

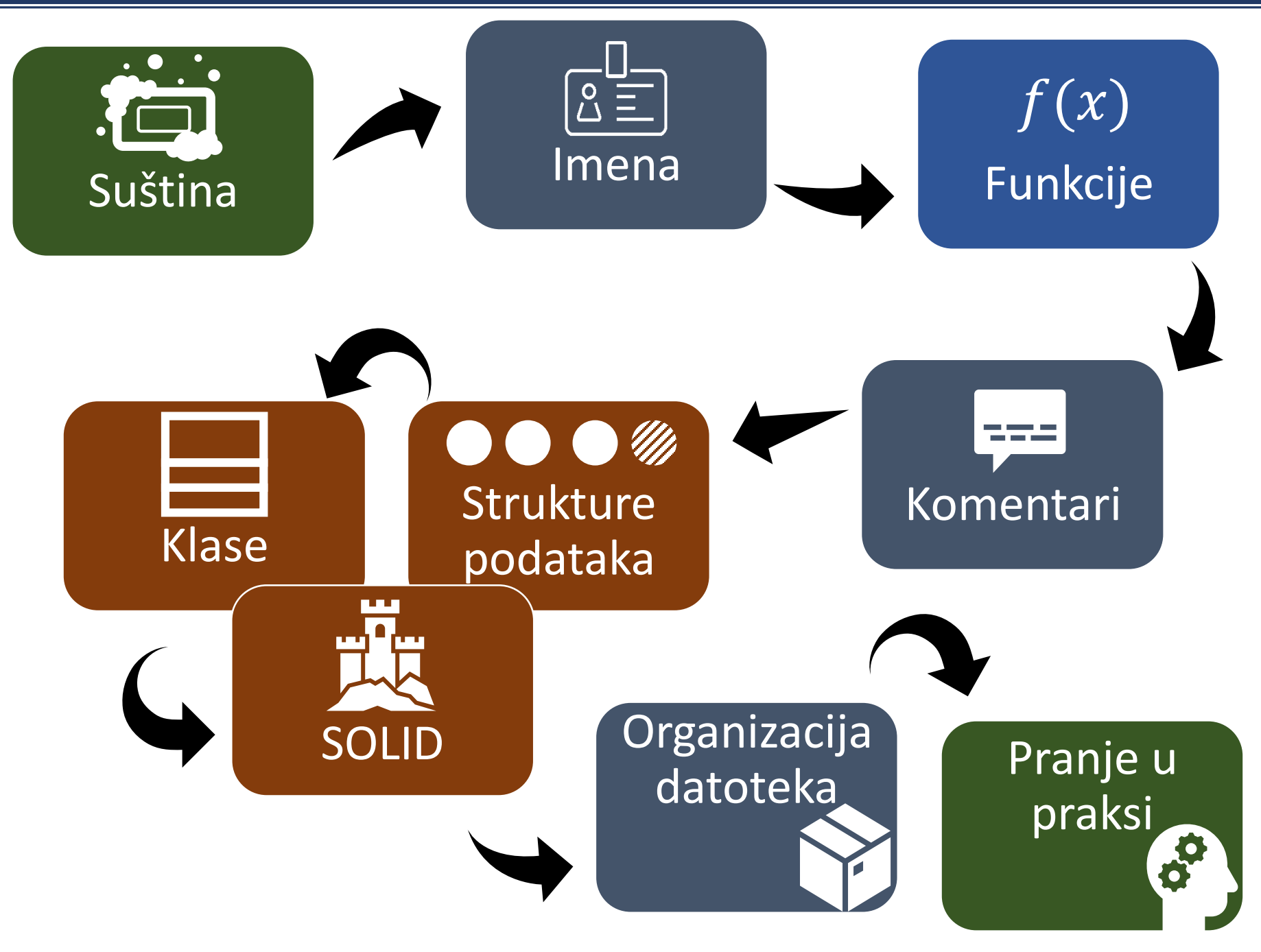
Specifikacija i modeliranje softvera

Čist dizajn
koda I

Nikola Luburić
nikola.luburic@uns.ac.rs

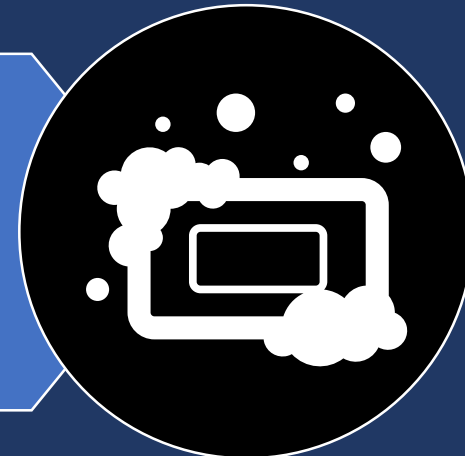
```
message =  
if not hasattr(self, '_headers_buffer'):  
    self._headers_buffer = []  
self._headers_buffer.append(("s %d %s\r\n" %  
    (self.protocol_version, code, message))  
    'latin-1', 'strict'))  
  
send_header(self, keyword, value):  
    """Send a MIME header to the headers buffer."""  
    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



Čist dizajn koda I

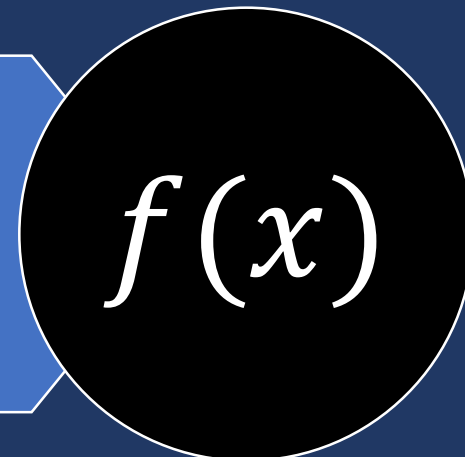
Šta predstavlja
čist kod?



Kako formirati
značajna imena?



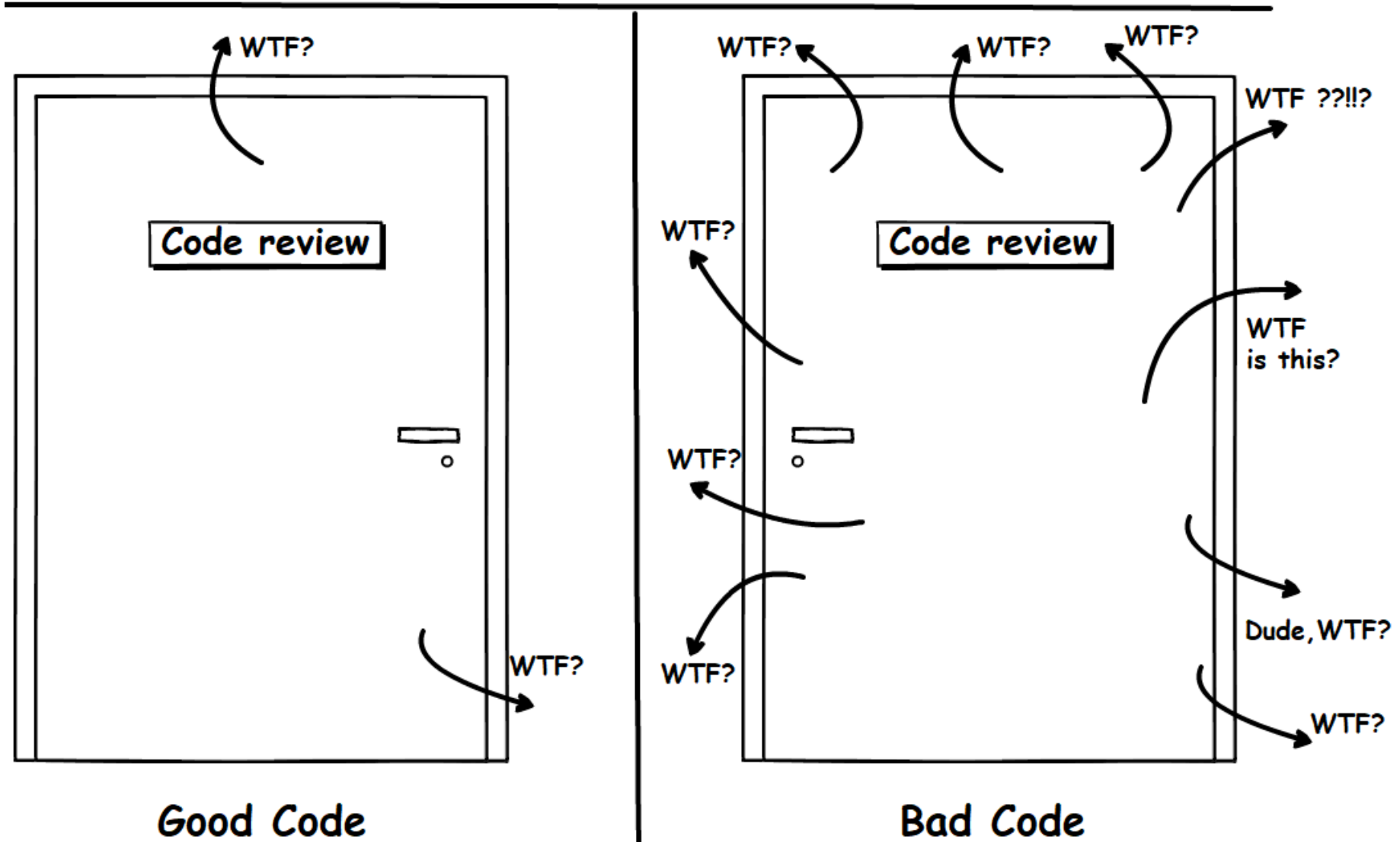
Kako pisati
čitljive funkcije?



Čist dizajn koda I



Code quality measurement: WTFs per minute



Čist dizajn koda I



Čist kod?



Bjarne Stroustrup

Creator of C++

I like my code to be elegant and efficient. The logic should be straightforward to make it hard for bugs to hide, the dependencies minimal to ease maintenance, error handling complete according to an articulated strategy, and performance close to optimal so as not to tempt people to make the code messy with unprincipled optimizations. Clean code does one thing well.

Čist dizajn koda I



Čist kod?



Dave Thomas

Godfather of the Eclipse strategy

Clean code can be read, and enhanced by a developer other than its author. It has unit and acceptance tests. It has meaningful names. It provides one way rather than many ways for doing one thing. It has minimal dependencies, which are explicitly defined, and provides a clear and minimal API. Code should be literate as not all information can be expressed in language constructs.

Čist dizajn koda I



Čist kod?



Michael Feathers

Working Effectively with Legacy Code

Clean code always looks like it was written by someone who cares. There is nothing obvious that you can do to make it better. All of those things were thought about by the code's author, and if you try to imagine improvements, you're led back to where you are, sitting in appreciation of the code someone left for you—code left by someone who cares deeply about the craft.

Čist dizajn koda I



Čist kod?

Čist kod je čitljiv kod –
jasan, elegantan, minimalan

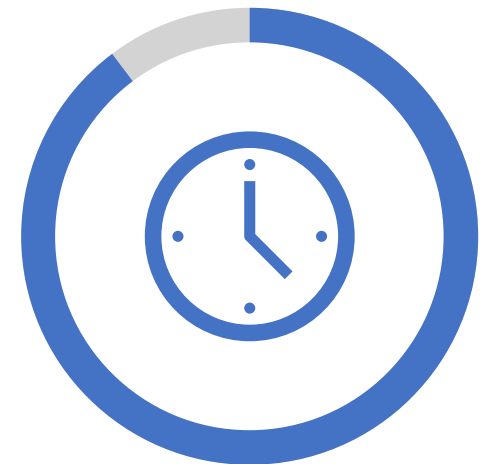
Koliko se kod čita, a koliko piše?

❖ Na zreloom projektu 10:1

❖ $devTime = R + W$

Čitljiv kod $R \rightarrow 0$

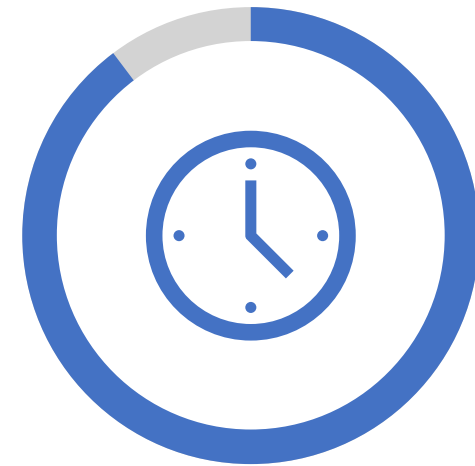
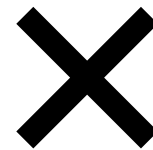
Aljkav kod $R \rightarrow \infty$



Čist dizajn koda I



Čist kod?



Održavanje košta
40 do 80%
ukupnog troška

Na zrelom projektu
odnos čitanja i
pisanja je 10:1



Većina troška razvoja je u čitanju koda
Čist dizajn koda **značajno** smanjuje troškove

Čist dizajn koda I



Čist kod?



Značajna
imena?

Imena prožimaju softver

Promenljive

Klase

Datoteke

Funkcije

Paketi

Jars, DLLs...

Ime treba da otkrije **svrhu** koncepta

- ❖ Zašto postoji?
- ❖ Šta radi?
- ❖ Kako se koristi?

Čist dizajn koda I



Čist kod?



Značajna
imena?

```
int d; // elapsed time in days
```

```
int elapsedTimeInDays;
```

```
int daysSinceCreation;
```

```
int daysSinceModification;
```

```
int fileAgeInDays;
```

Čist dizajn koda I



Čist kod?



Značajna
imena?

Šta radi ovaj kod?

```
public List<int[]> getThem() {  
    List<int[]> list = new ArrayList<int[]>();  
    for (int[] x : theList)  
        if (x[0] == 4)  
            list.add(x);  
    return list;  
}
```

*šta se ovde
nalazi?*

*koji je značaj
ovoga (0 i 4)?*

*šta da radim
sa ovim?*

$R \rightarrow \infty$

Čist dizajn koda I



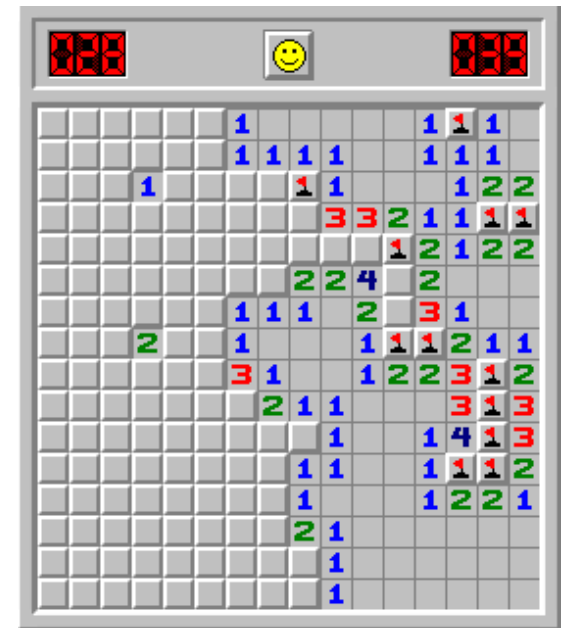
Čist kod?



Značajna
imena?

Šta radi ovaj kod?

```
public List<int[]> getFlaggedCells() {  
    List<int[]> flaggedCells = new ArrayList<int[]>();  
    for (int[] cell : gameBoard)  
        if (cell[STATUS_VALUE] == FLAGGED)  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```



Čist dizajn koda I



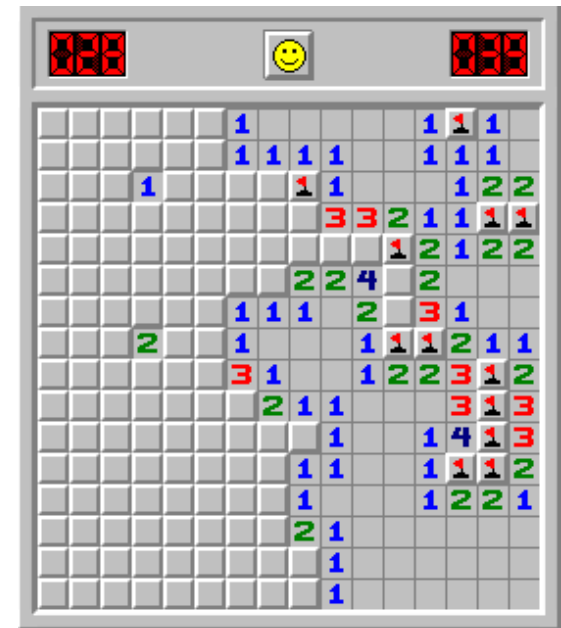
Čist kod?



Značajna
imena?

Šta radi ovaj kod?

```
public List<Cell> getFlaggedCells() {  
    List<Cell> flaggedCells = new ArrayList<Cell>();  
    for (Cell cell : gameBoard)  
        if (cell.isFlagged())  
            flaggedCells.add(cell);  
    return flaggedCells;  
}
```



Čist dizajn koda I



Čist kod?



Značajna
imena?

Refaktoriši kod

```
public static void c(char[] a1, char[] a2) {  
    for (int i = 0; i < a1.length; i++) {  
        a2[i] = a1[i];  
    }  
}
```

```
public static void copyChars(  
    char[] source, char[] destination) {  
    for (int i = 0; i < source.length; i++) {  
        destination[i] = source[i];  
    }  
}
```


Čist dizajn koda I



Čist kod?



Značajna
imena?

Optimizuj za pretragu



```
public void enroll(Course course) {  
    if(getNumOfActiveCourses() < 6) {  
        courses.add(course);  
    }  
}
```

*koje ponašanje
vidi klijent f-je?*

*šta je ovaj
magičan broj?*

*menjamo na
jednom mestu*

```
public void enroll(Course course) {  
    if(getNumOfActiveCourses() < MAX_COURSES) {  
        courses.add(course);  
    }  
}
```

*možemo da
pronađemo
gde se koristi*

Čist dizajn koda I



Čist kod?



Značajna
imena?

Optimizuj za pretragu



```
public int getNumOfActiveCourses() {  
    int c = 0;  
    for(int i = 0; i < es.length; i++)  
        if(es[i].s.equals("active"))  
            c++;  
    return c;  
}
```

*hoćemo
menjati?*

*šta ako bude potrebe da
tražimo „es“?
yes, does, goes, uses...*

Čist dizajn koda I



Čist kod?



Značajna
imena?

Optimizuj za pretragu



```
public int getNumOfActiveCourses() {  
    int numOfActiveCourses = 0;  
    for(Course c: courses)  
        if(c.isActive())  
            numOfActiveCourses++;  
    return numOfActiveCourses;  
}
```

*hoćemo
menjati?*

Čist dizajn koda I



Čist kod?



Značajna
imena?

Dužina imena

- ❖ *Scope* diktira dužinu imena

Promenljive

1-2 slova za 1-2 linije koda

(bar) 1 reč za parametre

1-2 reči za attribute klase

Više reči za globalne

Klase, Funkcije

Obrnuto

Što specifičnije to složenije

Globalne funkcije reč, dve

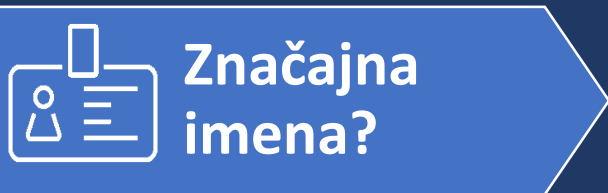
Privatne funkcije više reči

- ❖ Smernice, ne apsolutna pravila
- ❖ *Explanatory variable* krši smernice



isTestPage = *page.attributes*("Test")

Čist dizajn koda I



Kakve reči koristiti u imenima?

name



Imena klasa se sastoje od imenica
Product, WebPage, FileParser...

Imena funkcija opisuju radnju (glagoli)
buyProduct, createPage, open...

Imena dolaze iz domena rešenja
Pišemo za programere
algoritmi, šabloni, formule, strukture...

Imena dolaze iz domena problema
Dolazi od korisnika
entiteti sistema, poslovne operacije...

Čist dizajn koda I



Čist kod?



Značajna
imena?

Šta je problem sa ovim kodom?

```
public class AccountManager {  
    public Account getActiveAccount() {  
        ...  
    }  
    public Accounts getActiveAccounts() {  
        ...  
    }  
    public AccountInfo getActiveAccountInfo() {  
        ...  
    }  
}
```

*šta znači
menadžer?*

*šta je svrha
ovih funkcija?*

*moramo čitati
tela funkcija*

*moramo čitati
telo klase*

$R \rightarrow \infty$

Čist dizajn koda I



Čist kod?



Značajna
imena?

Beznačajne reči u imenima



- ❖ *Product, ProductInfo, ProductData*
- ❖ *customer, customer1, theCustomer*

„noise“ reči: *a, the, m_, 1, data, info...*

- ❖ *name, nameString, nameVariable*

redundantne reči: *variable, object, string, num...*

- ❖ *accounts, accountList*

opasne reči: *da li je ovo List objekat?*

Čist dizajn koda I



Čist kod?



Značajna
imena?

Šta izbegavati prilikom smišljanja imena?

Beznačajne reči u imenu

Ime koje zahteva mentalno mapiranje

Upotreba sinonima za isti koncept

fetch, retrieve, get...

Controller, Manager, Driver...

Being Cute

holyHandGrenade, bakeCookie...

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Šta je funkcija?

- ❖ Imenovani blok koda koji vrši neki zadatak
- ❖ Elementarni način organizacije koda

Opiši u rečenici šta radi funkcija:

- ❖ <https://pastebin.com/eCMhcNRp>

Opiši u rečenici šta radi funkcija:

```
public static String renderPageWithSetupsAndTeardowns(  
    PageData pageData, boolean isSuite) throws Exception {  
    boolean isTestPage = pageData.hasAttribute("Test");  
    if (isTestPage) {  
        WikiPage testPage = pageData.getWikiPage();  
        StringBuffer newPageContent = new StringBuffer();  
        includeSetupPages(testPage, newPageContent, isSuite);  
        newPageContent.append(pageData.getContent());  
        includeTeardownPages(testPage, newPageContent, isSuite);  
        pageData.setContent(newPageContent.toString());  
    }  
    return pageData.getHtml();  
}
```

*šta je svrha
ove funkcije?*

*šta su detalji
ove funkcije?*

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Šta čini funkciju čitljivom?

*ako moraš da gledaš
telo, nema dobro ime*

Značajno ime

Čist dizajn koda I



Čist kod?



Značajna
imena?



Čitljive
funkcije?

Funkcija treba da bude mala


❖ 3 do 6 linije koda

```
public static String renderPageWithSetupsAndTeardowns(  
    PageData pageData, boolean isSuite) throws Exception {  
    if (isTestPage(pageData))  
        includeSetupAndTearDown(pageData, isSuite);  
    return pageData.getHtml();  
}
```

Koliko linija koda u telu grananja ili petlje? 

Čist dizajn koda I

 Čist kod?

 Značajna
imena?

$f(x)$ Čitljive
funkcije?

Funkcija treba da radi jednu stvar

❖ Šta je jedna stvar?

1. *Proveri da li
je test stranica*

```
public static String renderPage(PageData pageData, boolean isSuite) throws Exception {  
    if (isTestPage(pageData))  
        includeSetupAndTearDown(pageData, isSuite);  
    return pageData.getHtml();  
}
```

2. *Uključi
dodatke*

3. *Generiši
HTML*

Čist dizajn koda I

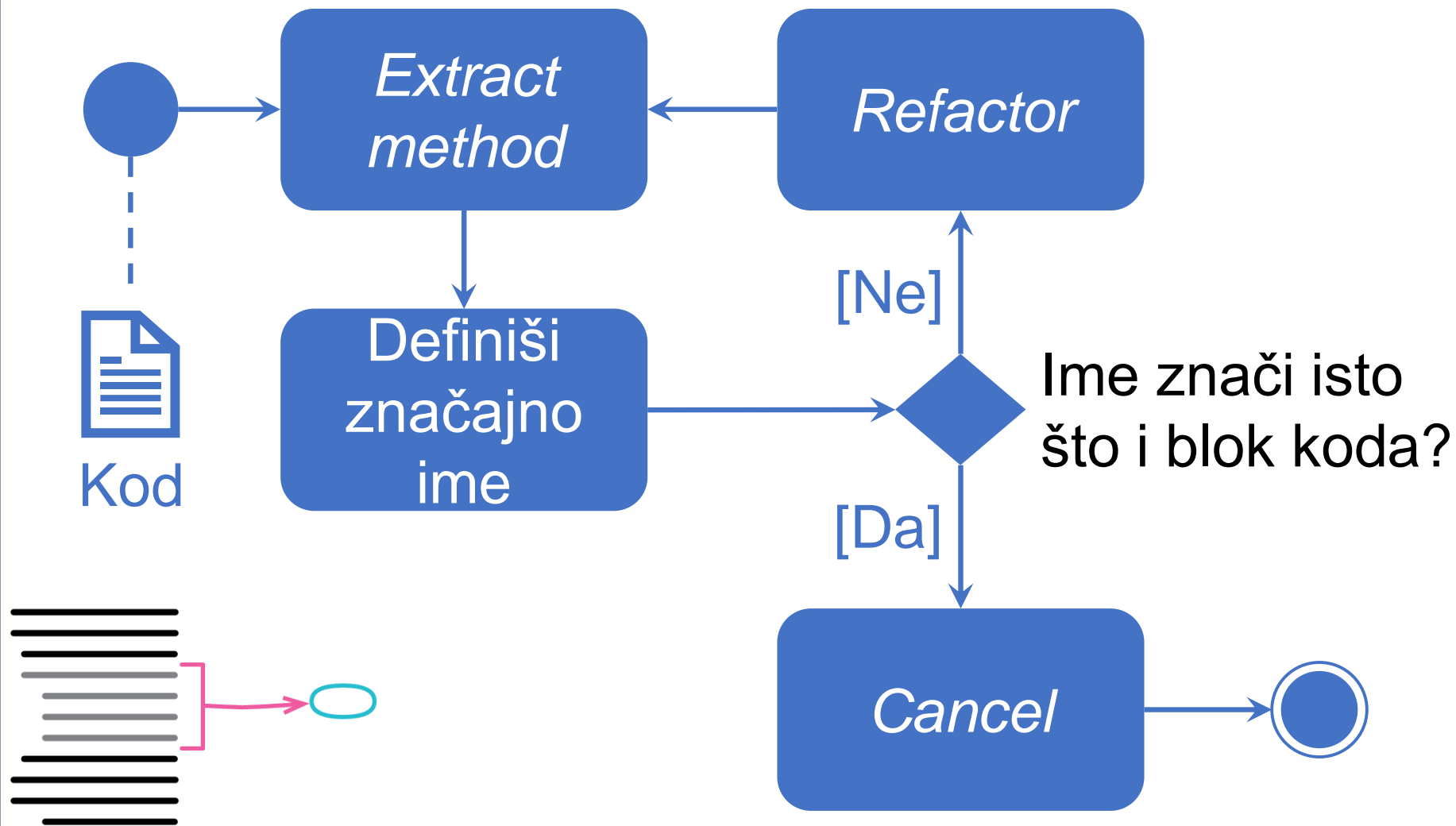
Čist kod?

Značajna
imena?

$f(x)$ Čitljive
funkcije?

Funkcija treba da radi jednu stvar

❖ Šta je jedna stvar?



Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Funkcija treba da radi jednu stvar

❖ Šta je jedna stvar?


```
public static String renderPageWithSetupsAndTearDowns(  
    PageData pageData, boolean isSuite) throws Exception {  
    if (isTestPage(pageData))  
        includeSetupAndTearDown(pageData, isSuite);  
    return pageData.getHtml();  
}
```

ime isto što i kod

```
includeSetupAndTearDownIfTestPage(pageData) {  
    if (isTestPage(pageData))  
        includeSetupAndTearDown(pageData, isSuite);  
}
```

Čist dizajn koda I

 Čist kod?

 Značajna
imena?

$f(x)$ Čitljive
funkcije?

Funkcija treba da radi jednu stvar

❖ Šta je jedna stvar?

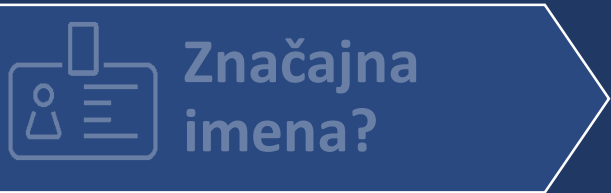
```
public int getNumOfActiveCourses() {  
    int numOfActiveCourses = 0;  
    for(Course c: courses)  
        if(c.status.equals("active"))  
            numOfActiveCourses++;  
    return numOfActiveCourses;  
}
```

šta ovo radi?

```
boolean isActive(Course c) {  
    return c.status.equals("active");  
}  
boolean isActive() {  
    return this.status.equals("active");  
}
```

šta ovo radi?

Čist dizajn koda I



Izrazi u funkciji treba da su na istom nivou
apstrakcije

```
c.status.equals("active")  
VS  
isActive(c)
```

```
userProfile.renderHtml(user)  
VS  
prettyPrint(user.biography)  
VS  
text.append("\n")
```

Mešanje apstrakcija dovodi do konfuzije

- ❖ Da li je neki detalj od posebnog značaja?

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Izrazi u funkciji treba da su na istom nivou apstrakcije

```
void saveEmployee(Employee emp) {  
    if (emp.getName() != null &&  
        emp.getSurname() != null &&  
        emp.getType() != null) {  
        ...  
    }  
}
```

```
void saveEmployee(Employee emp) {  
    if (isValid(emp)) {  
        ...  
    }  
}
```

```
void saveEmployee(Employee emp) {  
    if (isValid(emp) && emp.getAddress() != null) {  
        ...  
    }  
}
```

*što je
posebno?*

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Funkcija ne treba da ima nuspojave

- ❖ Izmena stanja (atributa, parametra...)
- ❖ Neočekivano i nedefinisano imenom

```
boolean checkPassword(String userName, String pass) {  
    User user = UserRepository.findByName(userName);  
    if (user != User.NULL) {  
        String hashPass = Crypto.hash(pass);  
        if (hashPass.equals(user.password)) {  
            Session.initialize();  
            return true;  
        }  
    }  
    return false;  
}
```

*šta je ovde
problem?*

checkPasswordAndInitializeSession(...)
Radi jednu stvar?

Čist dizajn koda I

 Čist kod?

 Značajna
imena?

$f(x)$ Čitljive
funkcije?

Funkcija ne treba da ima nuspojave

- ❖ Ili radi nešto (menja stanje)
- ❖ Ili odgovara na nešto (vraća podatak)
- ❖ *Command Query Separation*

```
public boolean set(String attribute, String value);
```

```
if(person.set("firstName", "Boban")) {  
    ...  
}
```

*šta je značenje
povratne vrednosti?*

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Funkcija treba da ima što manje parametra

- ❖ Idealno 0, 1 ili 2 su ok, 3+ izbegavati

Slučajevi korišćenja jednog parametra

Pitanje o parametru

```
isActive(course)    fileExists("myFile")
```

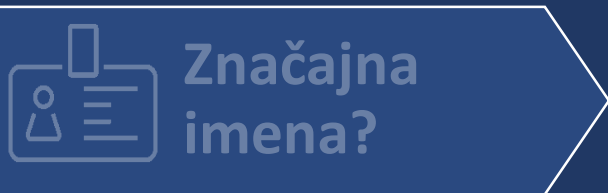
Transformacija parametra u povratnu vrednost

```
InputStream fileOpen(String fileName)
```

Obrada događaja

```
void releaseDraggedItem(Event mouseUp)
```


Čist dizajn koda I



Funkcija treba da ima što manje parametra

- ❖ Više parametra otežava čitanje

`a.equals(b)` VS `equals(a,b)`
`prettyPrint(biography, true)`

*„flag“ parametre
izbegavati*

- ❖ Mešanje redosleda više parametra

`assertEquals(msg, expected, actual)`
`writeField(outputStream, message)`

*„output“ parametre
izbegavati*

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Funkcija treba da ima što manje parametra

Strategije smanjivanja broja parametra

Definisanje funkcije kao metod klase

```
equals(a,b) > a.equals(b)
```

Definisanje parametra kao atribut klase

```
employee.giveBonus(monthlyPay, bonusLevel)
```

Grupisanje parametra u objekat ili listu

```
createEmployee(EmployeeDTO newEmployee)
```

Razdvajanje funkcije

```
prettyPrint(bio, true) > prettyPrintJSON(bio)
```

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Z8.1: Ispitaj dobre prakse
za povratnu vrednost

Šta čini funkciju čitljivom?

Mali broj
linija koda

Značajno ime

Radi ili odgovara
na jednu stvar

Minimalno
parametra

Izrazi na istom
nivou apstrakcije

Nema
nuspojava

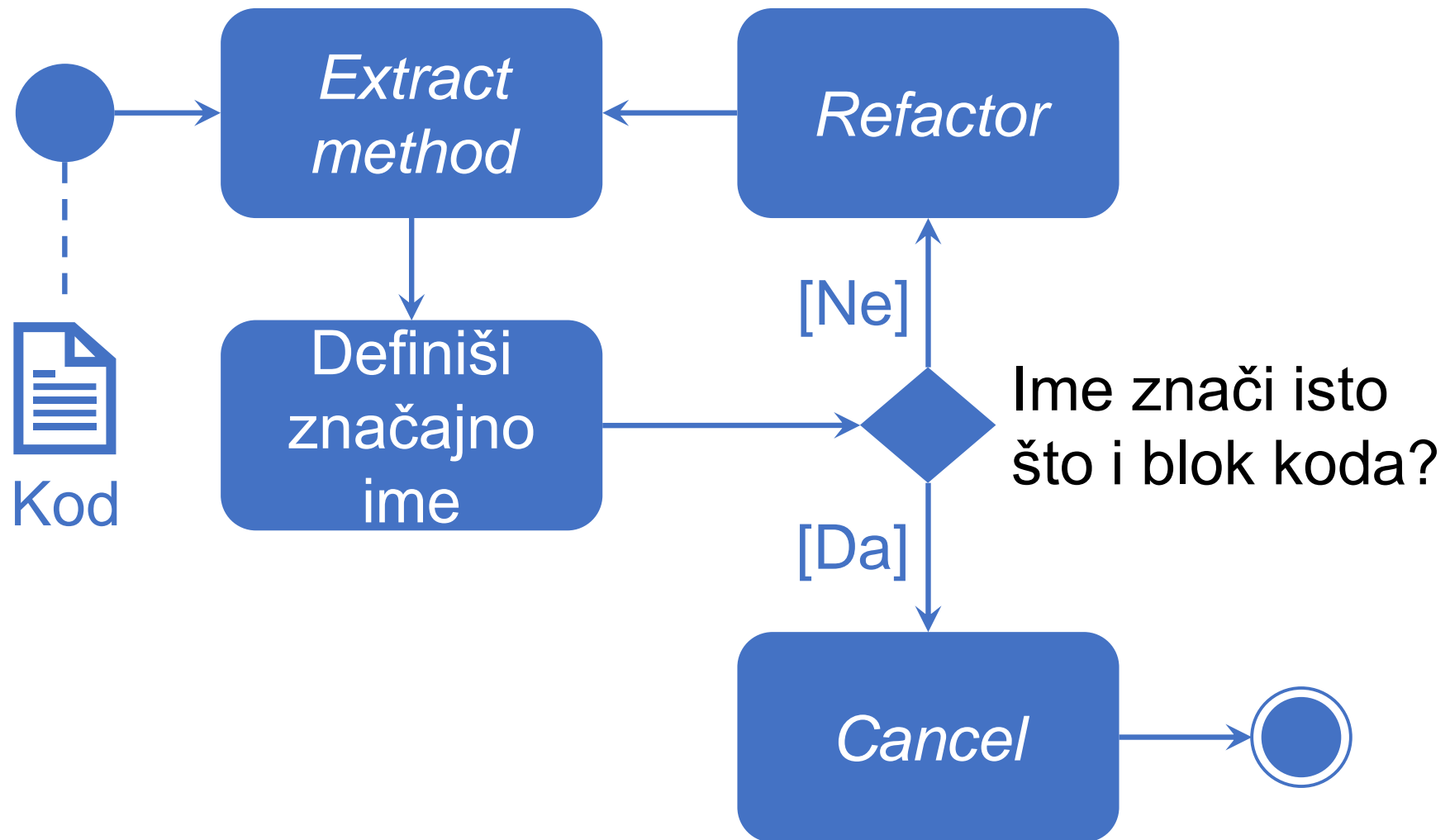
Čist dizajn koda I

Čist kod?

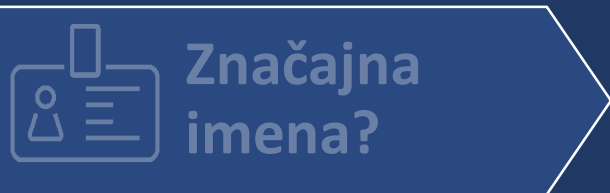
Značajna
imena?

$f(x)$ Čitljive
funkcije?

Kako pisati čitljive funkcije?

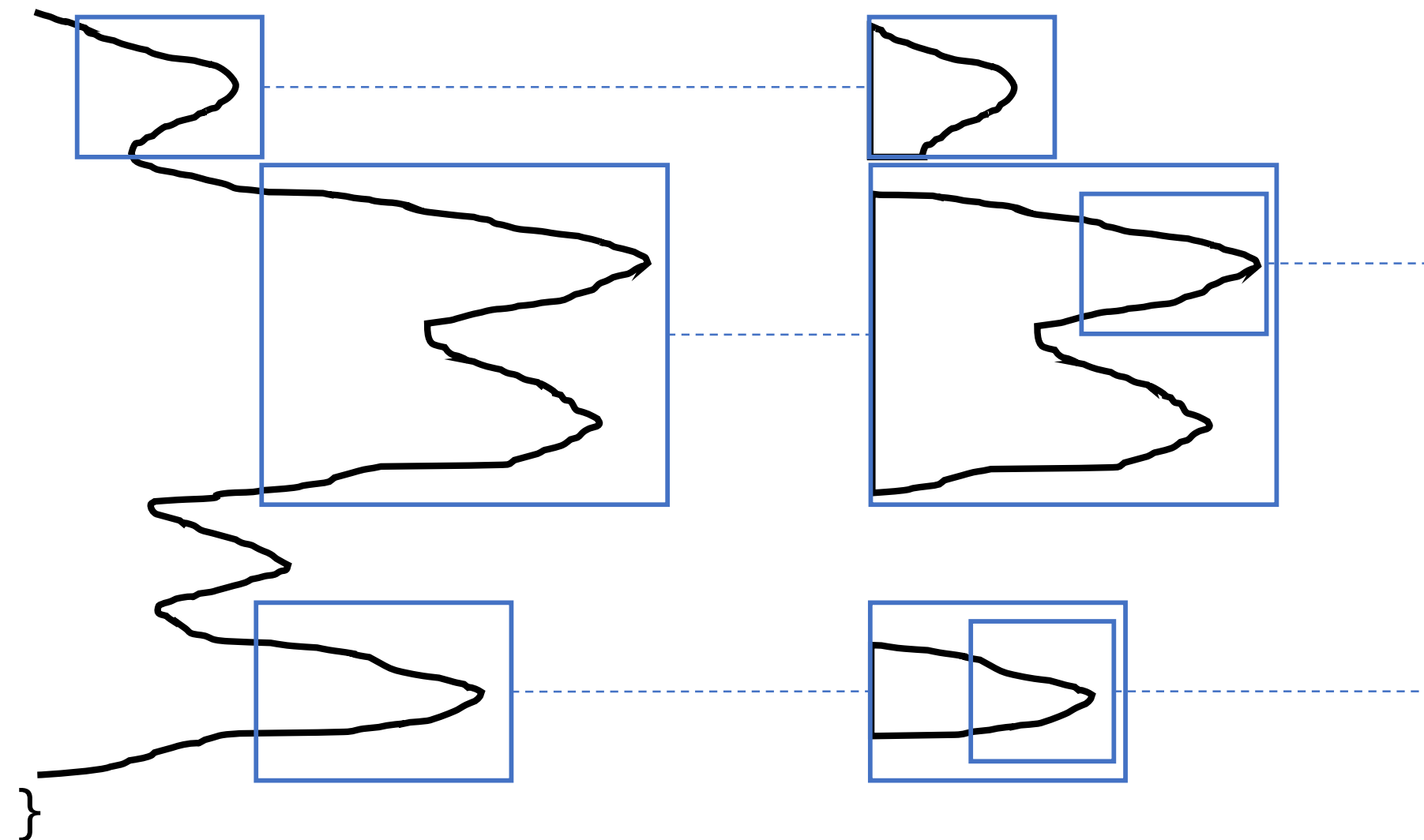


Čist dizajn koda I




Kako pisati čitljive funkcije?

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



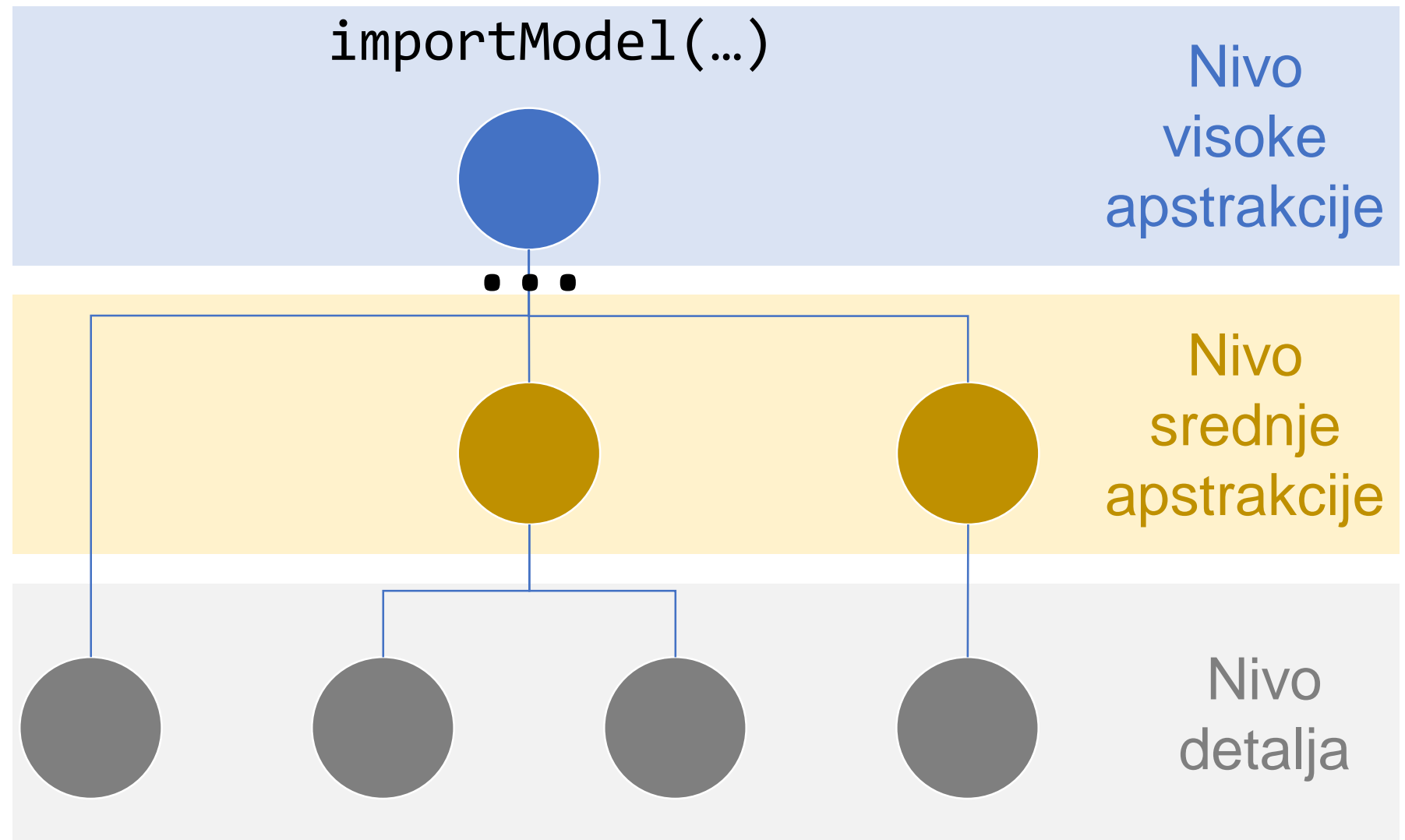
Čist dizajn koda I

 Čist kod?

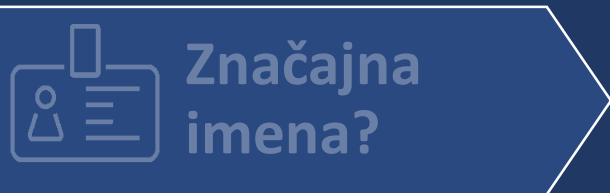
 Značajna
imena?

$f(x)$ Čitljive
funkcije?

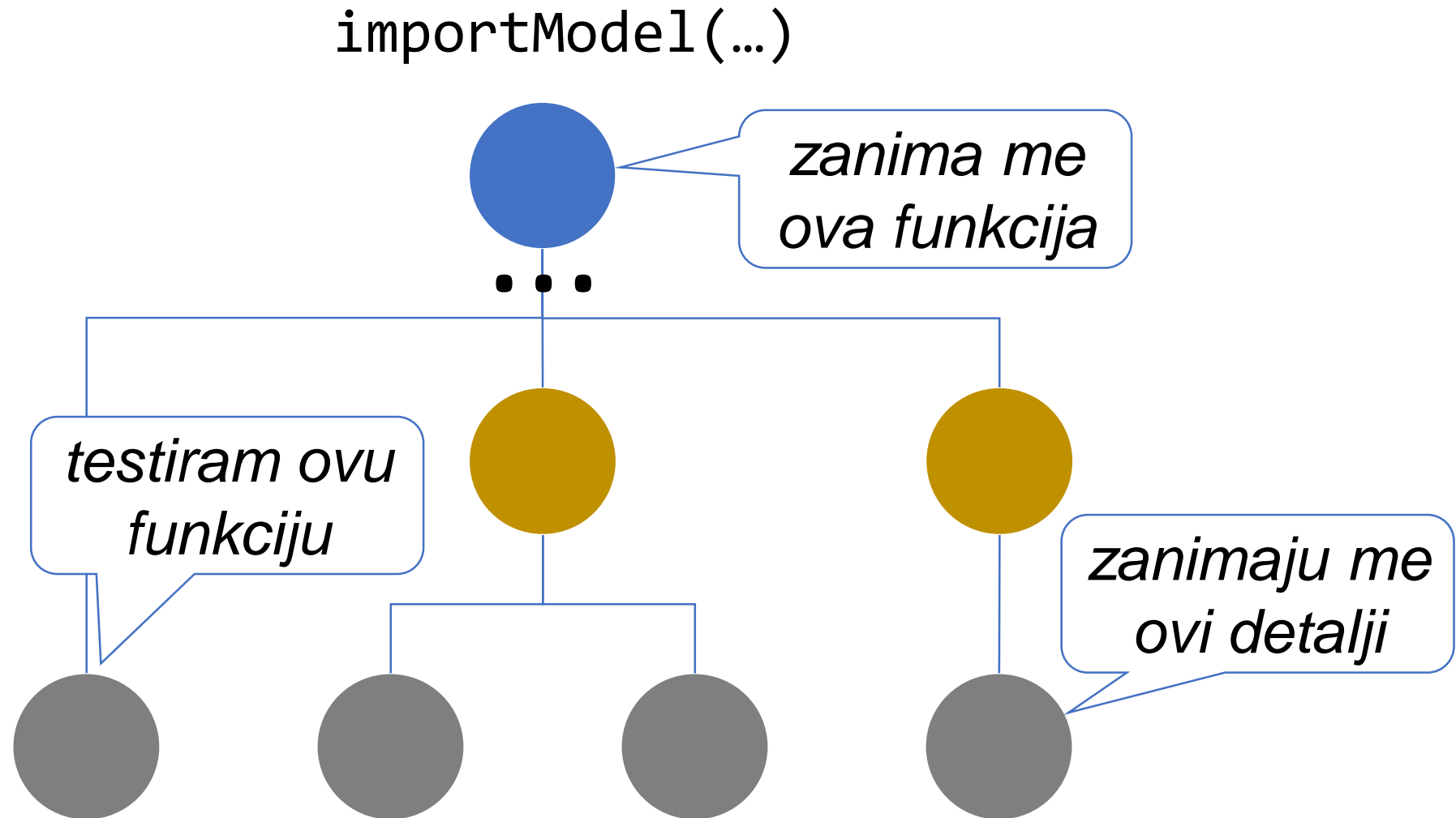
Kako pisati čitljive funkcije?



Čist dizajn koda I



Zašto pisati čitljive funkcije?





Čist kod?

Čist kod je čitljiv kod – jasan, elegantan, minimalan

$$devTime = R + W$$

Čitljiv kod $R \rightarrow 0$

Aljkav kod $R \rightarrow \infty$



Značajna imena?

Svrha

Zašto postoji

Šta radi

Kako se koristi

Struktura

Dužina

Imenice, glagoli

Rešenje, problem

Greške

Noise, cuteness

Sinonimi

Mentalni napor

$f(x)$

Čitljive funkcije?

Radi ili odgovara na jednu stvar

Nema nuspojava

Mali broj linija koda

Izrazi na istom nivou apstrakcije

Minimalno parametra

Značajno ime

Čist dizajn koda I



Čist kod?



Značajna
imena?

$f(x)$

Čitljive
funkcije?

Z8.2: Refaktoriši funkciju

❖ <https://pastebin.com/4jqS1h7m>

Z8.3: Proširi funkciju *statement*

❖ <https://pastebin.com/h4VKcREd>

❖ Podrži i štampanje HTML-a (parametar)

❖ Podrži drame i mjuzikle

❖ Refaktoriši kod da izmene budu jednostavne

Z8.4: Razmisli o aspektima čitljive funkcije