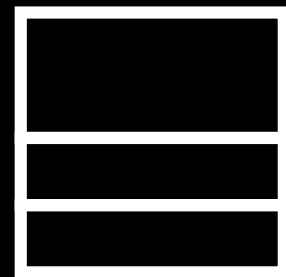
**Zašto treba izbegavati
komentare?**



Kako nastaju klase?



**Koja su svojstva dobro
pisanih klasa?**



Čist dizajn koda II



Komentari?



Robert Martin (Uncle Bob)

Author of Clean Code

The proper use of comments is to compensate for our failure to express ourselves in code. Comments are always failures. We must have them because we cannot always figure out how to express ourselves without them.

Čist dizajn koda II



Komentari?

Komentari ne nadoknađuju za loš kod

```
// Check to see if the employee  
// is eligible for full benefits  
if ((employee.flags & HOURLY_FLAG) &&  
    (employee.age > 65))
```

Izrazi komentar kroz kod

```
if (employee.isEligibleForFullBenefits())
```

Čist dizajn koda II



Komentari?

Kada su komentari korisni?

```
8 // Dear programmer:
9 // When I wrote this code, only god and
10 // I knew how it worked.
11 // Now, only god knows it!
12 //
13 // Therefore, if you are trying to optimize
14 // this routine and it fails (most surely),
15 // please increase this counter as a
16 // warning for the next person:
17 //
18 // total_hours_wasted_here = 254
19 //
20
```

Čist dizajn koda II



Komentari?

Kada su komentari korisni?

- ❖ Prvi korak u refaktorisanju
- ❖ Objašnjavanje namere

```
//Our best attempt to get a race condition  
//by creating a large number of threads.  
for (int i = 0; i < 25000; i++) {  
    BuilderThread builderThread =  
        new BuilderThread(builder, failFlag);  
    Thread thread = new Thread(builderThread);  
    thread.start();  
}  
assertEquals(false, failFlag.get());
```

Čist dizajn koda II



Komentari?

Kada su komentari korisni?

- ❖ Prvi korak u refaktorisanju
- ❖ Objašnjavanje namere
- ❖ Isticanje „osetljivog“ koda

```
public static SimpleDateFormat makeDate () {  
    //SimpleDateFormat is not thread safe, so we  
    //need to create each instance independently.  
    SimpleDateFormat df = new SimpleDateFormat(  
        "EEE, dd MMM yyyy HH:mm:ss z");  
    df.setTimeZone(TimeZone.getTimeZone("GMT"));  
    return df;  
}
```

Čist dizajn koda II



Komentari?

Kada su komentari korisni?

- ❖ Prvi korak u refaktorisanju
- ❖ Objašnjavanje namere
- ❖ Isticanje „osetljivog“ koda

```
// need to reset foo before calling bar due  
// to a bug in the foo component.
```

```
foo.reset()  
foo.bar();
```


Čist dizajn koda II



Komentari?

Kada su komentari korisni?

- ❖ Prvi korak u refaktorisanju
- ❖ Objašnjavanje namere
- ❖ Isticanje „osetljivog“ koda
- ❖ Dokumentovanje „javnog“ API-a

Z9.1: Istraži kako se dokumentuje kod za tvoj jezik

Interfejsi
koje koriste
drugi timovi

Endpoint-i
serverske
aplikacije

Kod

Javne klase
i funkcije
biblioteke

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

❖ Mumlanje

```
public void loadProperties() {  
    try {  
        String propPath = location + PROP_FILE;  
        FileInputStream propertiesStream =  
            new FileInputStream(propPath);  
        loadedProperties.load(propertiesStream);  
    }  
    catch(IOException e) {  
        // No prop file means defaults are loaded  
    }  
}
```

*koja pitanja
ovo povlači?*

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

❖ Mumlanje

*previše
informacije*

```
/* RFC 2045 - Multipurpose Internet Mail Extensions (MIME)  
Part One: Format of Internet Message Bodies section 6.8.  
Base64 Content-Transfer-Encoding The encoding process  
represents 24-bit groups of input bits as output strings of 4  
encoded characters. Proceeding from left to right, a 24-bit  
input group is formed by concatenating 3 8-bit input groups.  
These 24 bits are then treated as 4 concatenated 6-bit  
groups, each of which is translated into a single digit in  
the base64 alphabet. When encoding a bit stream via the  
base64 encoding, the bit stream must be presumed to be  
ordered with the most-significant-bit first. The first bit in  
the stream will be the high-order bit in the first 8-bit  
byte, and the eighth bit will be the low-order bit in the  
first 8-bit byte, and so on.*/
```

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost

```
/**
```

```
 * Port on which fit would run.
```

```
 * Defaults to 8082.
```

```
 * @param fitPort
```

```
 */
```

```
public void setFitPort(int fitPort)
```

```
{
```

```
    this.fitPort = fitPort;
```

```
}
```

*nije lokalan
podataka*

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost

```
/**  
    * @param title The title of the CD  
    * @param author The author of the CD  
 */  
void addCD(String title, String author) {  
    CD cd = new CD(title, author);  
    cdList.add(cd);  
}
```

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost

```
/**  
 * Returns the day of the month.  
 *  
 * @return the day of the month.  
 */  
public int getDayOfMonth() {  
    return dayOfMonth;  
}
```

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost
- ❖ Laž

```
// Utility method that returns when this.closed is  
// true. Throws exception if the timeout reached  
synchronized void waitForClose(long timeoutMillis)  
    throws Exception {  
    if(!closed) {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("Could not be closed");  
    }  
}
```

Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost
- ❖ Laž

```
MockRequest request;
```

```
private String HTTP_DATE_REGEXP =
```

```
    "[SMTWF][a-z]{2}\\,\\s[0-9]{2}\\s[JFMASOND][a-z]{2}\\s" +
```

```
    "[0-9]{4}\\s[0-9]{2}\\:[0-9]{2}\\:[0-9]{2}\\sGMT";
```

```
private Response response;
```

```
private FileResponder responder;
```

```
private Locale saveLocale;
```

```
// Example: "Tue, 02 Apr 2003 22:18:49 GMT"
```


Čist dizajn koda II



Komentari?

Šta je zlo kod komentara?

- ❖ Mumlanje
- ❖ Redundantnost
- ❖ Laž
- ❖ Zakomentarisan kod

```
InputStreamResponse response = new InputStreamResponse();  
response.setBody(formatter.getStream(), formatter.getCount());  
// InputStream resultsStream = formatter.getResultStream();  
// StreamReader reader = new StreamReader(resultsStream);  
// response.setContent(reader.read(formatter.getByteCount()));
```

*da li smem ovo
da obrišem?*

Čist dizajn koda II



Komentari?



Nastanak



klasa?

Kako nastaju klase?

Diktira problem

Strukture podataka

Ponašanje

Čist dizajn koda II



Komentari?



Nastanak
klasa?

Šta je struktura podataka, a šta objekat?

```
public class Play {  
    public String name;  
    public String type;  
}
```

*šta je formalno
instanca ove klase?*

```
public class Play {  
    private String name;  
    private String type;  
    public String getType() { return type; }  
    public void setType(String t) { this.type = t; }  
    public String getName() { return name; }  
    public void setName(String n) { this.name = n; }  
}
```

*šta je suštinski
instanca ove klase?*

Čist dizajn koda II



Komentari?



Nastanak

klasa?

Šta je struktura podataka, a šta objekat?

- ❖ Struktura podataka izlaže podatke, nema ponašanje

```
public interface FuelTank {  
    double getTankCapacityInLiters();  
    double getLitersOfGasoline();  
}
```

- ❖ Objekat sakriva podatke, izlaže ponašanje

```
public interface FuelTank {  
    double getPercentFuelRemaining();  
}
```

Čist dizajn koda II



Komentari?



Nastanak
klasa?

Kad treba grupa podataka, a kad objekat?

Square	
+ topLeft	: Point
+ side	: double

Rectangle	
+ topLeft	: Point
+ height	: double
+ width	: double

Circle	
+ center	: Point
+ radius	: double

Geometry	
+ area (Object shape)	: double

kako izgleda kod?

*šta se menja ako
dodamo račun za obim?*

*šta se menja ako
dodamo Triangle klasu?*

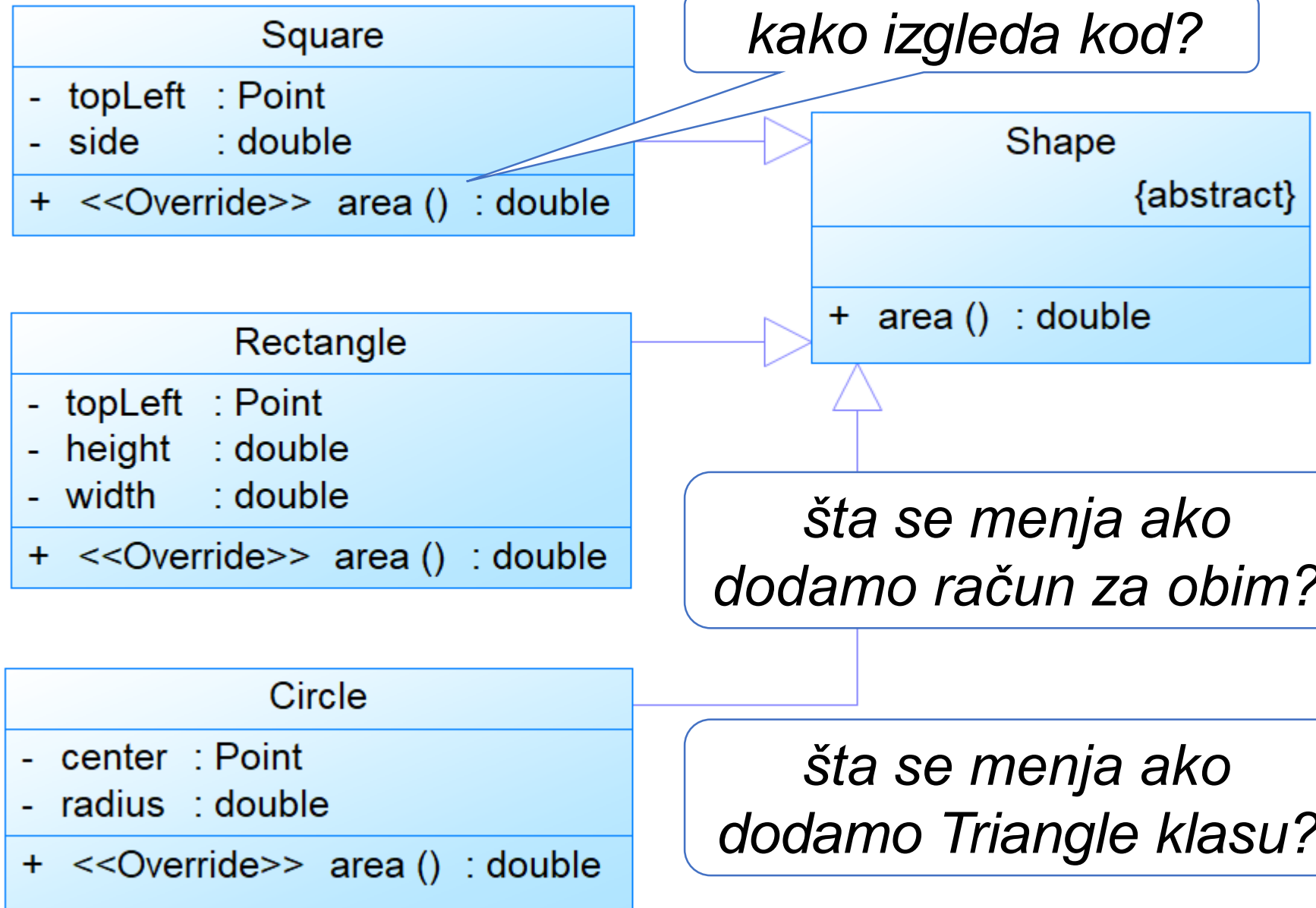
Čist dizajn koda II



Komentari?

● ● Nastanak
● ● klasa?

Kad treba grupa podataka, a kad objekat?



Čist dizajn koda II



Komentari?



Nastanak
klasa?

Kad treba grupa podataka, a kad objekat?

- ❖ Struktura podataka se ne menja
kada se novo ponašanje uvodi
- ❖ Postojeće ponašanje se menja
kada se nova struktura podataka uvodi

- ❖ Klasa objekta se menja
kada se novo ponašanje uvodi
- ❖ Postojeće ponašanje se ne menja
kada se nova klasa objekata uvodi

Čist dizajn koda II



Komentari?



Nastanak



klasa?

Kad treba grupa podataka, a kad objekat?

❖ Strukture podataka su stanje
Glupava grupa podataka (*DTO*)

❖ Objekti su ponašanje
Pamet i dinamika

❖ Hibridi su *smell*

Čist dizajn koda II



Komentari?



Nastanak



klasa?

Kako nastaju klase?

Diktira problem

Strukture podataka
Ponašanje

Diktira tehnologija

Šabloni odabranih
tehnologija (*framework...*)

Čist dizajn koda

Iz primitiva
Iz funkcija
Iz klase

Čist dizajn koda II



Komentari?



Nastanak
klasa?

Ekstrakcija klase iz primitiva

```
int countHigherPriority(Order[] orders) {  
    int count = 0;  
    for(Order o: orders)  
        if(o.prio.equals("high") || o.prio.equals("rush"))  
            count++;  
    return count;  
}
```

*izbroj sve sa većim
prioritetom od low?*

```
int countHigherPriority(Order[] orders, Priority p) {  
    int count = 0;  
    for(Order o: orders)  
        if(o.prio.higherThan(p))  
            count++;  
    return count;  
}
```

*šta sadrži
Priority klasa?*

Čist dizajn koda II



Komentari?

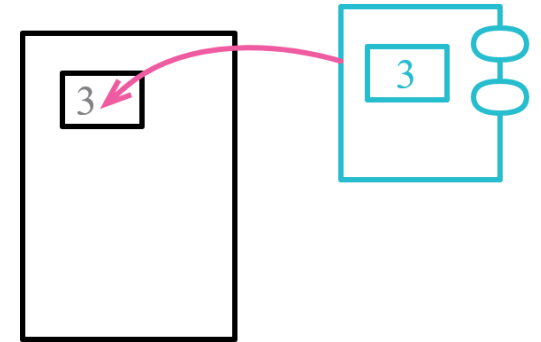


Nastanak
klasa?

Ekstrakcija klase iz primitiva

- ❖ Osnovni tipovi domena
- ❖ Grupa primitiva koje često „idu zajedno“

```
class Coordinates  
class Range  
class Money
```



- ❖ Posebno obratiti pažnju na *String*

Čist dizajn koda II



Komentari?

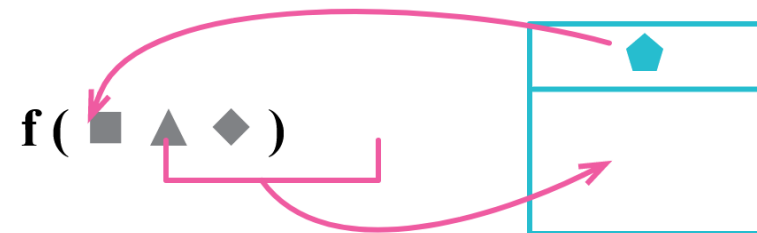


Nastanak
klasa?

Ekstrakcija klase iz parametra funkcije

❖ Grupa podataka koja često „idu zajedno“

```
double amountInvoiced(Date start, Date end) {...}  
double amountReceived(Date start, Date end) {...}  
double amountOverdue(Date start, Date end) {...}
```



```
double amountInvoiced(DateRange range) {...}  
double amountReceived(DateRange range) {...}  
double amountOverdue(DateRange range) {...}
```

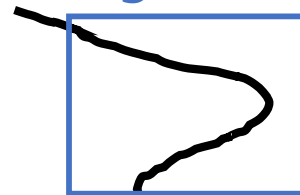
Čist dizajn koda II

Ekstrakcija klase iz funkcije

```
importModel(...) {
```

```
  Object A;
```

```
  Object B;
```



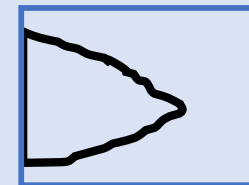
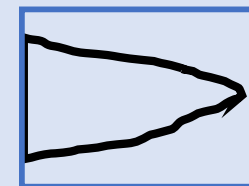
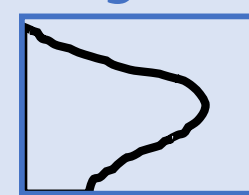
A

A, B

A, B

```
Object A;
```

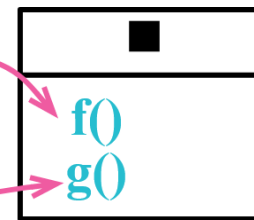
```
Object B;
```



Class AB

f(■)

g(■)



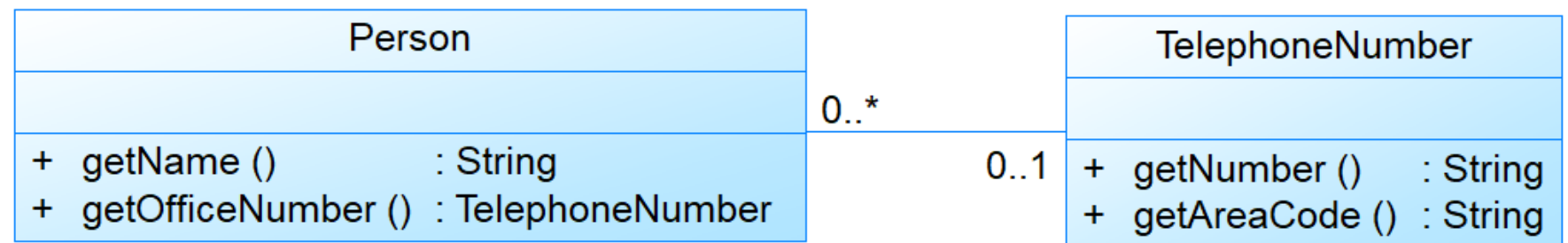
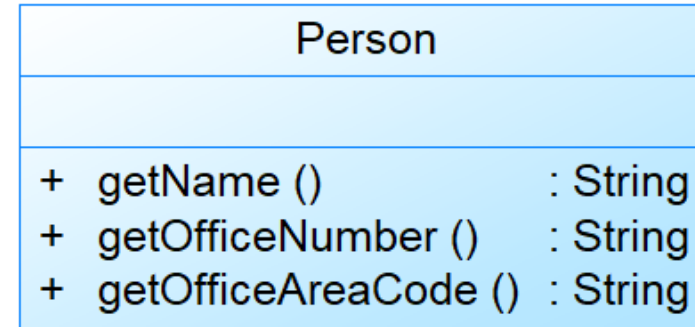
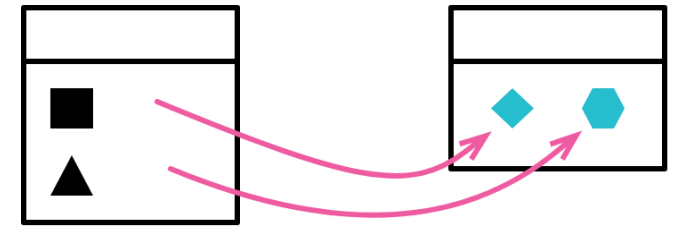
Komentari?

● ● Nastanak
● ● klasa?

Čist dizajn koda II

Ekstrakcija klasa iz klase

❖ Podela odgovornosti



Komentari?



Nastanak
klasa?

Čist dizajn koda II

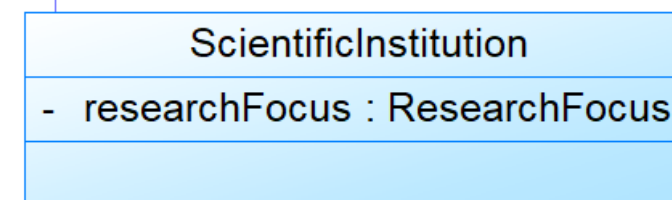
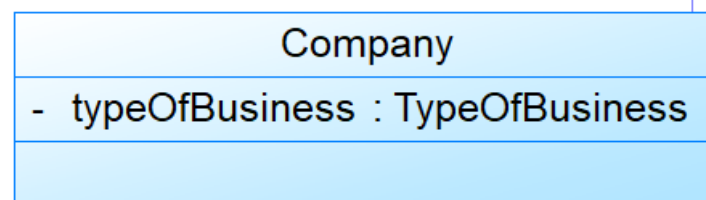
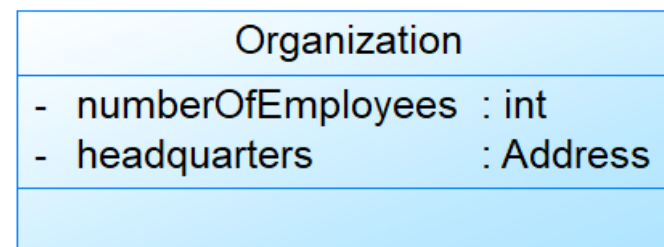
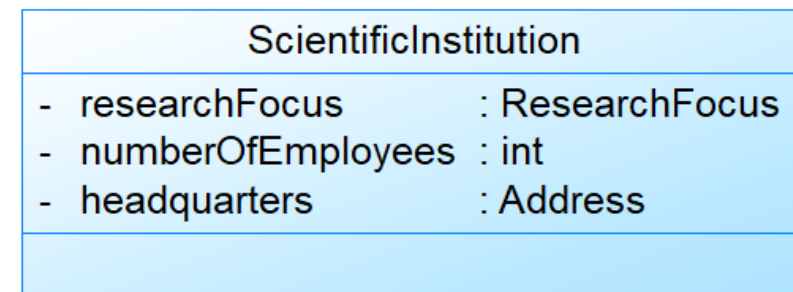
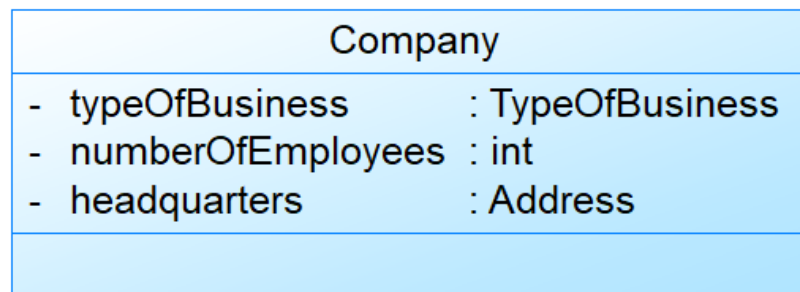
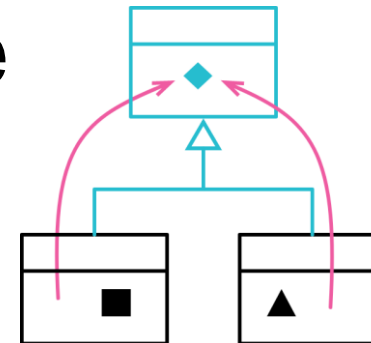


Komentari?

● ● Nastanak
● ● klasa?

Ekstrakcija klasa iz klase

- ❖ Podela odgovornosti
- ❖ Razrada hijerarhije



Čist dizajn koda II



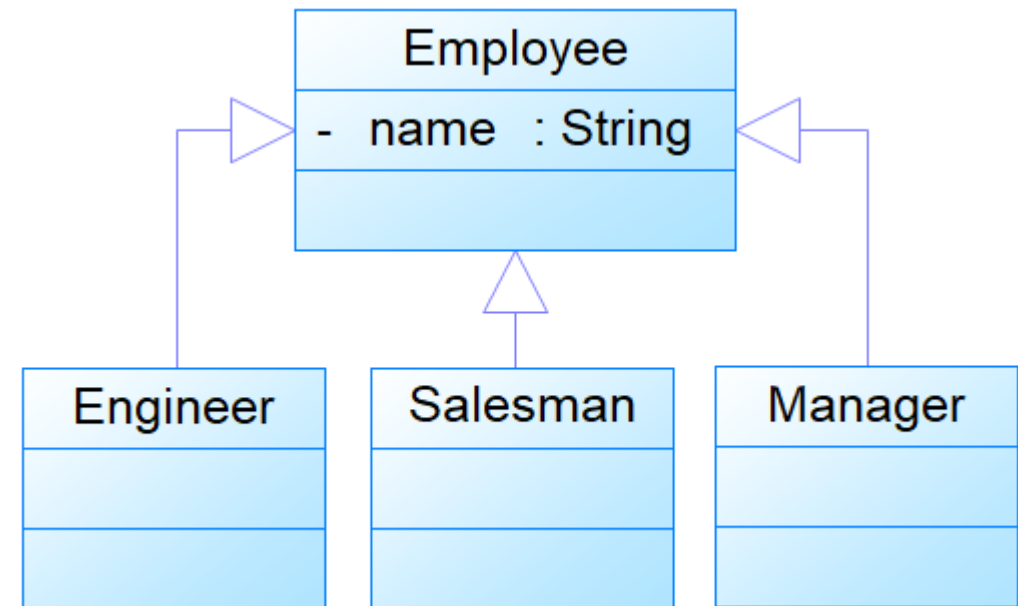
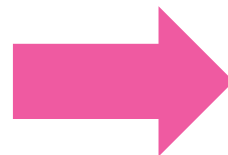
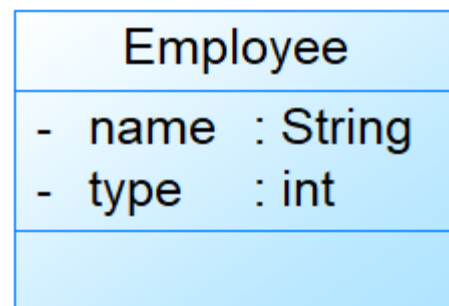
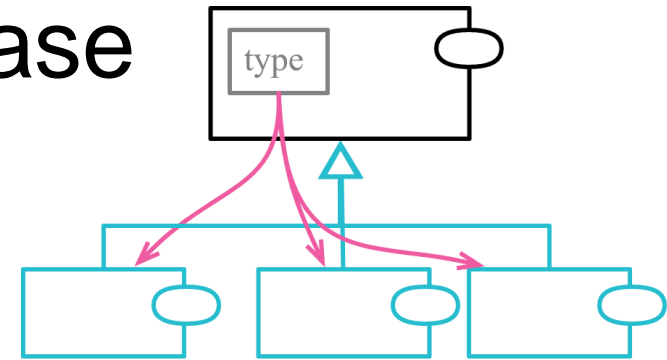
Komentari?



Nastanak
klasa?

Ekstrakcija klase iz klase

- ❖ Podela odgovornosti
- ❖ Razrada hijerarhije



- ❖ Polimorfizam umesto *switch* bloka

Čist dizajn koda II



Komentari?



Nastanak



klasa?

Kako nastaju klase?

Diktira problem

Strukture podataka
Ponašanje

Diktira tehnologija

Šabloni odabranih
tehnologija (*framework...*)

Čist dizajn koda

Iz primitiva
Iz funkcija
Iz klase

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Šta ima dobro napisana klasa?

Značajno ime

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klasa treba da bude mala

❖ Meri se u broju odgovornosti

```
public class SuperDashboard extends JFrame {  
    Class[] getDataBaseClasses()  
    MetadataFeeder getMetadataFeeder()  
    void addProject(Project project)  
    boolean setCurrentProject(Project project)  
    boolean removeProject(Project project)  
    Project loadProject(String projectName)  
    MetaProjectHeader getProgramMetadata()  
    void resetDashboard()  
}
```

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klasa treba da bude mala

- ❖ Meri se u broju odgovornosti

```
public class SuperDashboard extends JFrame {  
    Component getLastFocusedComponent()  
    void setLastFocused(Component lastFocused)  
    int getMajorVersionNumber()  
    int getMinorVersionNumber()  
}
```

Opiši klasu u do 25 reči bez „i“, „ili“, „ako“

- ❖ *SuperDashboard* vodi računa o poslednje pristupljenoj komponenti i verziji aplikacije

Čist dizajn koda II



Komentari?



Nastanak
klasa?

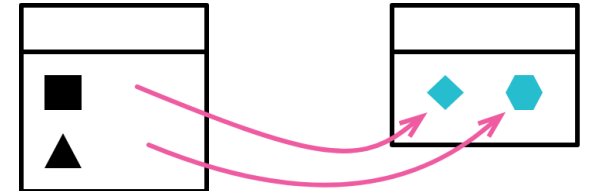


Dobro pisane
klase?

Klasa treba da bude mala

❖ Meri se u broju odgovornosti

```
public class SuperDashboard extends JFrame {  
    Component getLastFocusedComponent()  
    void setLastFocused(Component lastFocused)  
}  
  
public class Version  
    int getMajorVersionNumber()  
    int getMinorVersionNumber()  
}
```



Klasa treba da ima jednu odgovornost

Čist dizajn koda II

Kompleksnost je neizbežna ako smo ozbiljni



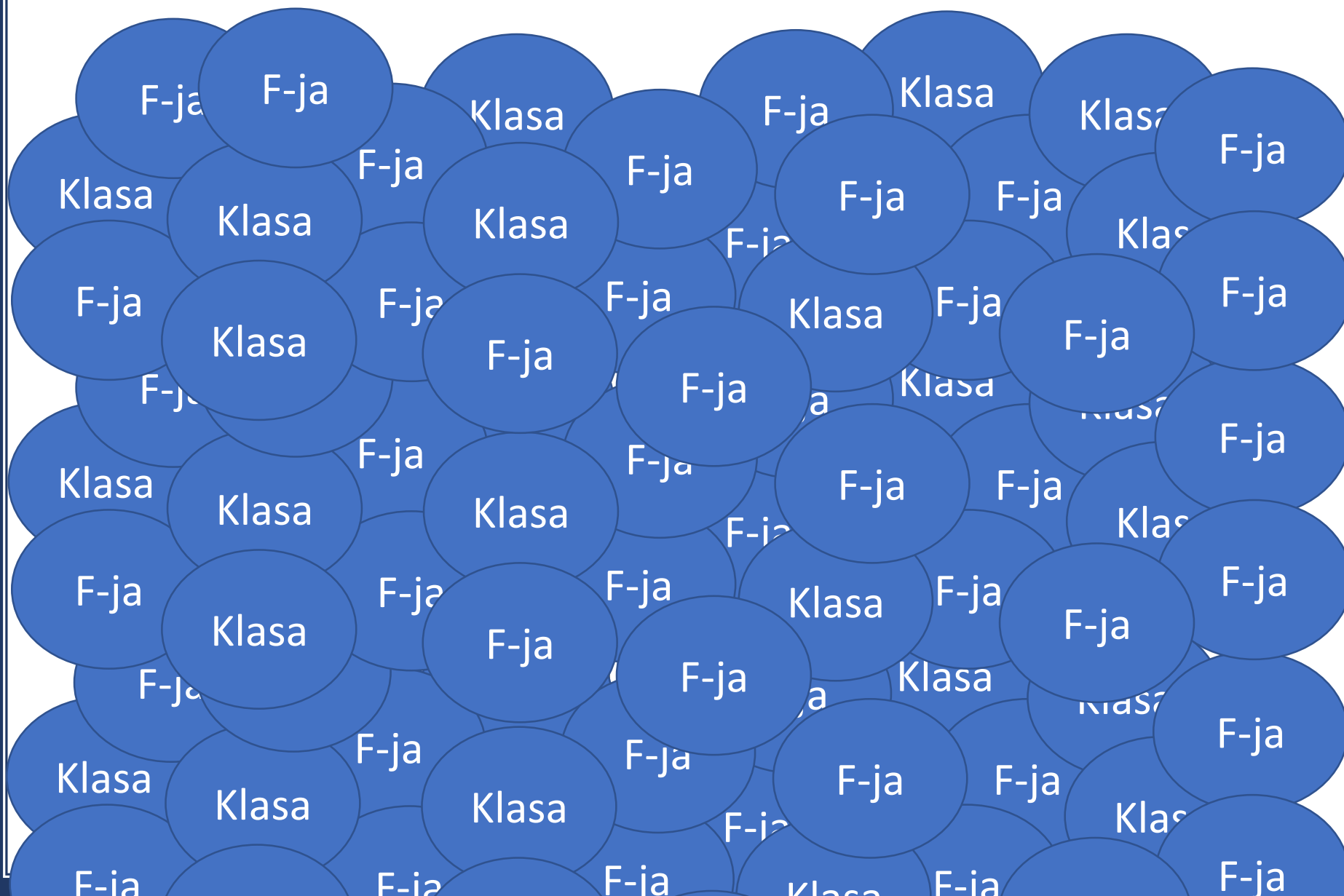
Komentari?



Nastanak
klasa?



Dobro pisane
klase?



Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klasa treba da ima visoku koheziju

- ❖ Stepen upotrebe polja od strane metoda
- ❖ Max kad sve metode koriste sva polja

```
class Stack {  
    int topOfStack = 0;  
    List<Object> elements = new LinkedList<Object>();  
  
    int size() { return topOfStack; }  
    void push(int element) {  
        topOfStack++;  
        elements.add(element);  
    }  
}
```

*visoka ili niska
kohezija?*

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klasa treba da ima visoku koheziju

- ❖ Stepen upotrebe polja od strane metoda
- ❖ Max kad sve metode koriste sva polja

```
class Example {
```

```
    Object A;
```

```
    Object B;
```

```
    Object C;
```

```
    void methodA()    // uses A
```

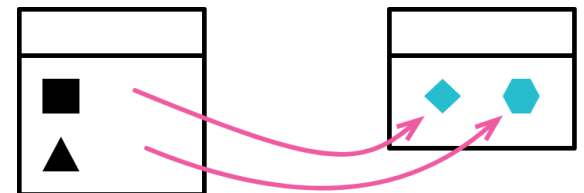
```
    void methodBA()  // uses B and A
```

```
    void methodC1()  // uses C
```

```
    void methodC2()  // uses C
```

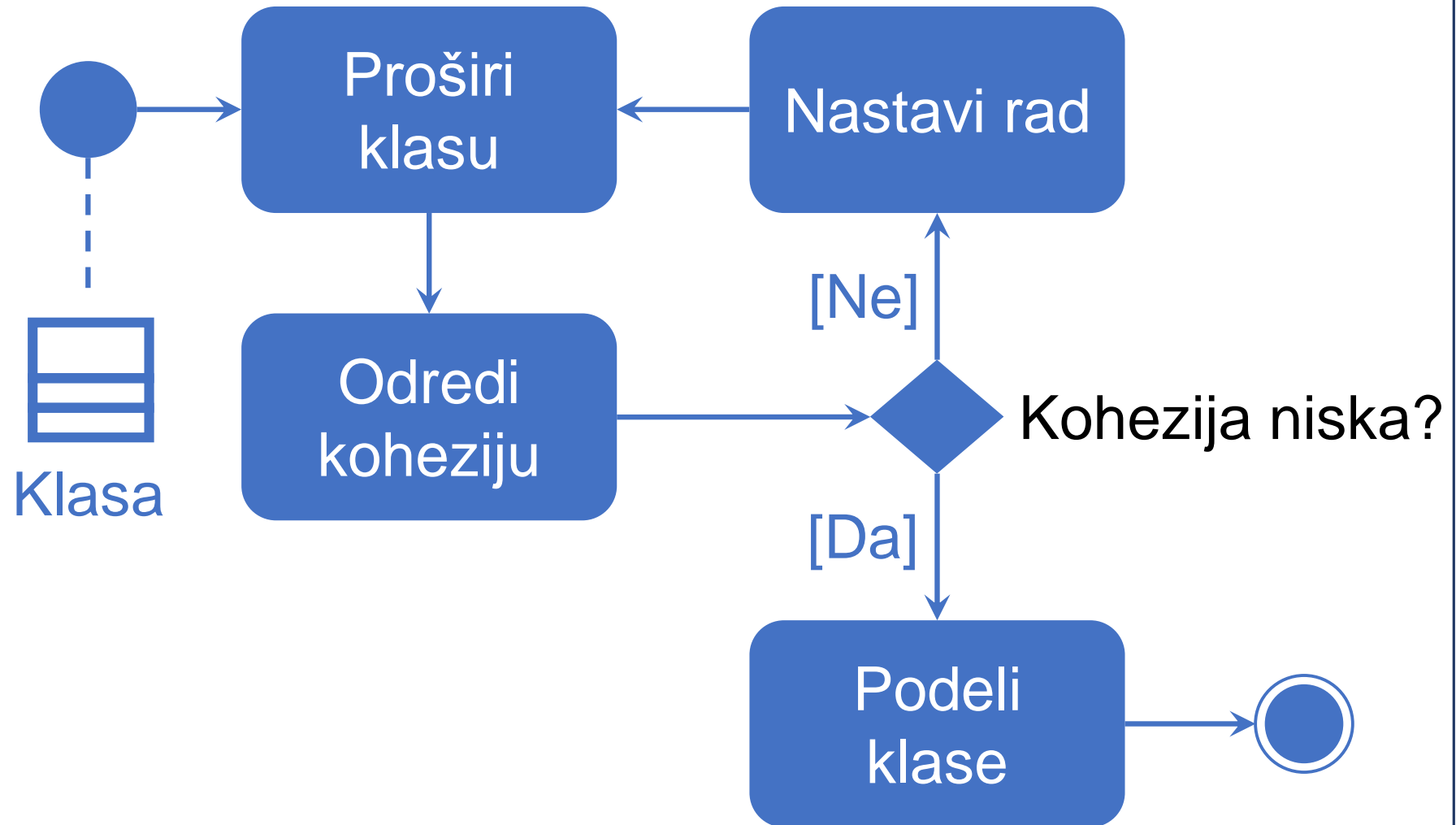
```
}
```

*visoka ili niska
kohezija?*



Čist dizajn koda II

Kako održavati visoku koheziju klase?



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Čist dizajn koda II

Kohezija je širi pojam



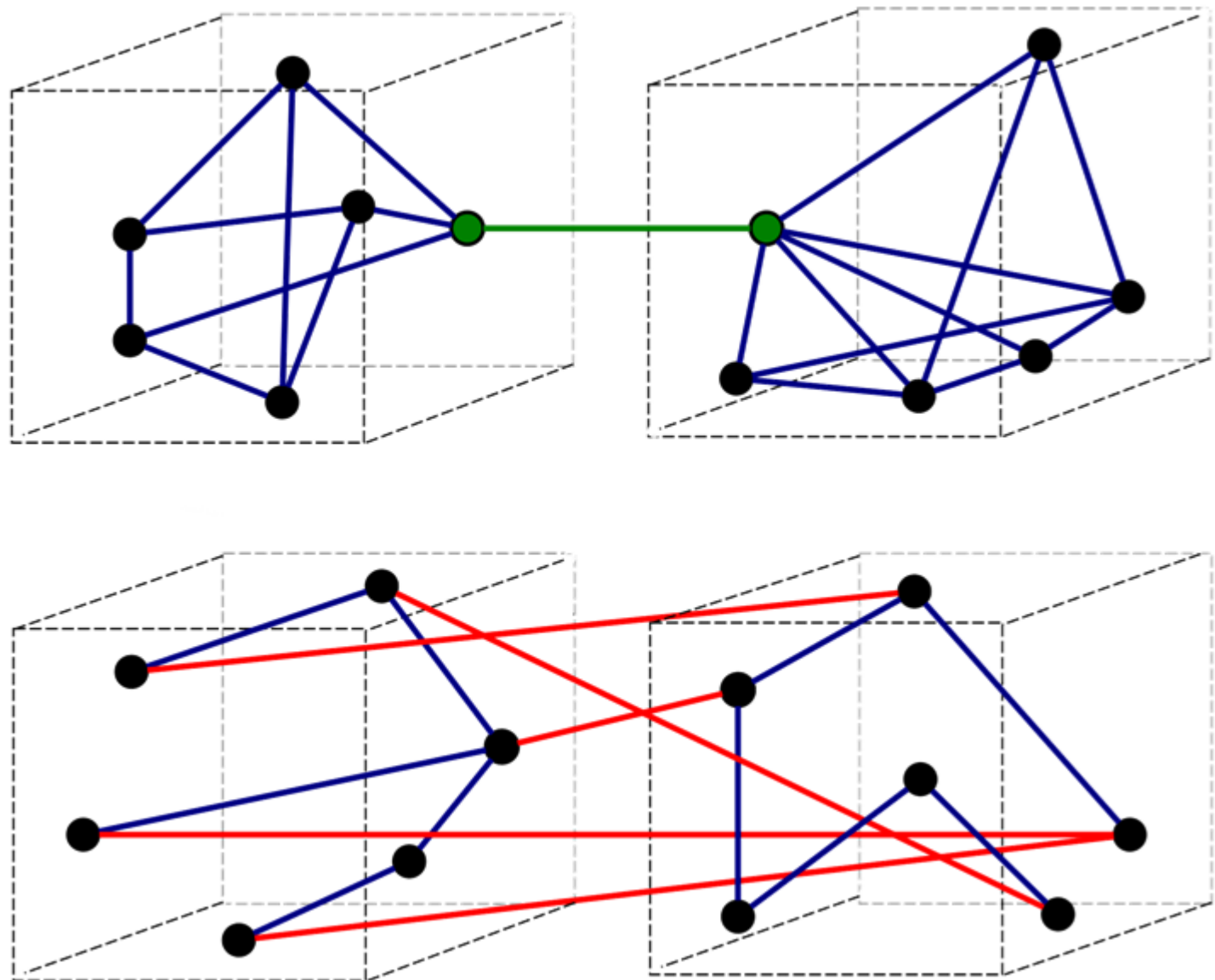
Komentari?



Nastanak
klasa?



Dobro pisane
klase?



Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Objekti treba da kriju svoje detalje

❖ *Law of Demeter*

```
class C {  
    Object A;  
    void M(Object P) {  
        // Kod od M sme pozivati:  
        // 1) metode klase C  
        // 2) metode objekta P  
        // 3) metode objekata kreiranih unutar M  
        // 4) metode objekta A  
        // 5) funkcije globalnih promenljivih  
    }  
}
```

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Objekti treba da kriju svoje detalje

❖ *Law of Demeter*

```
class C {  
    Object A;  
    void M(Object P) {  
        // Kod od M ne treba pozivati:  
        //     metode objekata koji su povratna  
        //     vrednost prethodnih pet slučajeve  
    }  
}
```

❖ *Use only one dot*

❖ Ne odnosi se na strukture podataka

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Objekti treba da kriju svoje detalje

❖ *Law of Demeter*

Paperboy
- collectedMoney : int
+ collectPayment (Buyer b) : void

Wallet
- money : int
+ getMoney () : int
+ reduceMoney (int amount) : void

0..1

0..1

Buyer
+ getWallet () : Wallet

```
void collectPayment(Buyer b) {  
    if(b.getWallet().getMoney() > 10) {  
        b.getWallet().reduceMoney(10);  
        this.collectedMoney += 10;  
    }  
}
```

šta sve može dečak da uradi sa novčanikom?

Čist dizajn koda II

Objekti treba da kriju svoje detalje

❖ *Law of Demeter*

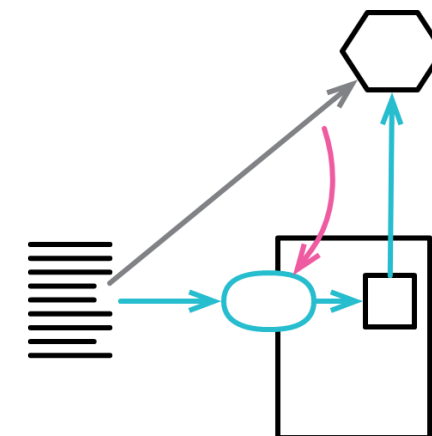
Paperboy
- collectedMoney : int
+ collectPayment (Buyer b) : void

Wallet
- money : int
+ getMoney () : int
+ reduceMoney (int amount) : void

0..1	0..1
Buyer	
+ pay (int amount) : void	

```
void collectPayment(Buyer b) {  
    b.pay(10);  
    this.collectedMoney += 10;  
}
```

"tell, don't ask"



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klase treba izolovati od promena

❖ *Law of Demeter*

```
void applyDiscount(Customer customer,  
    int orderId, double discount) {  
    Order[] orders = customer.getOrders();  
    Order order = orders.find(orderId);  
    Totals totals = order.getTotals();  
    totals.setGrandTotal(  
        totals.getGrandTotal() - discount);  
    totals.setDiscount(discount);  
}
```

*koliko nivoa
apstrakcije?*

Čist dizajn koda II



Komentari?



Nastanak
klasa?



Dobro pisane
klase?

Klase treba izolovati od promena

❖ *Law of Demeter*

```
void applyDiscount(Customer customer,  
    int orderId, double discount) {  
    customer  
        .getOrder(orderId)  
        .applyDiscount(discount);  
}
```

customer

.applyDiscountToOrder(orderId, discount) ?

Good Guideline of Demeter

Čist dizajn koda II



Komentari?



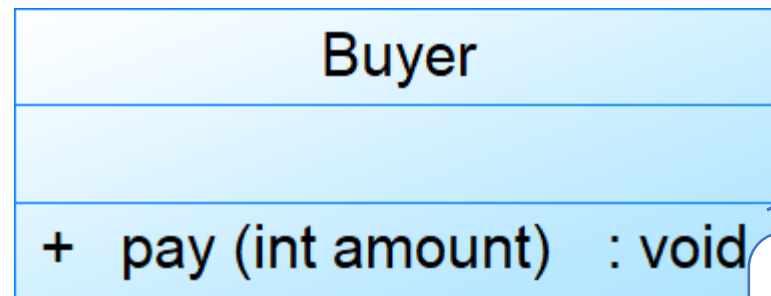
Nastanak
klasa?



Dobro pisane
klase?

Klase treba izolovati od promena

❖ *Law of Demeter*

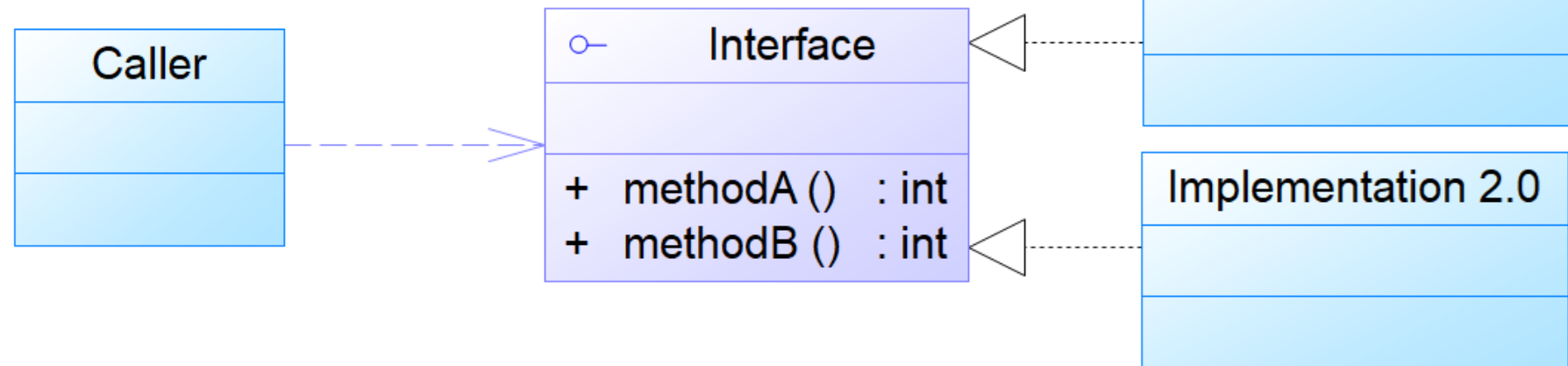


*novčanik,
kartica, bitcoin?*

*Paperboy
se ne menja*

*Caller ne
brine za ovo*

❖ Apstraktne klase, interfejsi



Čist dizajn koda II



Komentari?

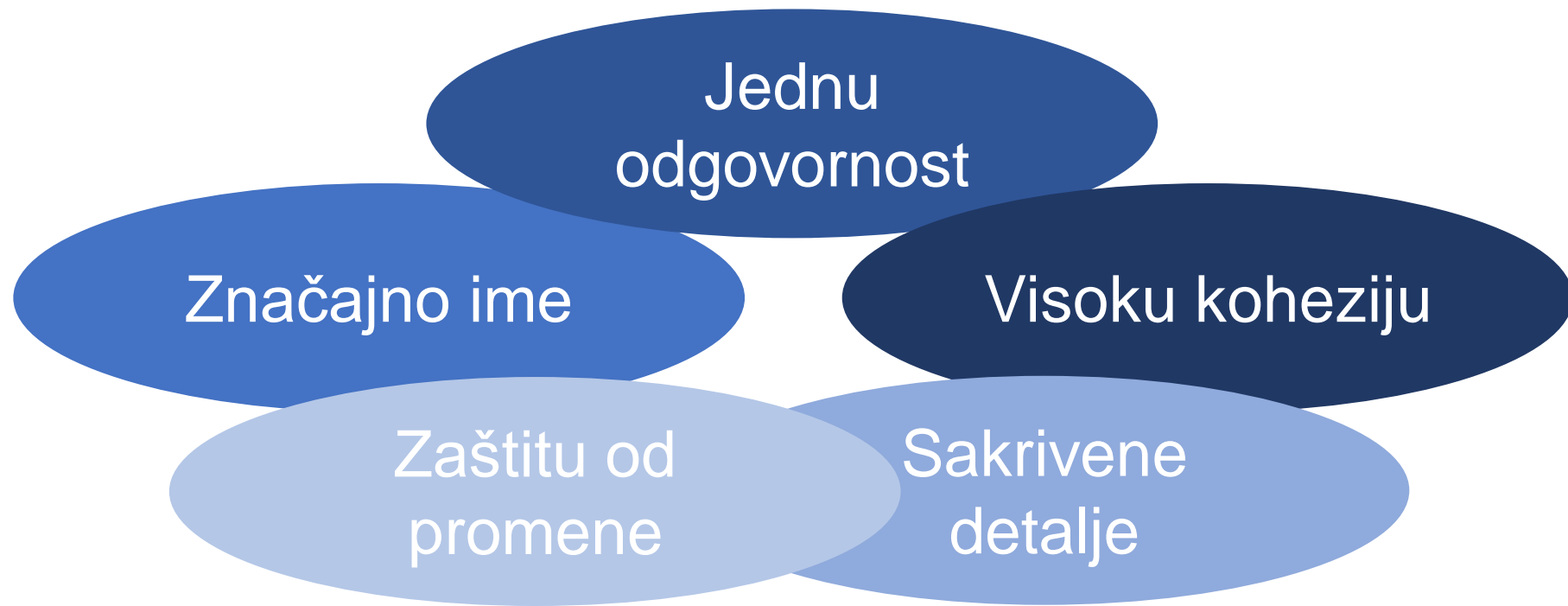


Nastanak
klasa?



Dobro pisane
klase?

Šta ima dobro napisana klasa?



Čist dizajn koda II



Komentari?



Nastanak



klasa?



Dobro pisane
klase?

Z9.2: Refaktoriši kod



<https://pastebin.com/3ctc0RRj>