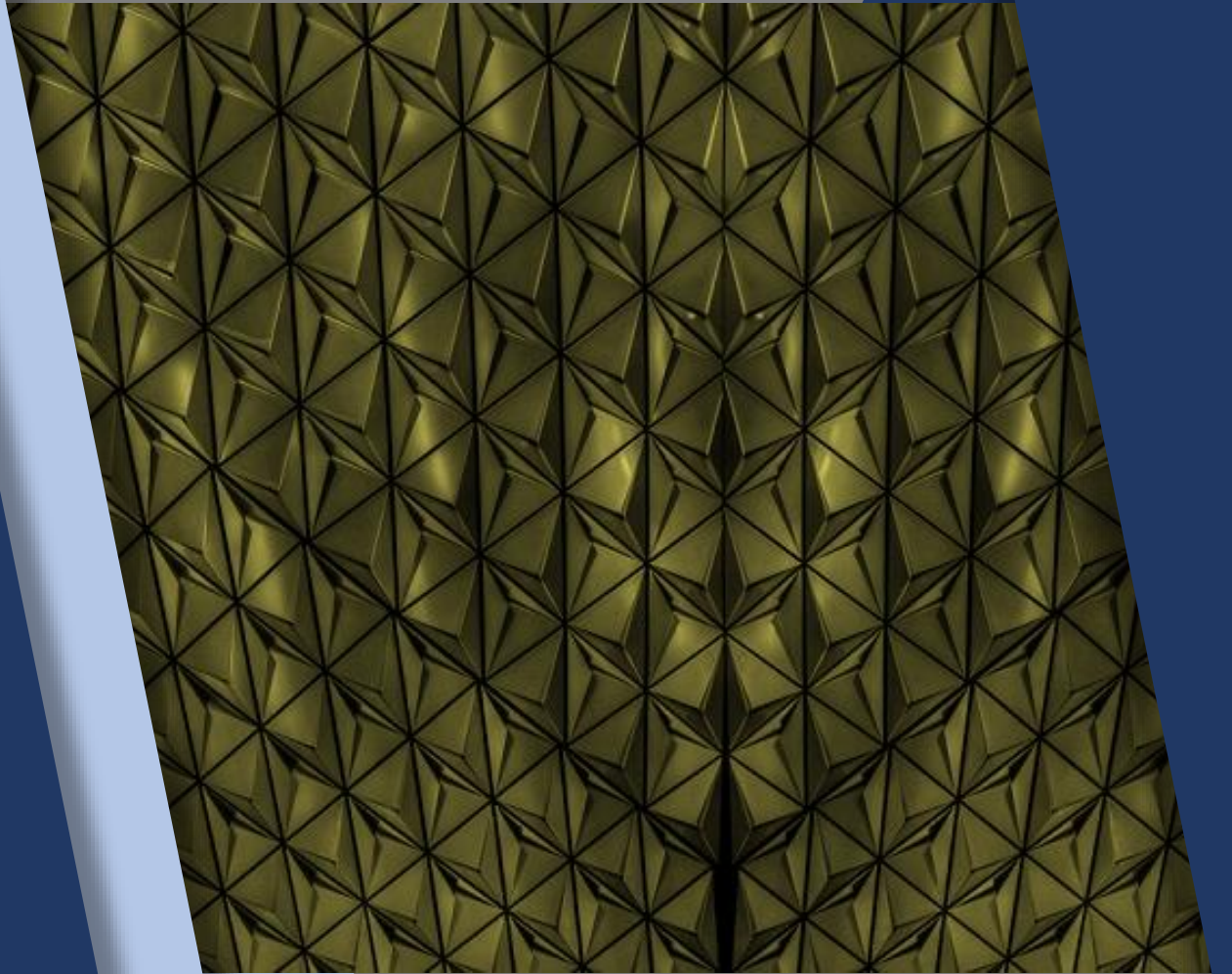


Specifikacija i modeliranje softvera

Dizajn
šabloni

Nikola Luburić
nikola.luburic@uns.ac.rs



Dizajn šabloni

*aplikacija za
praćenje zdravlja
pacijenta*

*pametni uređaji
čitaju fiziološke
podatke*

*veštačka inteligencija
pomaže u dijagnozi i
preporukama*

*pacijent unosi podatke
o svom stanju putem
formi i tastera*

*UI se prilagođava različitim
potrebama pacijenta*



Dizajn šabloni

audio taster je krupan i emituje zvuk na hover

kako izgleda dijagram klase tastera?

regular taster je kompaktan i ima tooltip

pacijent unosi podatke o svom stanju putem formi i tastera

dve teme:

- regular
- audio

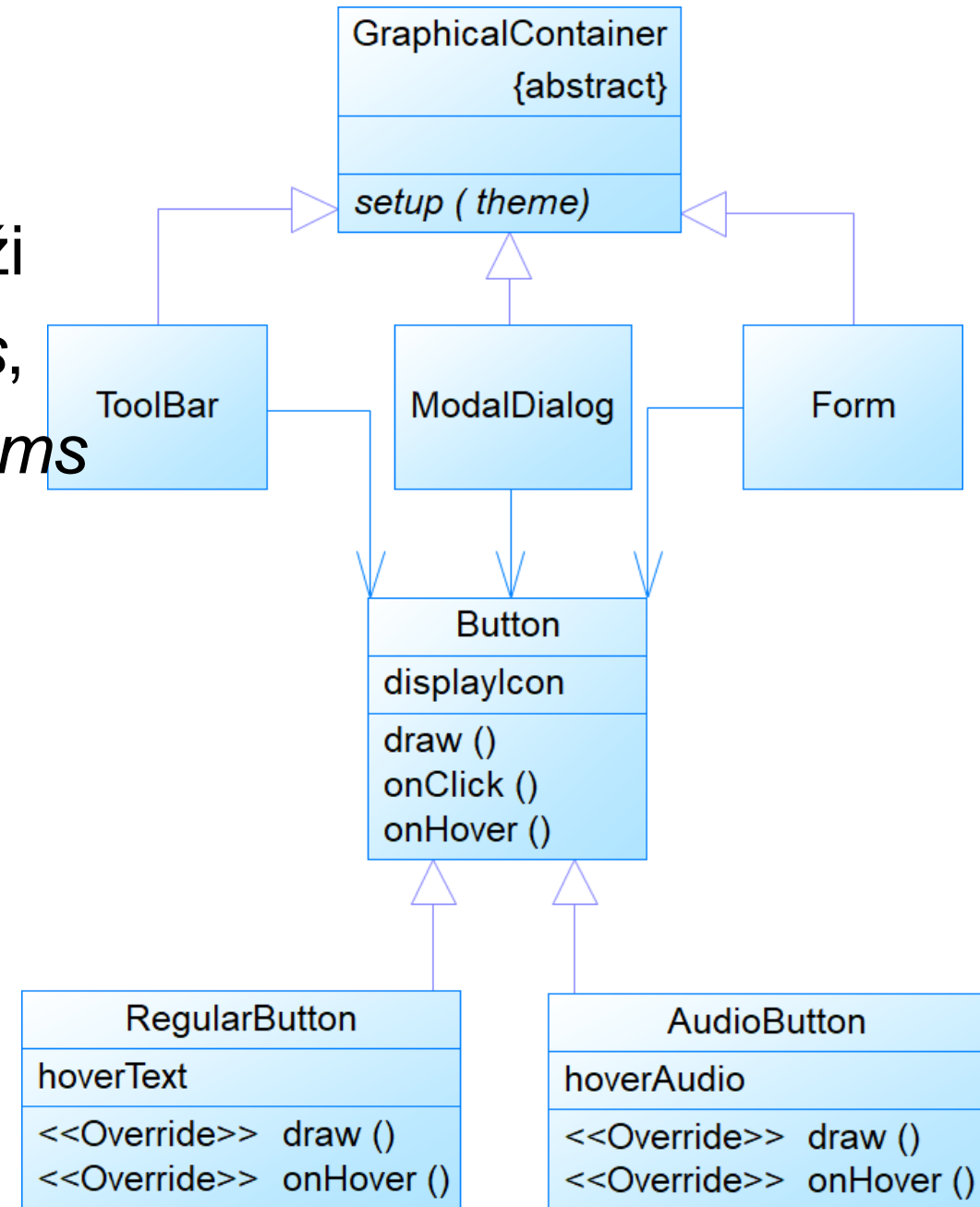
UI se prilagođava različitim potrebama pacijenta



Dizajn šabloni

Kako izgleda kod funkcije setup za:

- ❖ ToolBar koji sadrži tastere *Anamnesis*, *Activities* i *Symptoms*
- ❖ Form koji sadrži tastere *Reference activity* i *Discard*
- ❖ Vrednost theme je *regular* ili *audio*



šta se menja ako dodamo child temu?

kako da smanjimo broj promena?

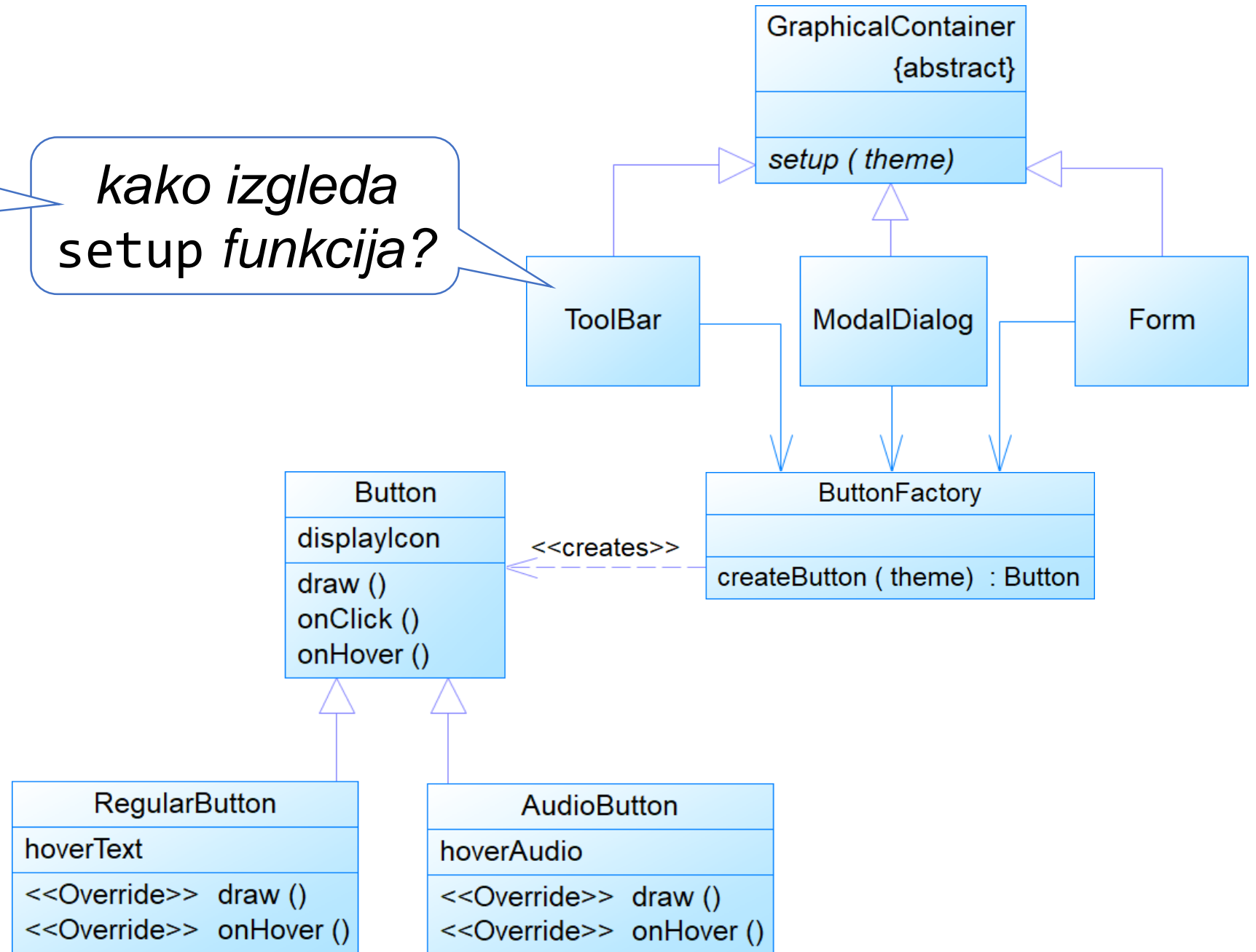
Dizajn šabloni

*šta se menja
ako dodamo
child temu?*

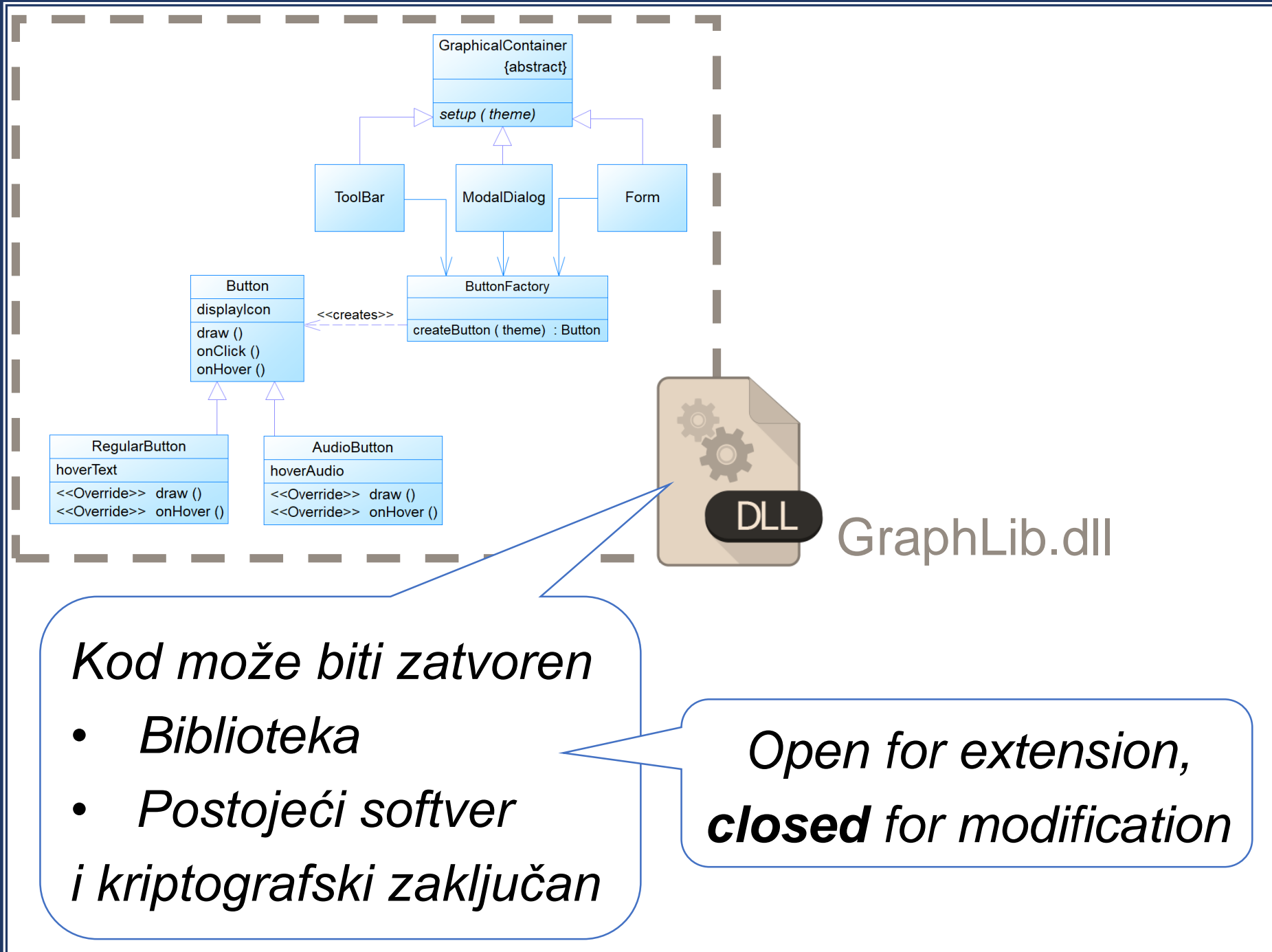
*kako izgleda
setup funkcija?*

*šta ako ne
možemo da
menjamo kod?*

*zašto ne
bismo mogli?*



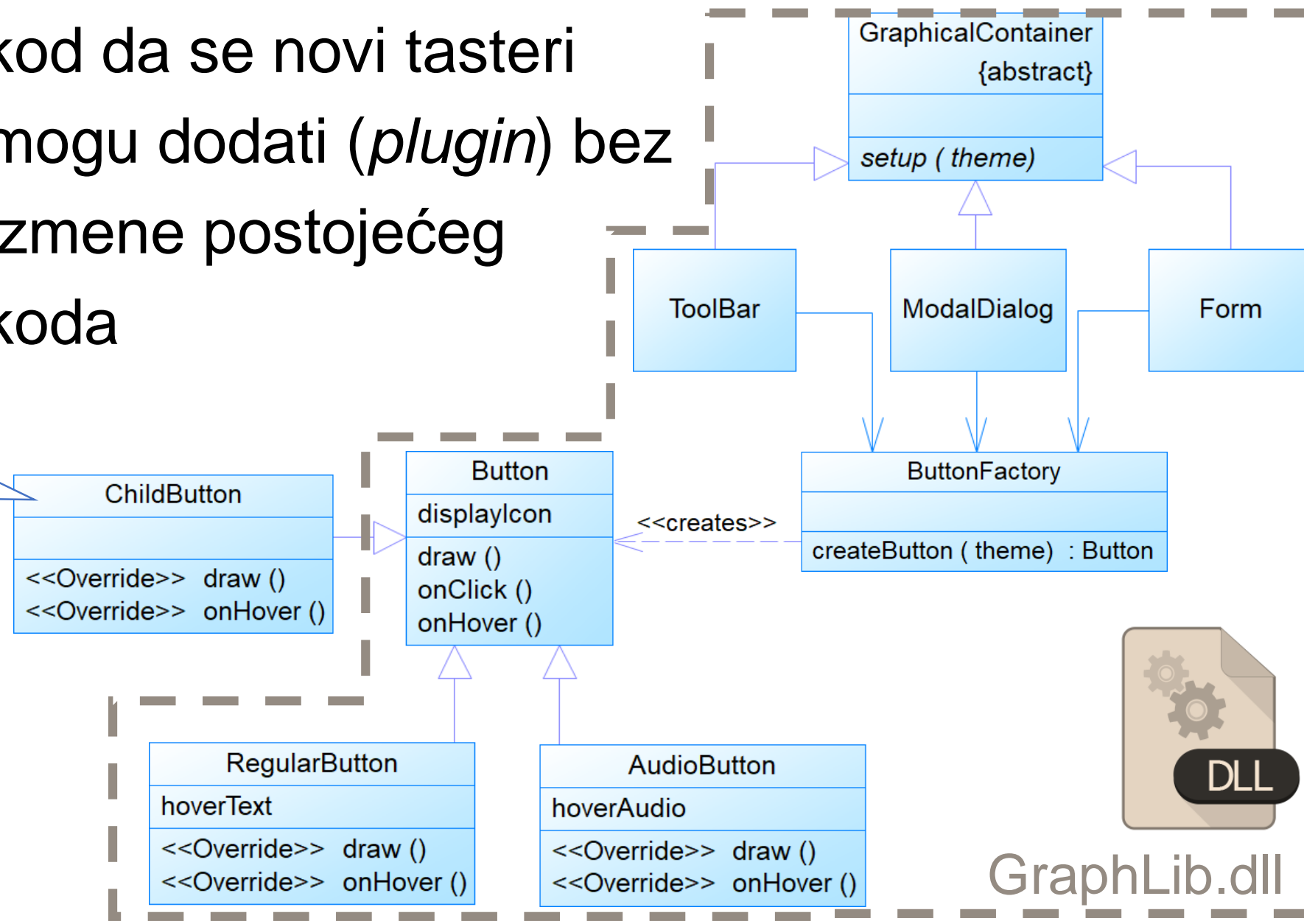
Dizajn šabloni



Dizajn šabloni

Izmeni dijagram i napiši kod da se novi tasteri mogu dodati (*plugin*) bez izmene postojećeg koda

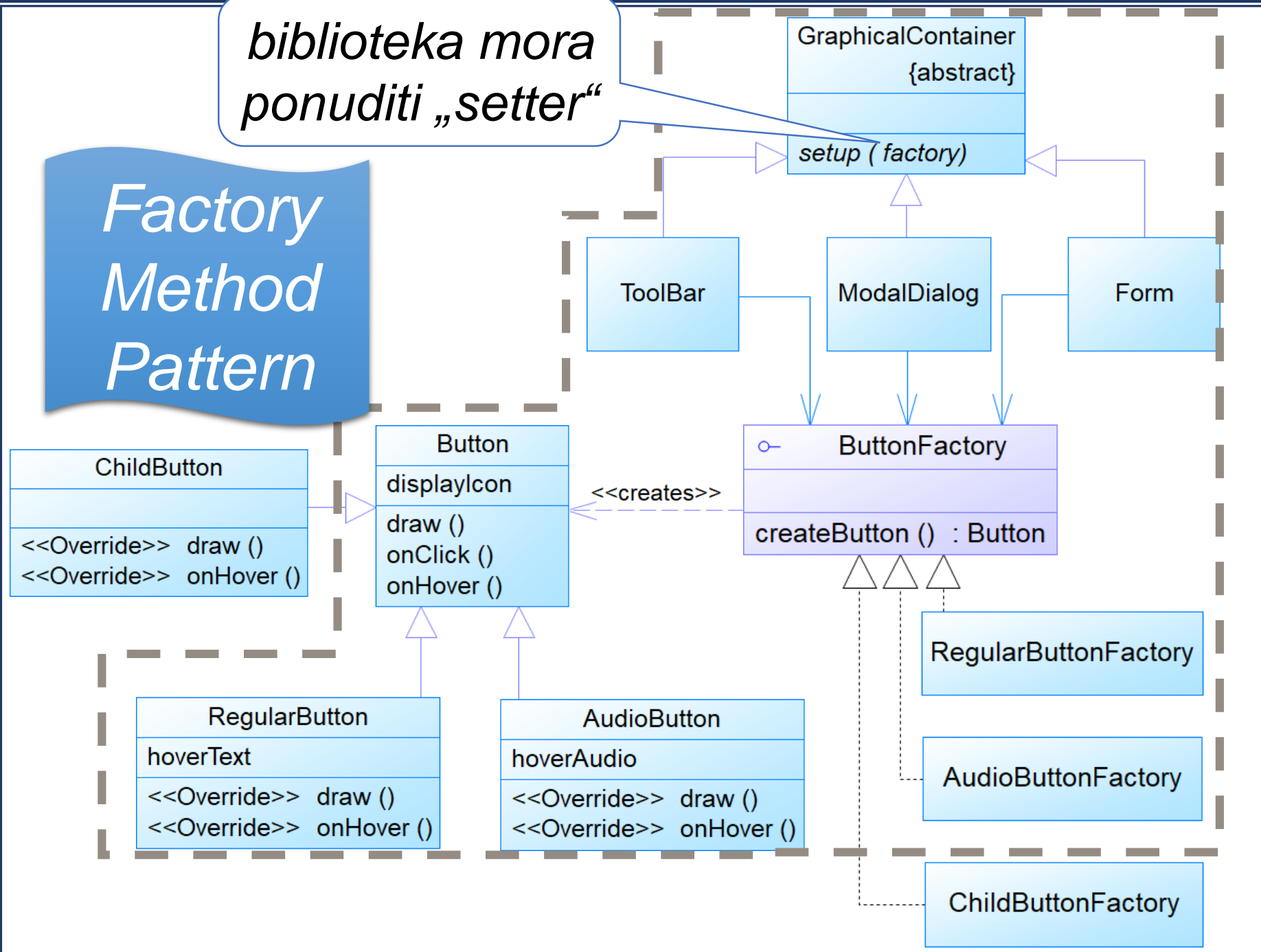
klasa koja ne pripada GraphLib.dll-u



Dizajn šabloni

Factory Method Pattern

biblioteka mora ponuditi „setter“



*šta se
desilo?*

Dizajn šabloni

Factory Method Pattern

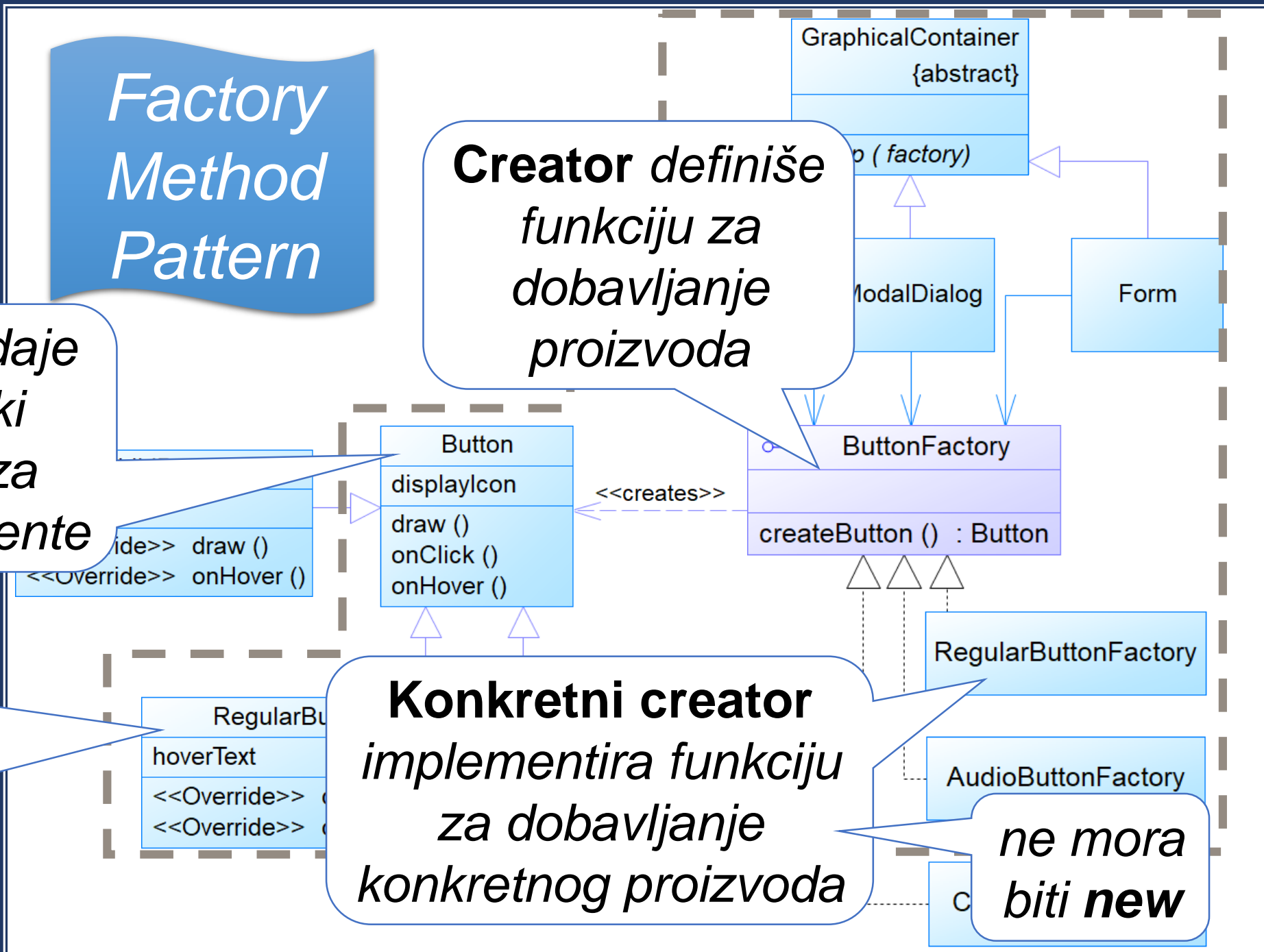
Proizvod daje zajednički interfejs za njegove klijente

Konkretni proizvodi su različite implementacije

Creator definiše funkciju za dobavljanje proizvoda

Konkretni creator implementira funkciju za dobavljanje konkretnog proizvoda

ne mora biti **new**



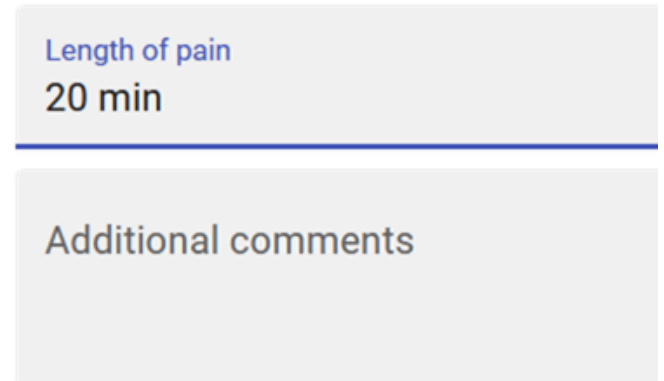
Definisanje interfejsa za kreiranje objekata, gde podklase odlučuju koji tip objekta vraćaju i kako

Dizajn šabloni

Factory M.

Nacrtaj diagram klasa po uzoru na prethodni, gde Form može da sadrži:

❖ Polja za unos podataka



A form with two input fields. The first field is labeled "Length of pain" in blue text and contains the text "20 min". The second field is labeled "Additional comments" in grey text and is currently empty. Both fields have a light grey background and a thin blue border.

❖ Kalendar



A calendar for February 2020. The header shows "FEB 2020" with a dropdown arrow, and navigation arrows for previous and next months. The days of the week are listed as S, M, T, W, T, F, S. The calendar grid shows the days of the month. The 11th is highlighted with a blue circle, and the 14th is circled with a thin blue line. The 1st is also visible in the top right corner of the grid.

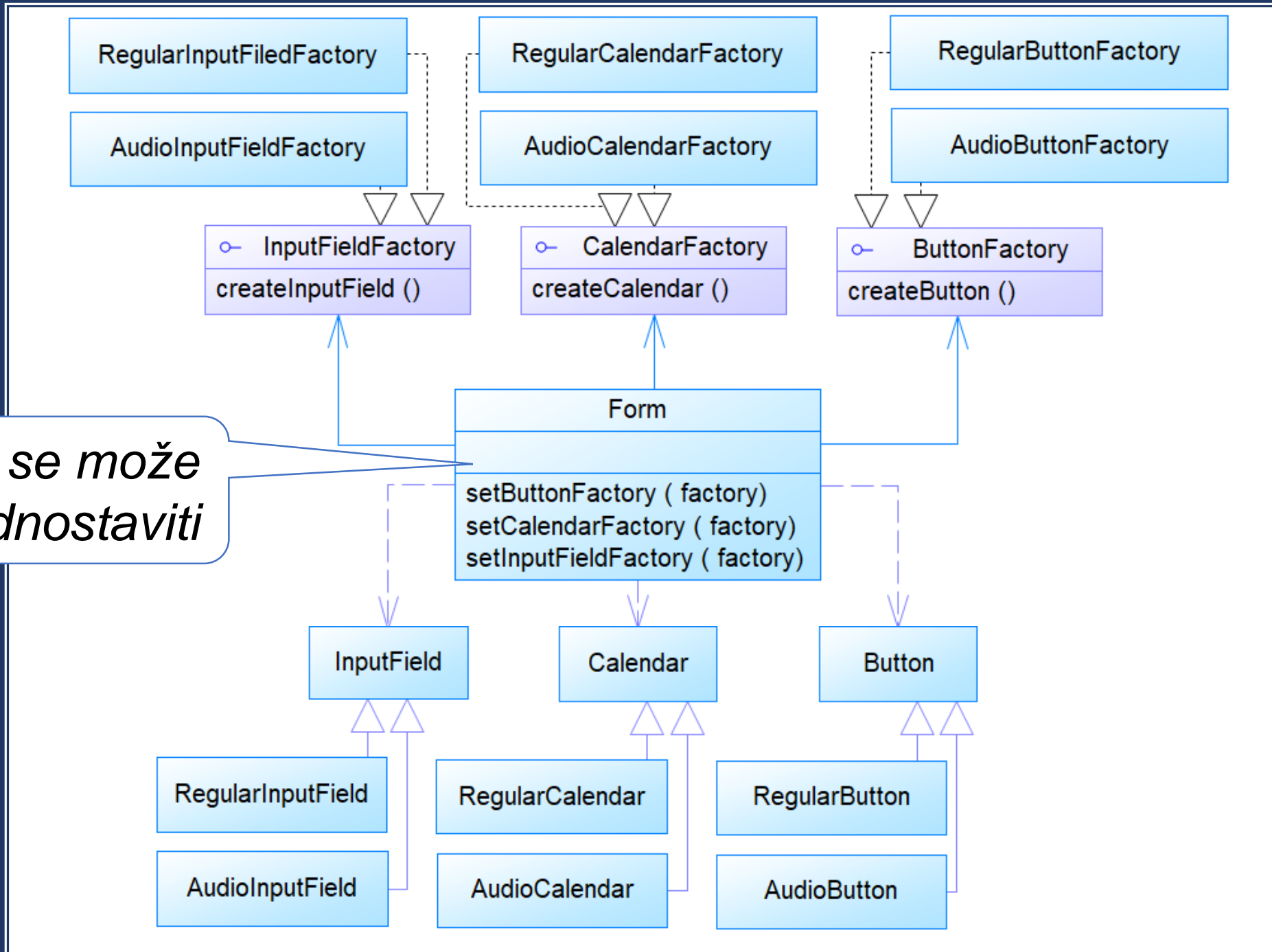
❖ Tastere

Navedeni elementi treba da se prilagode temi (*regular, audio...*)

Dizajn šabloni

Factory M.

*kako se može
pojednostaviti*



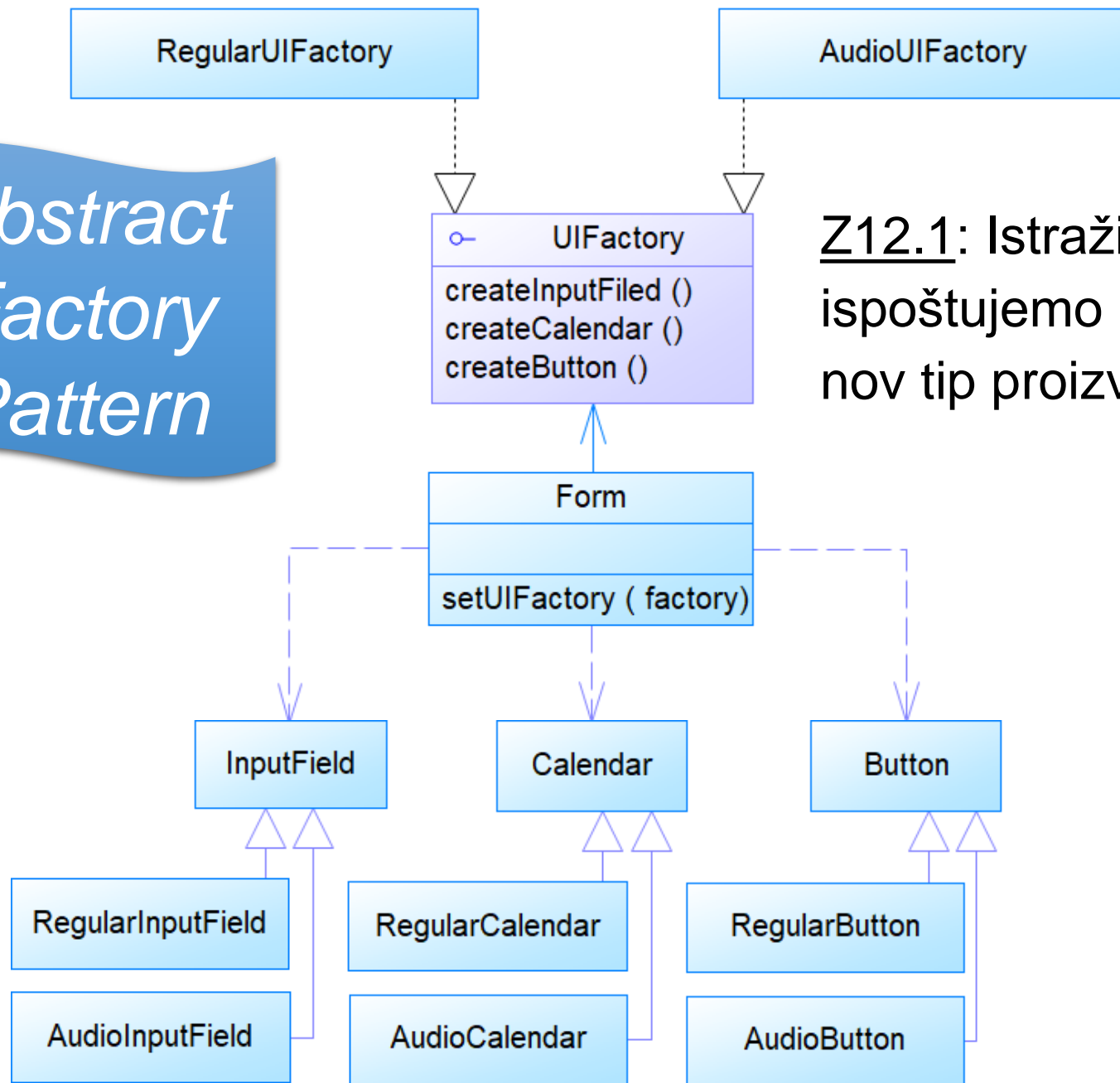
Dizajn šabloni

Factory M.

*šta menjati
da se doda
nova tema?*

*šta menjati da
se doda nov
tip proizvoda?*

Abstract Factory Pattern



Z12.1: Istraži kako da
ispoštujemo OCP za
nov tip proizvoda

Factory M.

Abstract F.

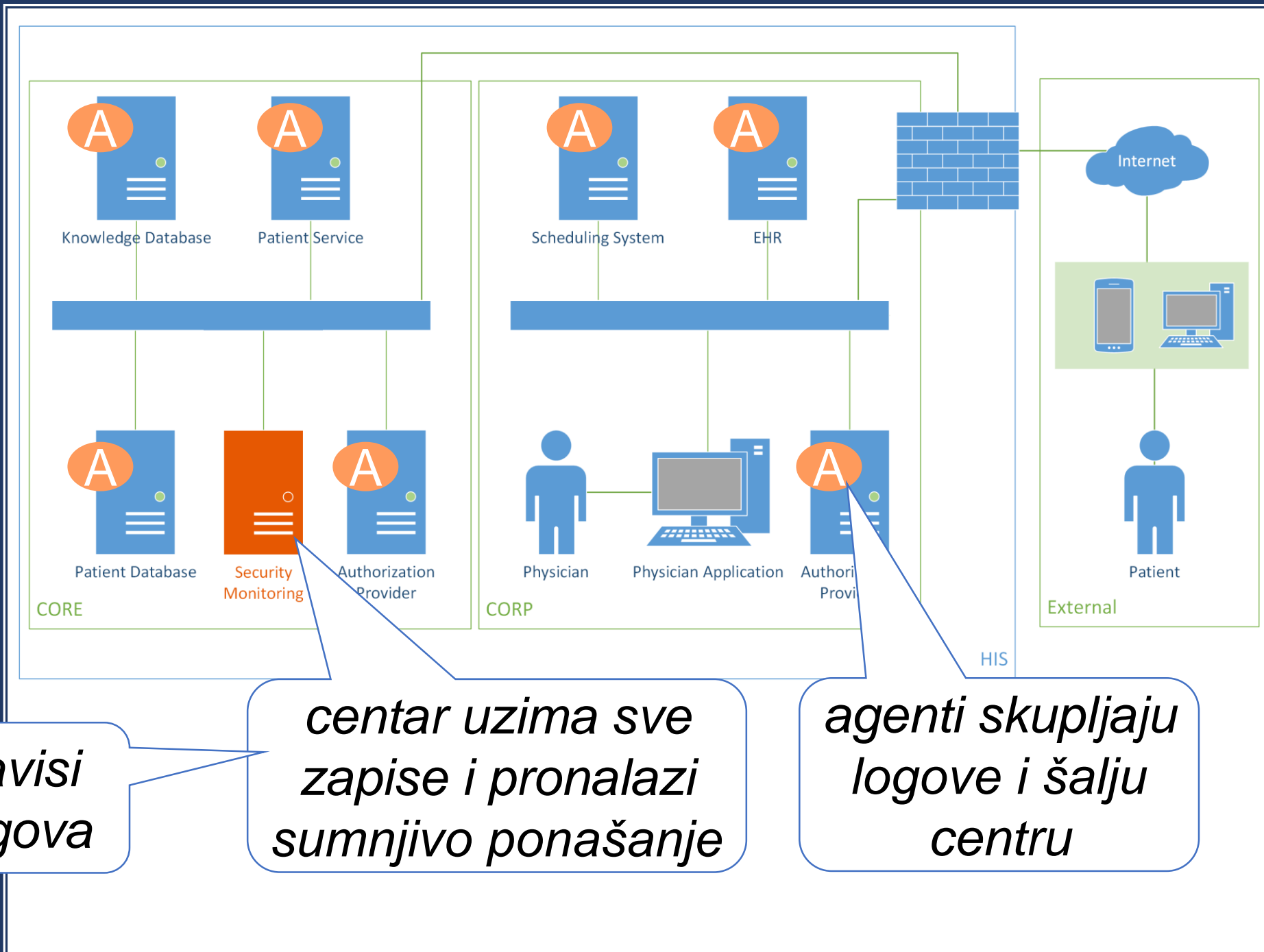
Definisanje interfejsa za kreiranje
familije objekata sličnih po
funkciji koju rešavaju u aplikaciji

Dizajn šabloni

Factory M.

Abstract F.

šta vidimo?



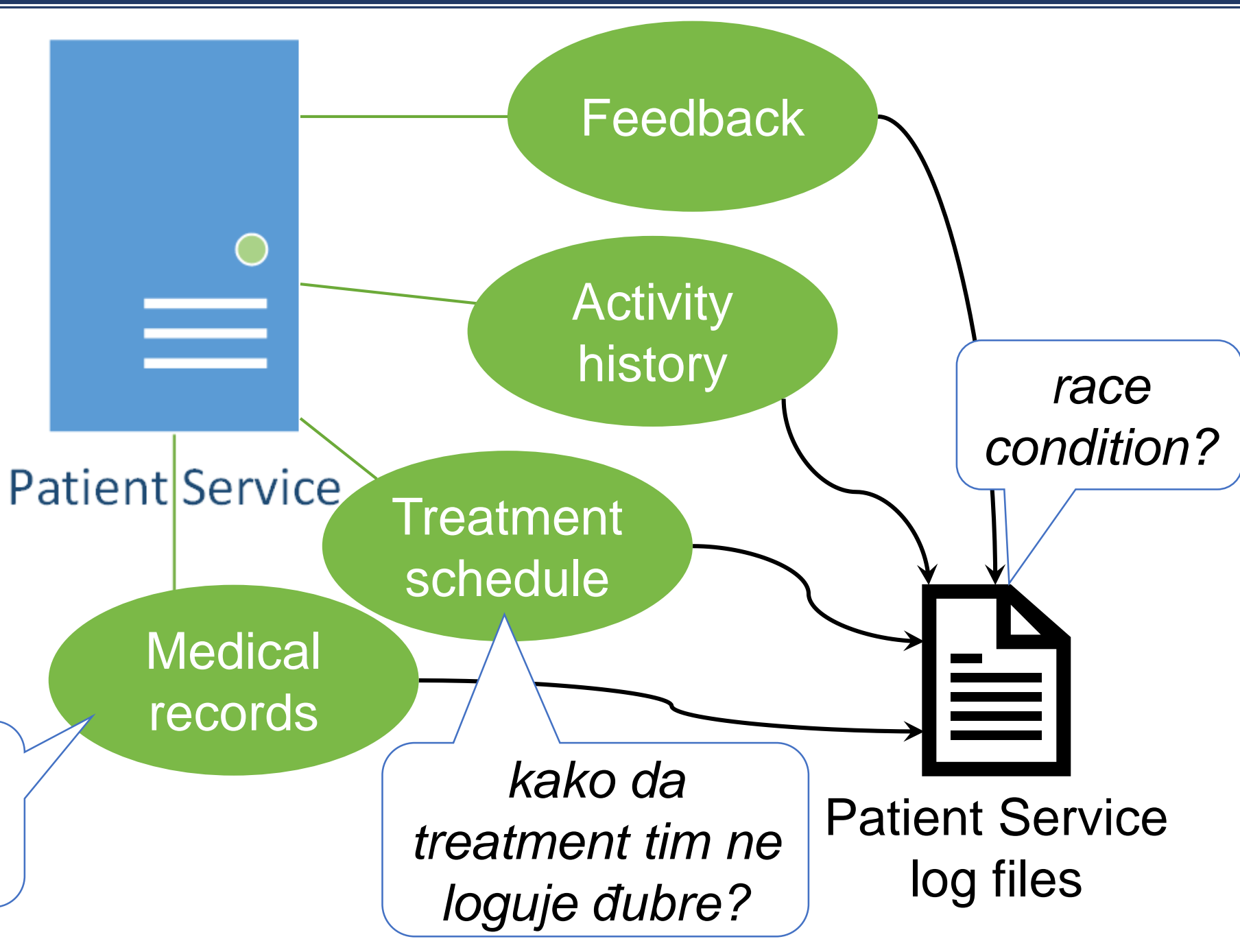
Dizajn šabloni

Factory M.

Abstract F.

šta je
feature?

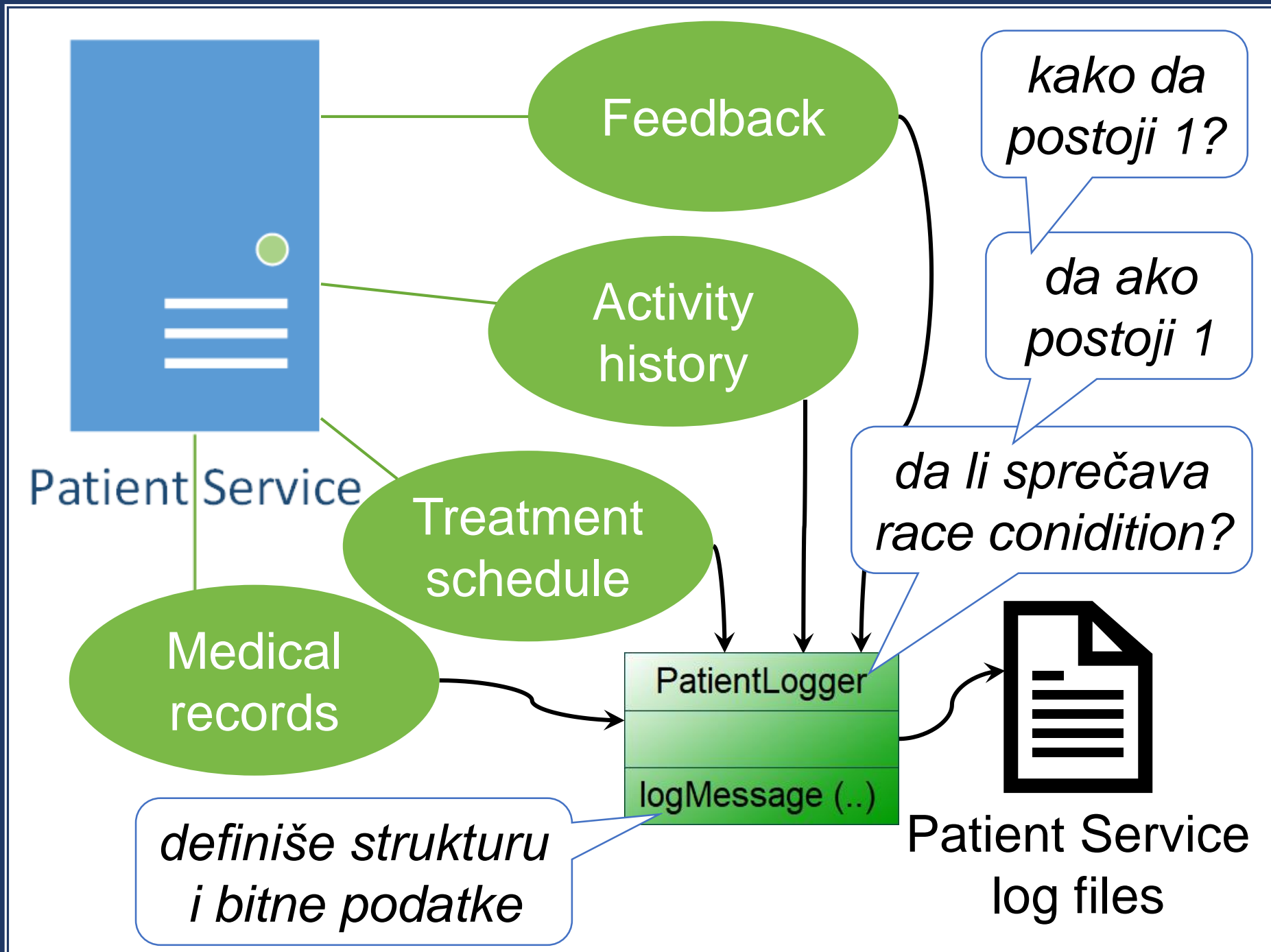
svaki feature
razvija odvojen tim



Dizajn šabloni

Factory M.

Abstract F.



Dizajn šabloni

Factory M.

Abstract F.

*šta su
problemi?*

*šta sprečava
new Logger()?*

Feedback

Activity
history

Treatment
schedule

Medical
records

PatientLogger

logMessage(..)

*kako da
postoji 1?*

```
main() {  
    PatientLogger logger = new ...  
    Feedback feedback = new  
        Feedback(logger);  
    Activities activities = new  
        Activities(logger);  
    ...  
}
```

*svaki servis prosleđuje
logger kao parametar
tamo gde treba logovati*

Dizajn šabloni

Factory M.

Abstract F.

```
class PatientLogger {
```



*kako da
postoji 1?*

```
    private PatientLogger() {}
```

*šta ovo
znači?*

```
    public static PatientLogger getInstance() {
```

```
        return new PatientLogger();
```

*kako da vrati
jednu instancu?*

```
    }
```

```
    public void logMessage(..) { ... }
```

```
}
```

Dizajn šabloni

Factory M.

Abstract F.

Singleton Pattern

```
class PatientLogger {  
    private static PatientLogger instance;  
    private PatientLogger() {}  
    public static PatientLogger getInstance() {  
        if(instance == null)  
            instance = new PatientLogger();  
        return instance;  
    }  
    public void  
}
```

race condition?

kako da postoji 1?

Eager initialization

```
    synchronized(PatientLogger.class) {  
        if (instance == null) {  
            instance = new PatientLogger();  
        }  
    }  
}
```

Lazy initialization

PatientLogger

logMessage(..)

kako da postoji 1?

Eager initialization

race condition?

Lazy initialization

Dizajn šabloni

Factory M.

Abstract F.

eager vs lazy?

Singleton Pattern

```
class PatientLogger {  
    private static PatientLogger instance =  
        new PatientLogger();  
  
    private PatientLogger() {}  
  
    public static PatientLogger getInstance() {  
        return instance;  
    }  
  
    public void logMessage(..) { ... }  
}
```

PatientLogger

logMessage(..)

*kako da
postoji 1?*

Eager initialization

Dizajn šabloni

Factory M.

Abstract F.

Singleton

*zašto je logger
ok kandidat?*

Šta su problemi?

Globalno stanje
uključuje sve što
referencira

**Sakrivanje
zavisnosti**
među klasama

**Otežano
testiranje**
kako *mock*-ovati

Z12.2: Istražiti slučajeve gde ima smisla koristiti *Singleton* i razmotriti kada je bolje koristiti statičke metode

Dizajn šabloni

Factory M.

Abstract F.

Singleton

Definisanje mehanizma za
globalni pristup jedinstvenoj
instanci klase

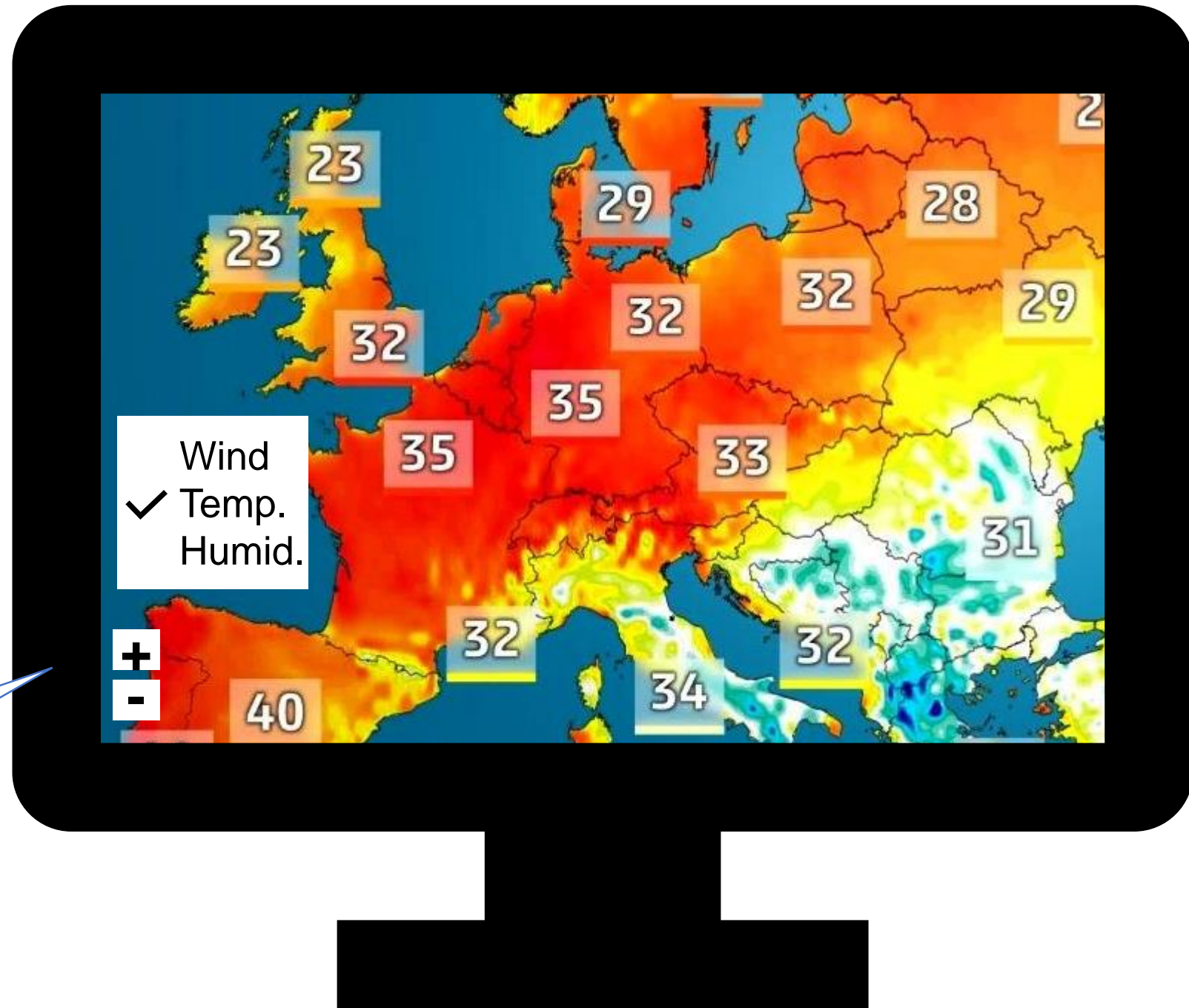
Dizajn šabloni

Factory M.

Abstract F.

Singleton

*aplikacija za
detaljni pregled
i analizu klime*

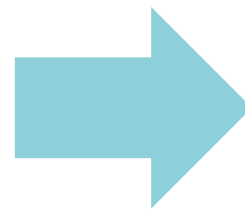


Dizajn šabloni

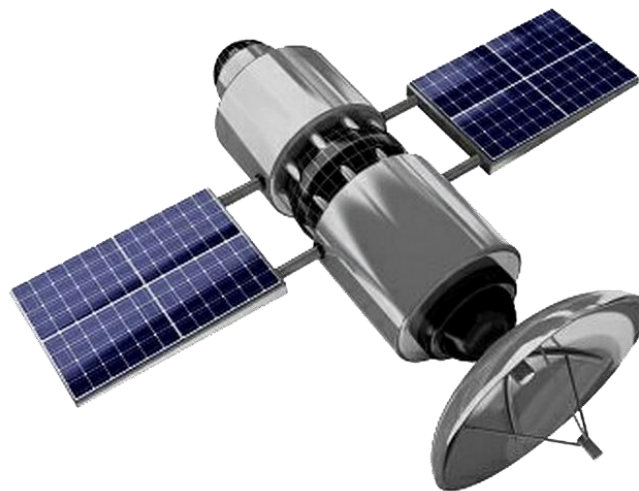
Factory M.

Abstract F.

Singleton



Geographic
Information
System

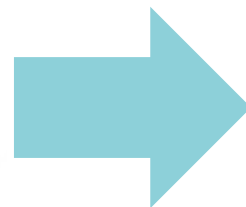


Dizajn šabloni

Factory M.

Abstract F.

Singleton



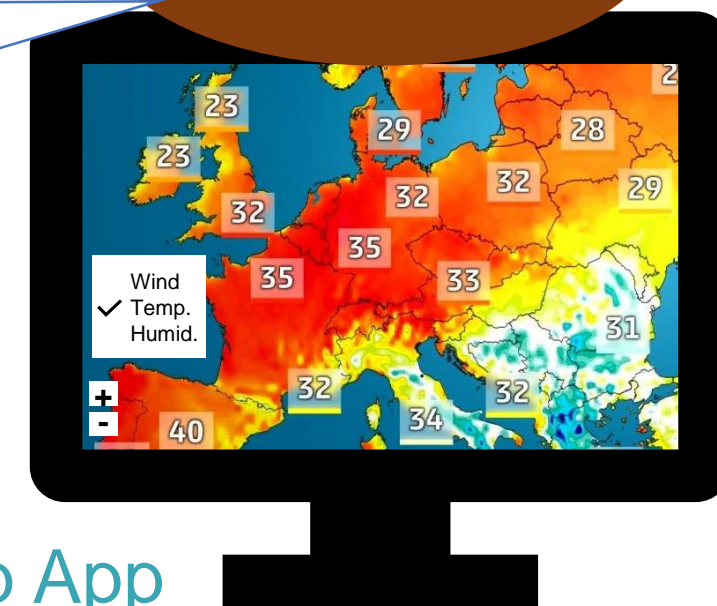
Data
Feed

Image
Feed



DrawingLib.dll

*3rd-party
biblioteka za
performantno
iscrtavanje*



Server App

Desktop App

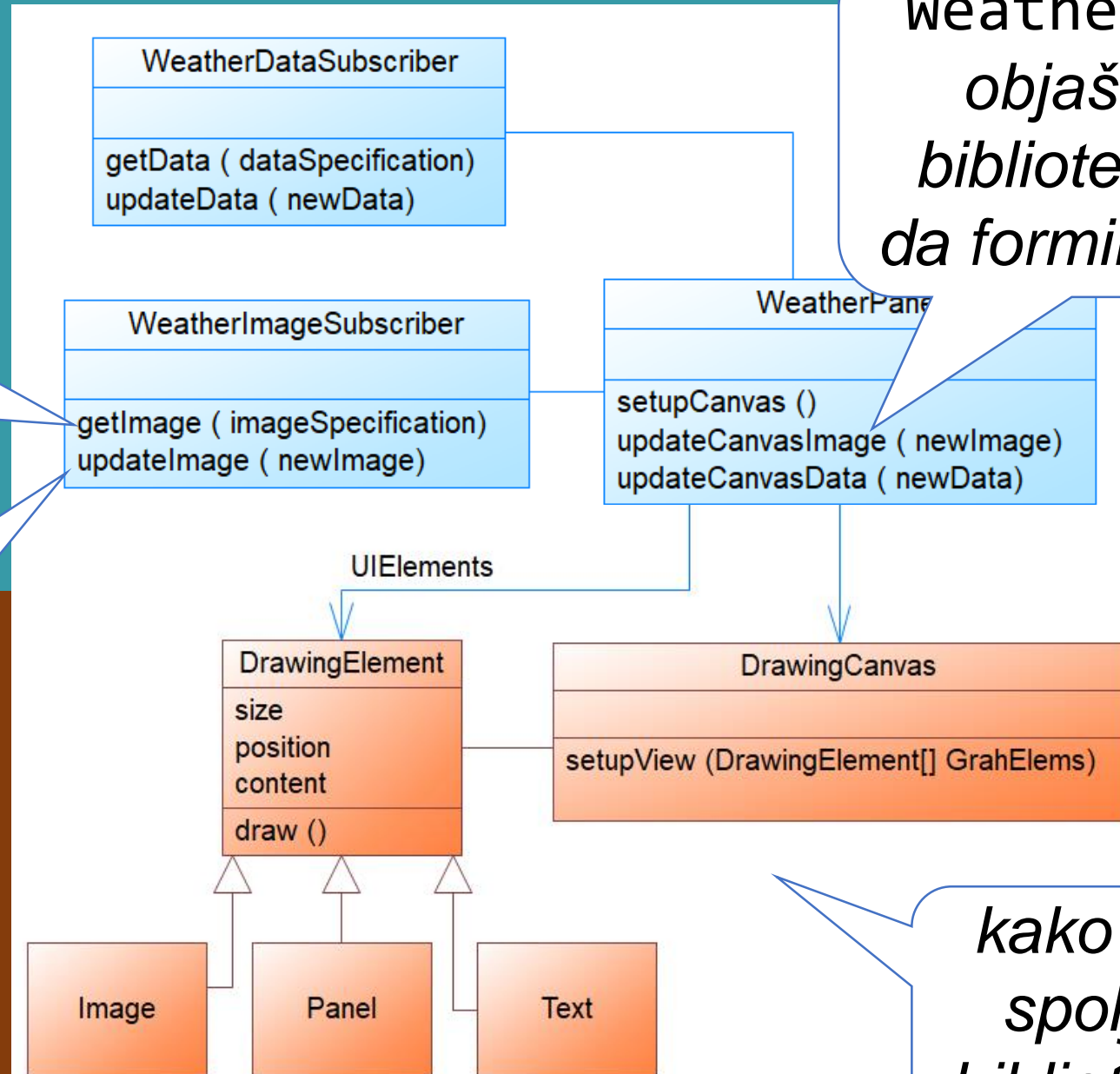
Dizajn šabloni

getImage dobavi sliku na reakciju korisnika (+,-,move...)

updateImage osveži sliku kad se spoljni podaci promene

WeatherPanel objašnjava biblioteci kako da formira prikaz

kako radi spoljna biblioteka?



Dizajn šabloni

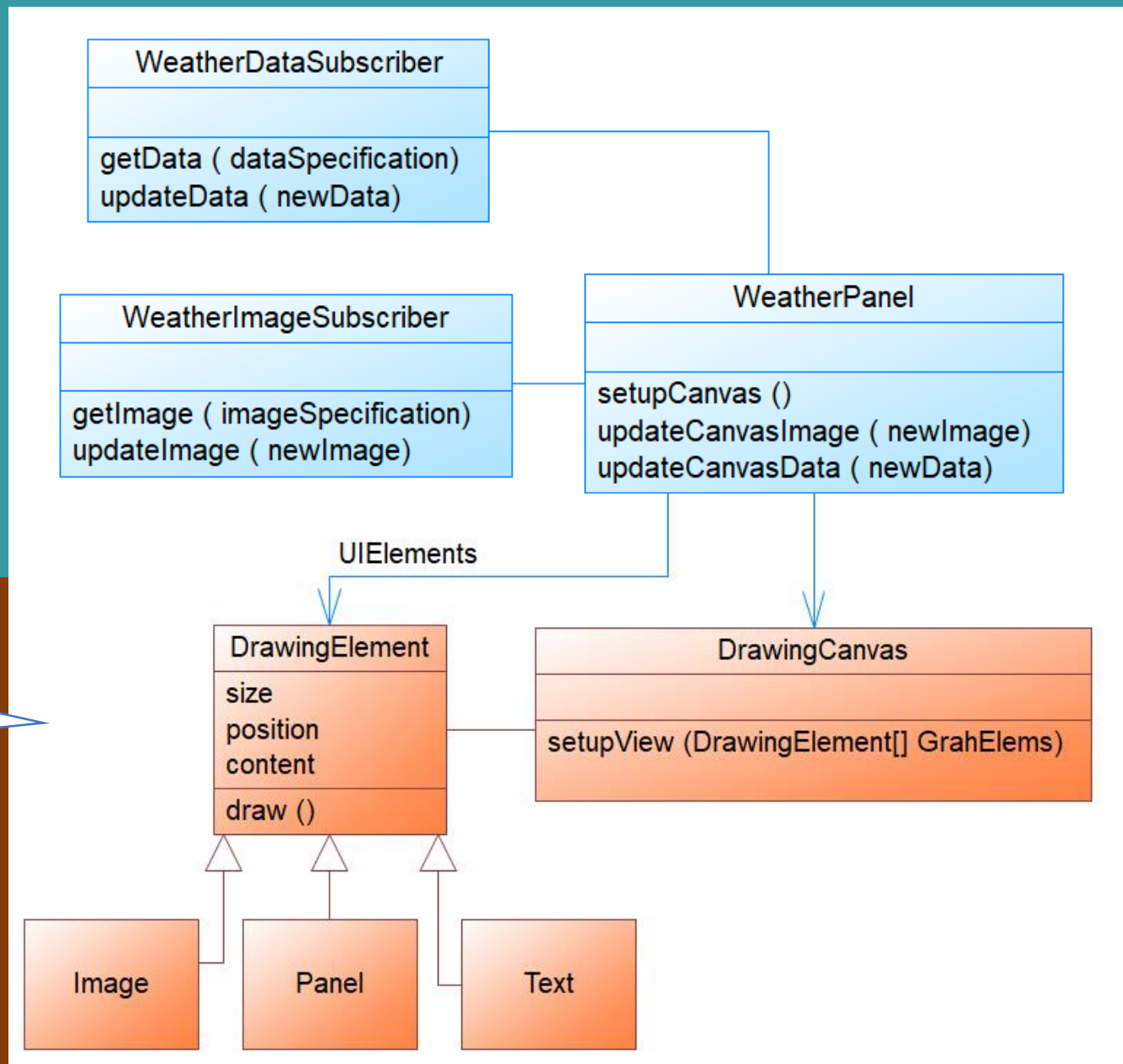
Factory M.

Abstract F.

Singleton

*DrawingLib.dll ima
ozbiljnu ranjivost*

*šta da
radimo?*



Dizajn šabloni

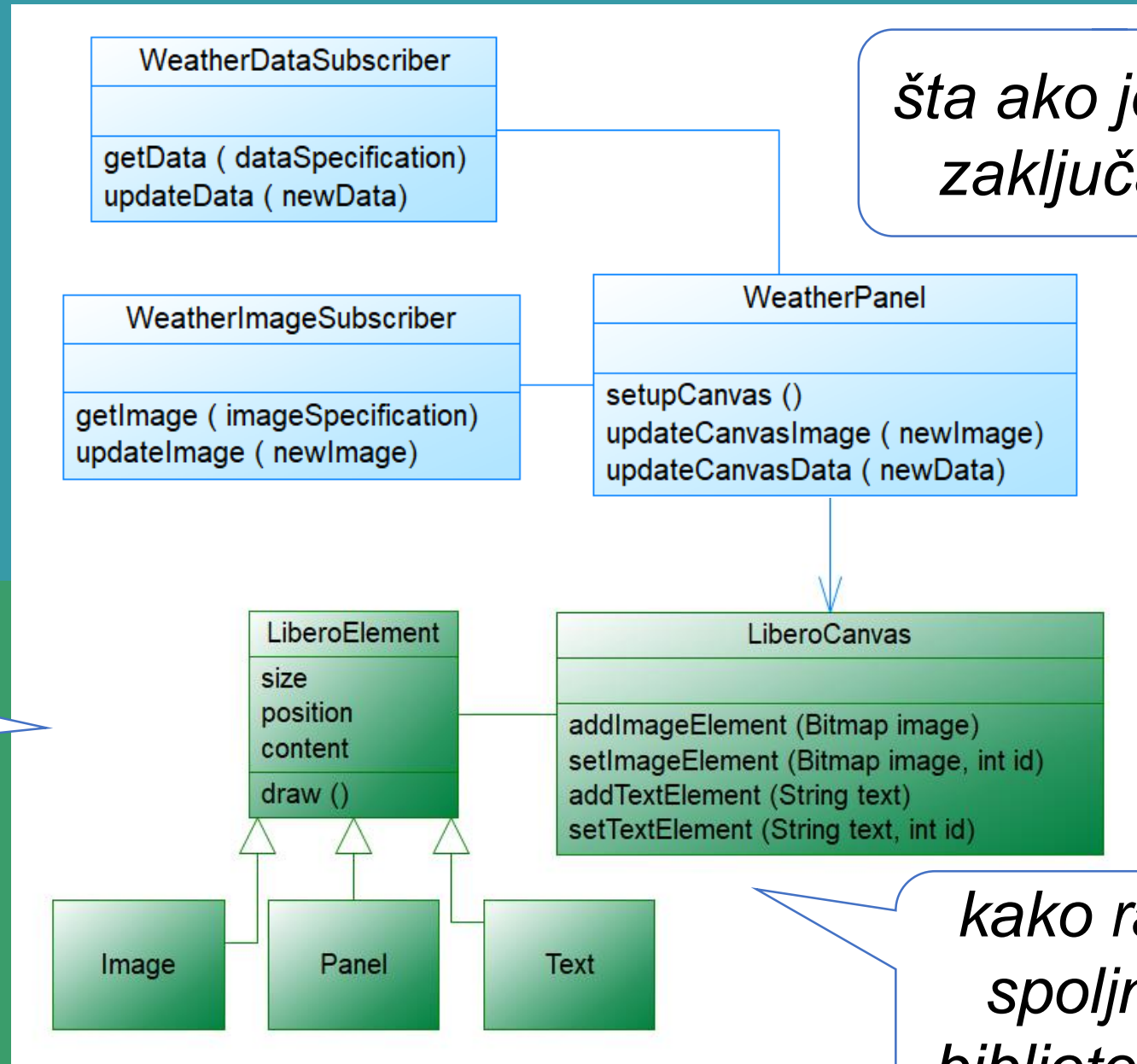
Factory M.

Abstract F.

Singleton

*LiberoDrawer.dll
ima drugi API*

*šta da
radimo?*



*šta ako je kod
zaključan?*

*kako radi
spoljna
biblioteka?*

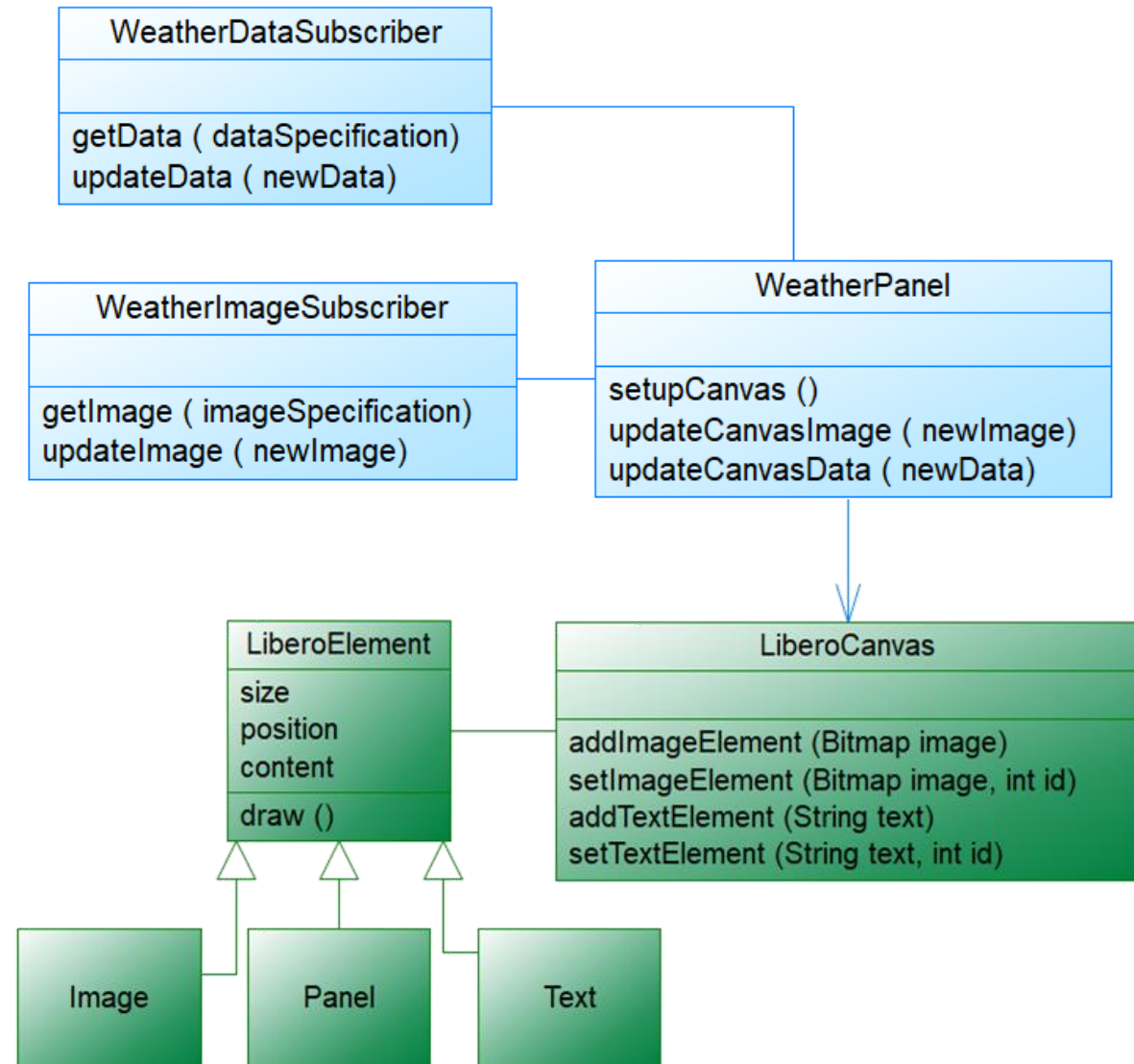
Dizajn šabloni

Factory M.

Abstract F.

Singleton

Dizajniraj rešenje za problem zamene biblioteke, takvo da je moguće integrisati kod sa drugim API-em i u slučaju da se WeatherPanel naknadno zaključa



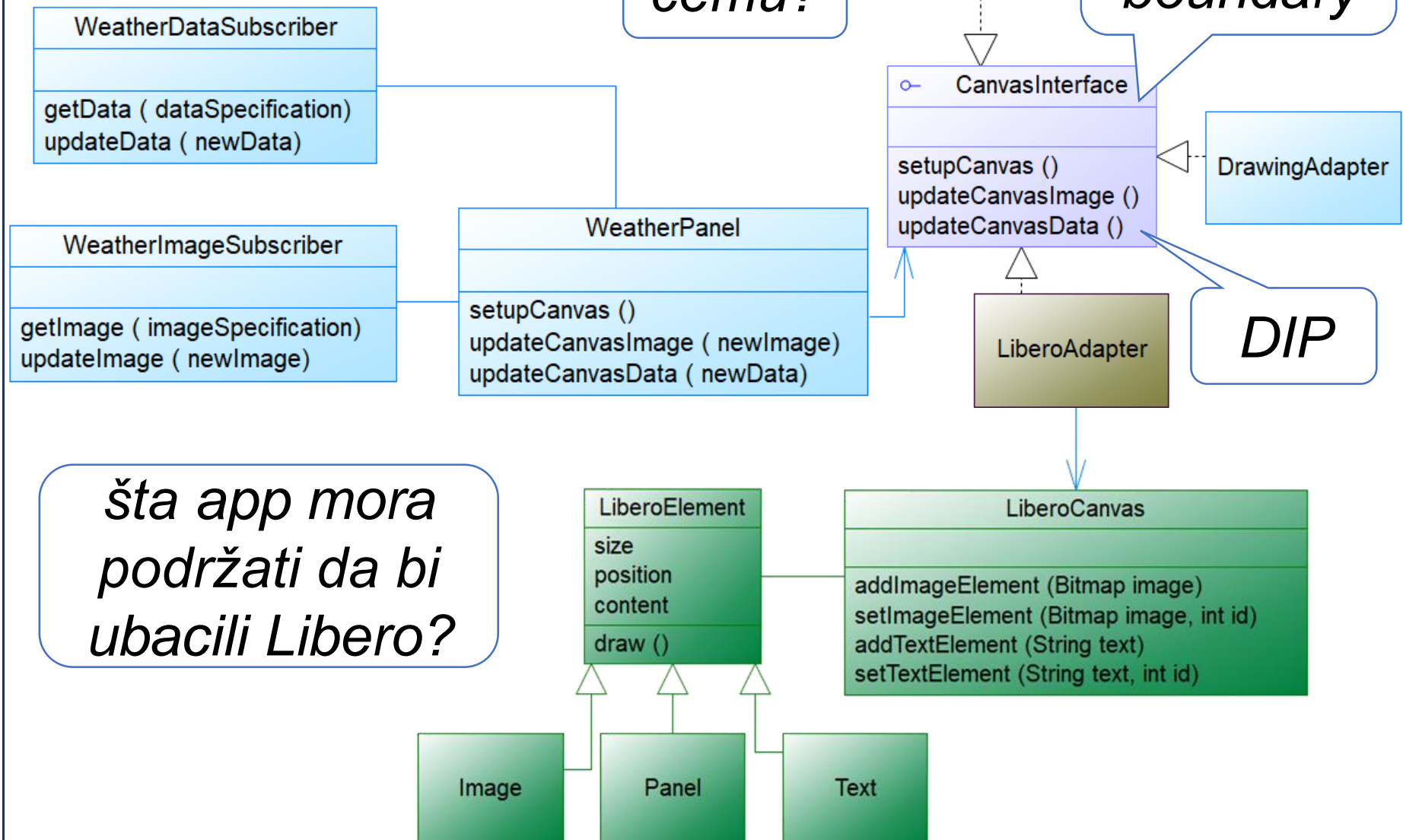
Dizajn šabloni

Factory M.

Abstract F.

Singleton

Adapter Pattern



Dizajn šabloni

Factory M.

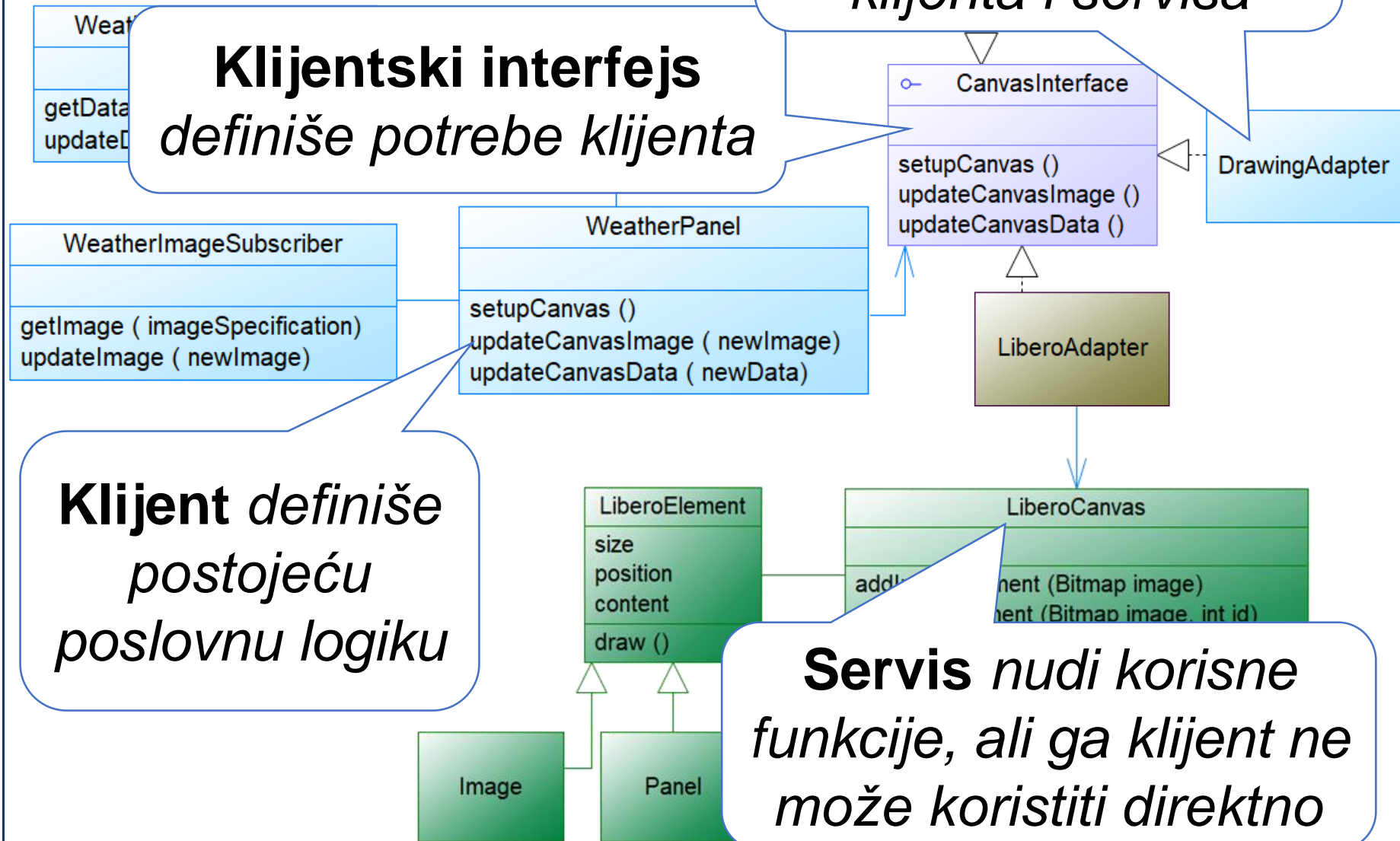
Abstract F.

Singleton

Adapter Pattern

Klijentski interfejs
definiše potrebe klijenta

Adapter *translira komunikaciju između klijenta i servisa*



Klijent *definiše postojeću poslovnu logiku*

Servis *nudi korisne funkcije, ali ga klijent ne može koristiti direktno*

Dizajn šabloni

Factory M.

Abstract F.

Singleton

Adapter

Definisanje mehanizma
kolaboracije objekata sa
nekompatibilnim interfejsima