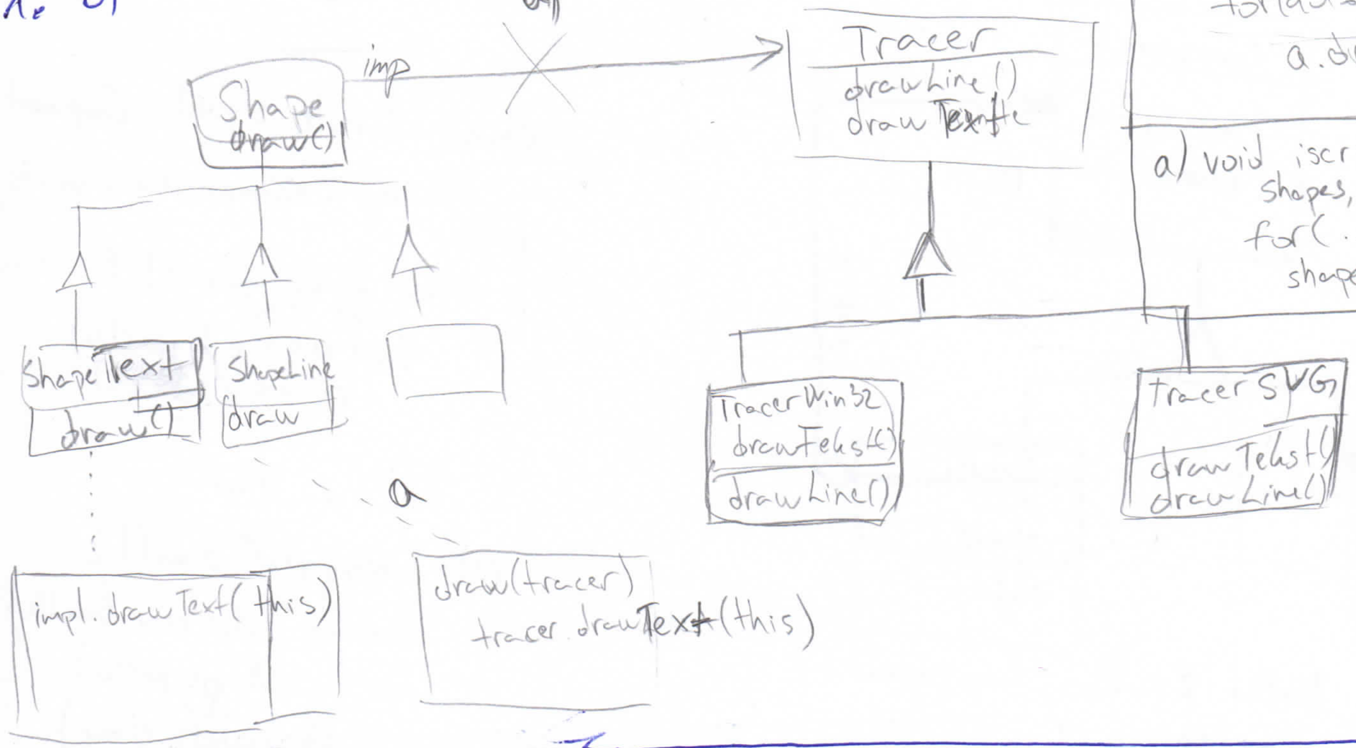
Rok 2012.

1. b)



```
b)
void iscrtaj(vector<Shape*>&
                     shapes)
    for(auto a : shapes)
        a.draw()
```

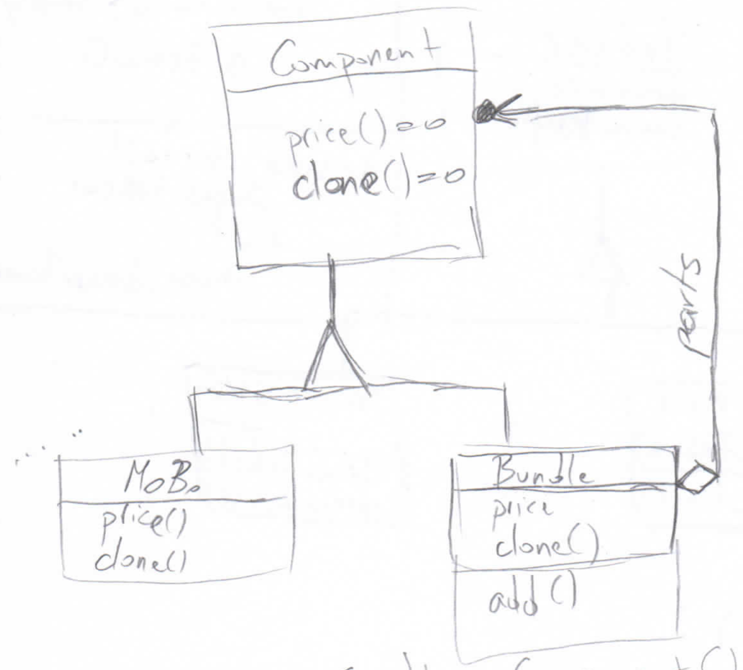
```
a) void iscrtaj(
        shapes, tracer
    for( ...
        shape.draw(tracer)
```

a)

pseudo za a) slučaj:

```
class Shape {
    virtual void draw(Tracer& t)=0;
};
class ShapeText : public Shape {
        draw(Tracer& t) {
            t.drawText(*this);
        }
};
```

2.



```
class Bundle : public Component {
  std::vector<Component*> parts;
public:
  virtual ~~double~~ price() {
    double rw;
    for (auto c : parts)
      rw += c.price()

    return rw
  virtual Component* clone() {
    Bundle* pb = new Bundle;
    for (auto& p: parts)
      pb->.add(p.clone)   ili samo p
    return pb;
}
```

```
class MoBo {public Component()
double price;
string socket;
  price() {
    return price }
  }

  clone(){
    MoBo mobo = new MoBo(price, socket)
    return mobo;
  }
}
```

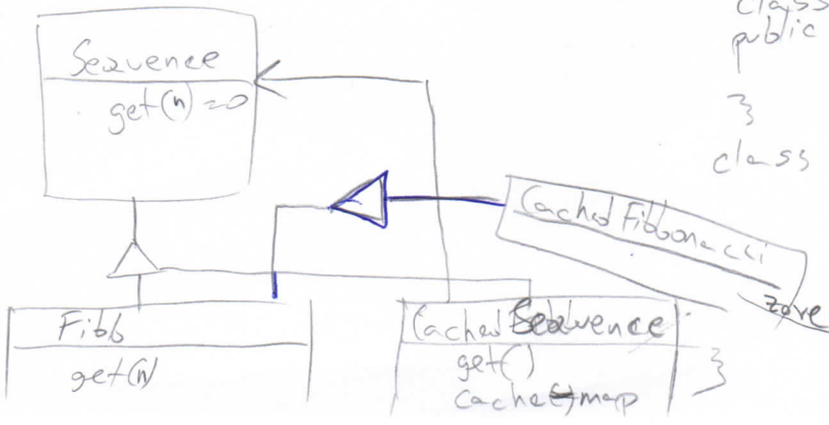
problem što get iz Fibb ne vidi cached map

3.

Fibb(0) 0
fibb(1) 1
fibb(2) 2 1 0
fibb(3) 3 2 1 0 1
fibb(4) 4 3 2 1 0 1 2 1 0
fibb(5) 5 4 3 2 1 0 1 2 1 0 3 2 1 0 1

```
class Cached Sequence
  public: Sequence
  Sequence seq;
  map <int, int> cached;
  virtual int get(int n) {
    auto it = cached.find(n)
    if (it == cached.end()) {
      int rw = seq.get(n)
      ca cache.insert(make_pair(n,rw));
      return rw;
    }
    else {
      return cached[n]
    }
```

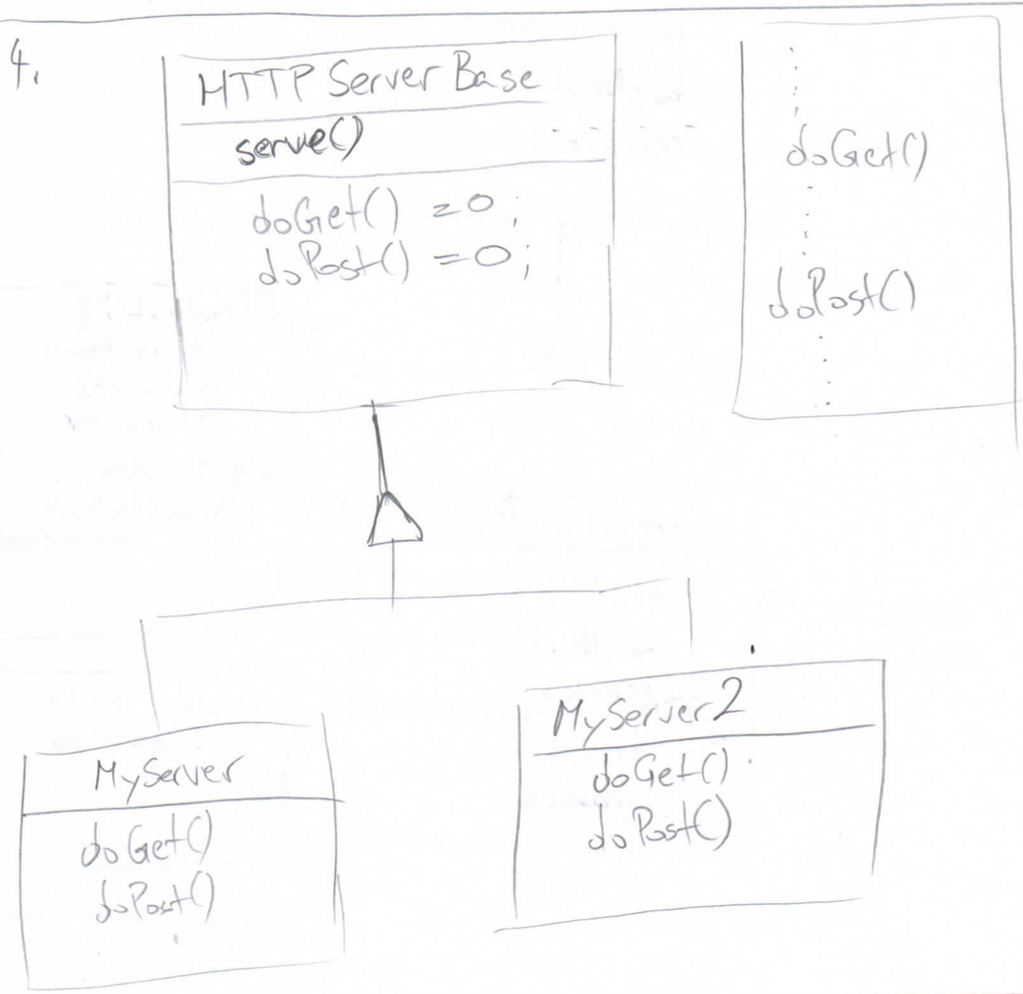
```
class Sequence {
public
    virtual int get(int n) = 0;
}
class Fibb {public Sequence {
  virtual int get(n) {
      return get(n-1) + get(n-2);
  }
}
```



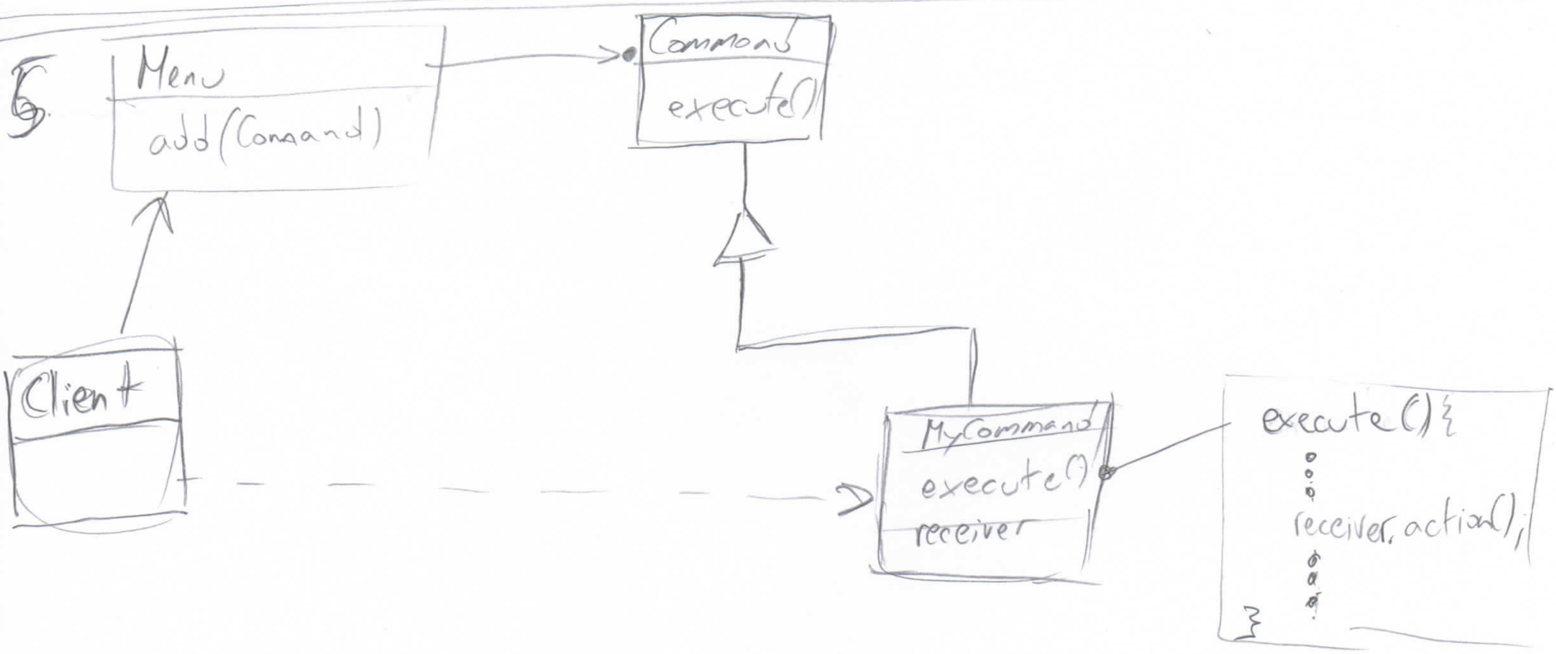
Fibb::get -> budući da je overridean

naredba
- specijalni slučaj strategije jer ima samo jednu metodu

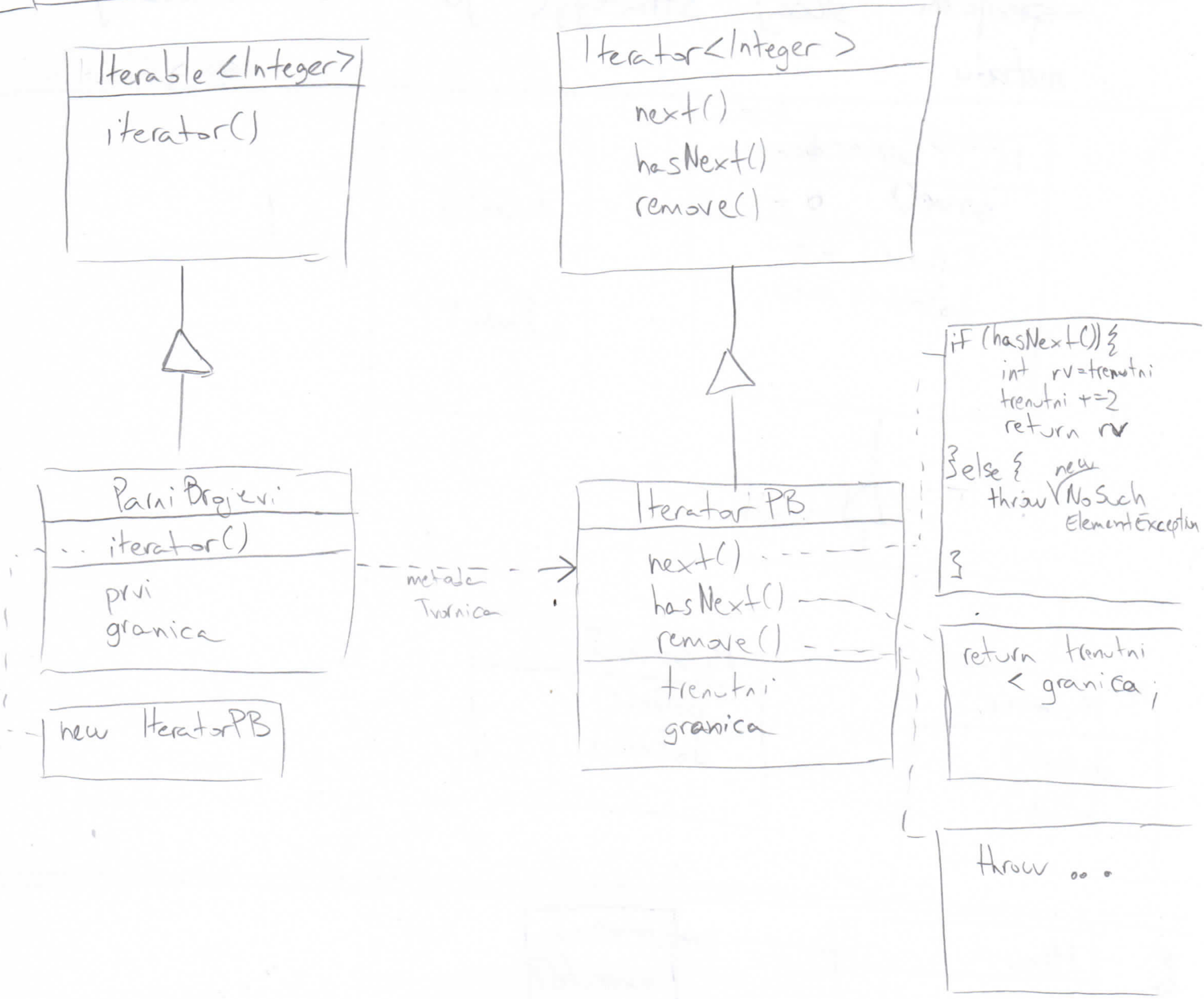
4.



5.



6. Rok 2012.



```
if (hasNext()) {
    int rv=trenutni
    trenutni +=2
    return rv
}else {
    throw new NoSuch
          ElementException
}
```

```
return trenutni
    < granica;
```

```
throw ...
```

1.MI 2013.

2. a)

```
class Comparator {
        virtual bool compere (int, int) =0
    }

    void poredaj (std::vector<int>&. L, Comparator & cmp ) {
                    ⋮
        if(cmp.compare(L(j),L[ min_j]) {
                    ⋮
    }
```

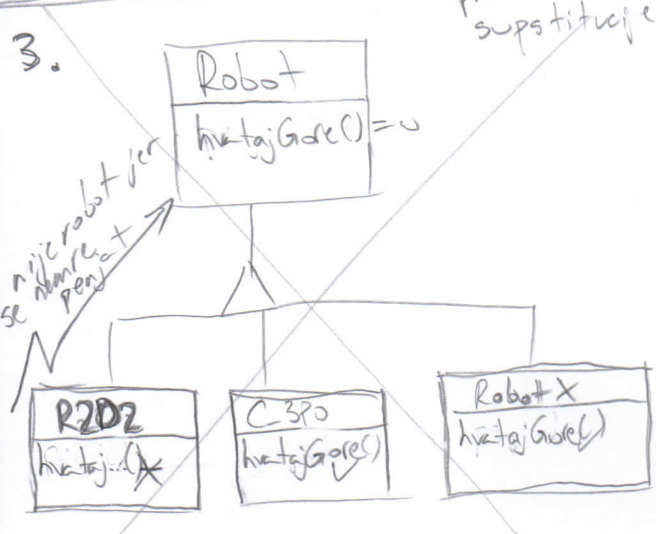
b)

```
def poredaj (L, cmp):
        ⋮
        if  cmp(L[j], L[min_j])):
        ⋮
```

c)

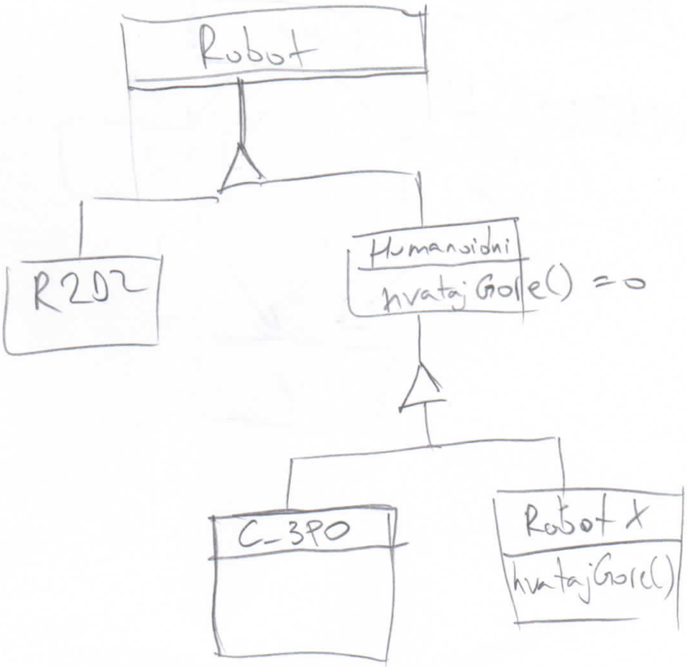
```
template<typename T, typename Comparator>
void poredaj(std::vector<T>& L, Comparator & cmp) {
        ⋮

}
```

3.

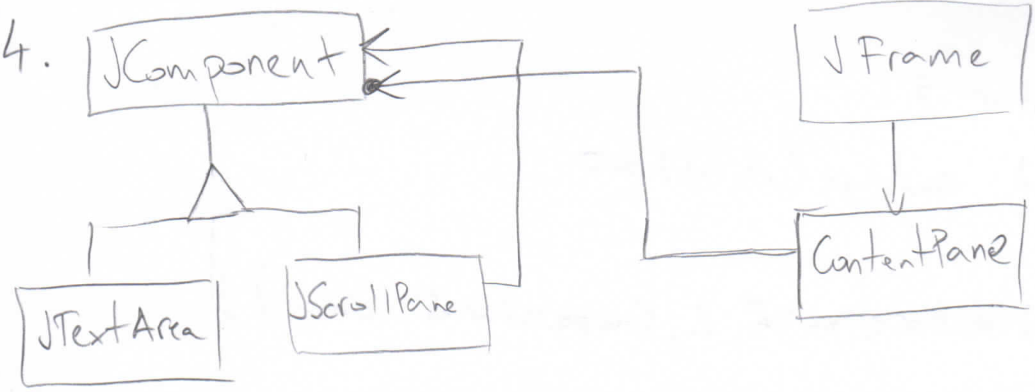


prekršeno načelo supstitucije

NOVA ORGANIZACIJA:

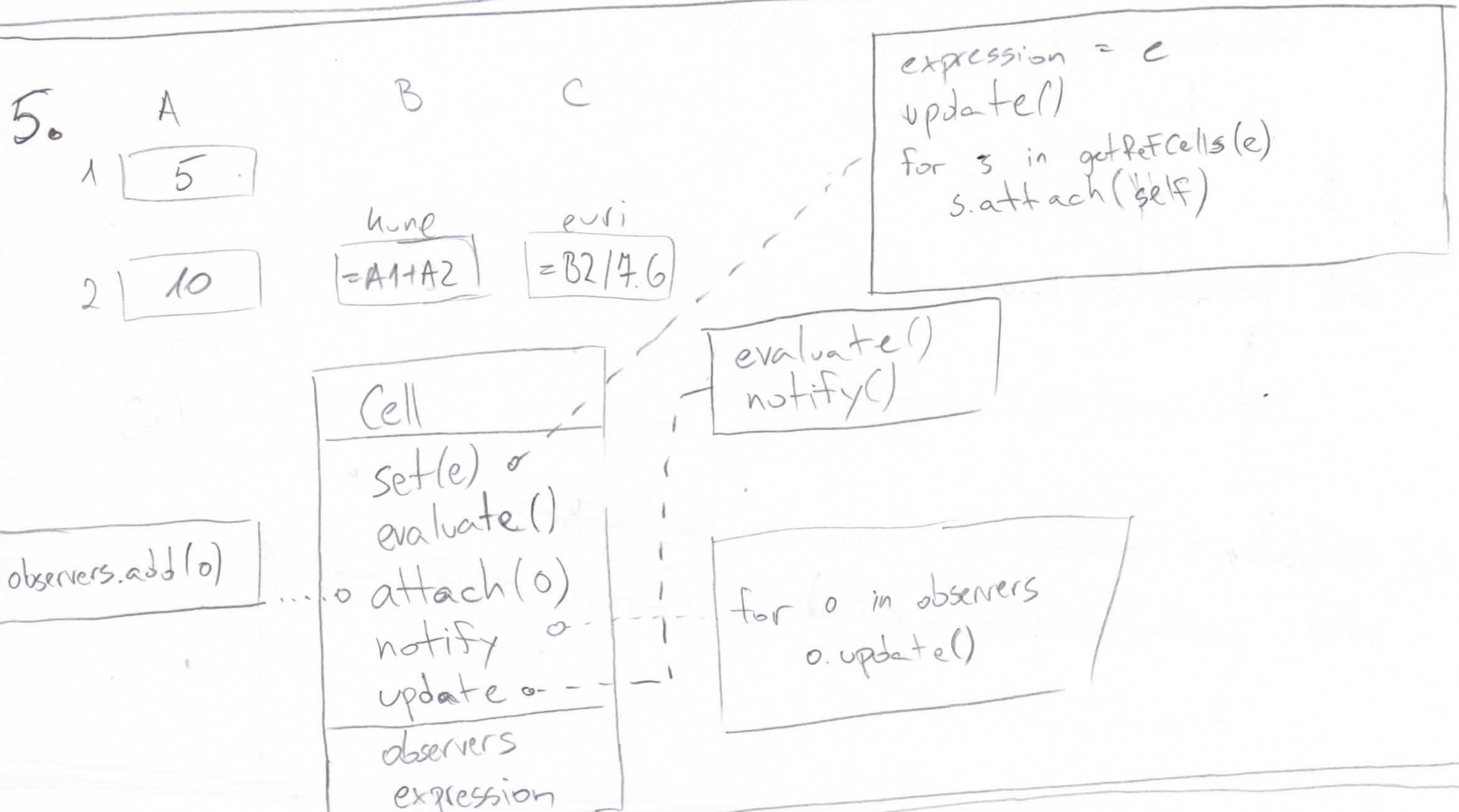


4.



5.

	A	B	C
1	5		
2	10	kune =A1+A2	euri =B2/7.6



```
expression = e
update()
for s in getRefCells(e)
   s.attach(self)
```

```
evaluate()
notify()
```

```
observers.add(o)
```

```
for o in observers
   o.update()
```

6. METODA TVORNICA

OVISNOSTI: