

- ① Razmatraćemo $f(x)$ nad 2D poljem cijelih brojeva. Uzeg zodajemo pokazivačem na 1. element image i brojem stupaca u polju width.

```
def getGradient(image, width, x, y):
    dx = image[x + y * width] - image[x - 1 + y * width];
    dy = image[x + y * width] - image[x + (y - 1) * width];
    phi = math.atan2(dx, dy)
    mag = math.sqrt(dx * dx + dy * dy)
    return mag, phi;
```

Fja određuje iznos i smjer gradijenta u koordinatama x, y zadaog 2D polja. Poznato je da u tipičnim 2D poljima s kojima naš algoritam treba raditi ima puno blakova s istim vrijednostima gradijenta (upr. s $dx=2, dy=1$). U tim se pozicijama fje sqrt i atan2 repetitivno pozivaju i istim kadrinim vrijednostima. Kako bismo mogli snimiti broj poziva fje, fja primjerom cacheirajućeg Proxyja?

- ② Razmotrite slj. program u C++-u i dorazbijte o kojemu oblikovnom dorascu se radi. Navedite što će se ispisati na ekranu nакон pokretanja programa. Predložite implementaciju ekvivalentnog programa u C-u.

```
#include <vector>
#include <cmath>
#include <iostream>

class X {
public:
    virtual double action(double) = 0;
};

class X1 : public X {
public:
    virtual double action(double x) {
        return x*x;
    }
};

class X2 : public X {
public:
    virtual double action(double x) {
        return fabs(x);
    }
};
```

```
class C {
private:
    X* px_;
public:
    C(X* px) : px_(px) {}

    double operate(
        std::vector<double> vec) {
        double rv=0
        for (auto x: vec) {
            rv += px_->action(x);
        }
        return rv;
    }

    int main() {
        C c(new X1());
        std::vector<double> data({1, 2, 1});
        std::cout << c.operate(data) << "\n";
    }
};
```

- 3.) Poznato je da ka raspodajaju redoslijed `findNearestNeighbour` koja u 1. argumentu prima `Gdekaže` objekti brojeva zadani polinomnim iteratorom, a u 2. argumentu cijeli broj. Ta funkcija pronađe elem. `Gdekaže` koji je najbliži zadanim cijelom broju. Predložite dolikovo rješenje koje bi omogućilo primjenu `fje findNearestNeighbour` nad podskupom velikih `Gdekaža`. Vaše rješ. treba izbjegći potrebu sa kopiranjem elemenata, stvaranjem novih `Gdekaža` i njenim postavljanjem.
- Prištavite primjeru vašeg rješenja u konkretnim situacijama gdje u dozir treba uzeti:
- elemente originalne `Gdekaže` su delovi sa 7,
 - svaki drugi element originalne `Gdekaže`.

- 4.) Skicirajte dijagram razreda za domenu cestovnog prometa ako su poznate slj. informacije. Važna cestovna vozila su automobili i motocikli. Konkretni automobili su Mazda 2, Citroen C4 i Škoda Fabia, dok su konkretni motocikli Kawasaki ZX-12R i Tomos APN6. Svaki automobil ima točno 4 kotača, dok motocikli imaju 2. Automobili mogu imati dizelski ili četverotaktni benzinski motor. Motocikli mogu imati šerotaktni ili 2taktni benzinski motor. Neki šerotaktni motori mogu biti korisni i u motociklima i u automobilima.

⑤ Razmatrao nared XyzyButton koji enkapsulira odgovarajuću element graf. svčelja (dugme, gumb). Razred XyzyButton nije predviđen za nadfektivacije, a isporučuje se u obliku biblioteke Xyzy. Logika je namijenjena da bez izmjena bude korištena od strane različitih aplikacija. Točnije razmatrao aplikacijski prog. koji stvara objekte leftMotor i leftButton tipova Motor tj. XyzyButton, te provodi potrebne radnje. Radi bi se na pritisak dugmeta dogodio poziv leftMotor.start(). Skrivajte svčelja i izvedbu razreda XyzyButton. Gljivne detalje izvedbe aplikacijskog programa, kao i svčelje i izvedbu eventualnih dodatnih komponenti. Nacrtajte strukturu dijagrama razreda i dijagramu slijednosti komponenta. Konektivite uskladost organizacije s oblikom rješenja. Logika smo spominjali na predavanju. Navedite o kojem se dol. obrazcu radi te povezate komponente sa slijednjim obrazcima.

⑥ Razmotrite slj. program u C++-u i obrazložite o kojem se oblikovnom obrazcu (obrazcima) radi.

```
class Y {
public:
    virtual double asdfg (std::string)=0;
};

class Y1: public Y {
public:
    virtual double asdfg (std::string) {
        //...
    };
};

class Y2: public Y {
public:
    virtual double asdfg (std::string) {
        //...
    };
};

class X {
    Y *qy-i;
public:
    virtual double qerty ()=0;
};
```

```
class X1: public X {
public:
    std::string val-;
    X1 (std::string v); val-(v){}

    virtual double qerty () {
        py -> asdfg (val-);
    };
};

class X2: public X {
public:
    std::list<X*> o-;
    X2 (std::list<X*> o-);

    virtual double qerty () {
        py -> asdfg (*x2);
        for (auto x: o-){
            x.qerty ();
        }
    };
};
```

Teorija:

1. Prilikom testiranja prog. sustava u razvoju najbolje je:
- a) testirati svaku kompj. zaslonsku aplikaciju programom
 - b) testirati citav sustav istovremeno
 - c) testirajući ne provodi dok citav sustav nije dovršen
 - d) testirati istovremeno sve kompj. iz istog paketa

2. Kojim obrazcima često stvaramo konkretnu iteraciju?

- a) strategijom
- b) jedinstv. objektom
- c) param. bloknicom
- d) metodom bloknicom

3. Metoda notify subjekta tipično poziva slj. metodu primatrača:

- a) create
- b) update
- c) attach
- d) clone

4. Osnov. razred Observer s kojim u obrazcu Host interagiraju Objekti tipično imaju sljedeći podatkovni element:

- a) polaz. na implementaciji obj.
- b) polaz. na objektu
- c) polaz. na tju cilja
- d) Cijelu obj. tog istog razreda

5. Kako se ostvaruje veza od objektne prilagodljive do prilagodljeg dojelita?

- a) preko Objekta
- b) delegacijom
- c) nadjeđivanjem
- d) uhi pedan odgovar.

6. Koji Objekti tipično pristupaju stanicu konteksta u obrazcu Stanje?

- a) Razno, preko oblikog polaz. na razred koji igra ulogu stavka
- b) Koristećem priblažnog iteratora
- c) razno, preko slat polaz. na razred koji igra ulogu konkretnog stanja
- d) Objekt stanja konteksta u račku ne pristupaći razno

7. Koja od navedenih komponentata obrazca Posjetitelj ovise o razredu broj drugih komponentata?

- a) element
- b) naredba
- c) Konkr. elementi
- d) Genr. posjetitelji

8. Odnos dvojne metode i strategije?

- a) ozivima koristi delegaciju, a strategija uveljeva u njoj
- b) ozivna je bolje prilagodljiva
- c) ozivna sačjeva u njoj objelata
- d) ozivna implicira u njoj brzini izvedenja

⑨ Kako se tipično omogućava sljedeći pristup kroz sve elemente u kompozitnog konstruktoru?

a) dodavanjem plassivaca na roditelja u konzervatu

b) -||-

sljedeći element konzervata

c) dodavanjem metode getChild (int index)

d) dodavanjem plassivaca na roditelja u apstraktnoj konzervati

⑩ Koja je namjera dorasca Prototipa?

a) odrediti razred novog objekta odabivom odgovarajućeg postojecog objekta

b) prilagoditi sučinje postojecog obj. poštovanjem prototipa

c) odrediti razred novog obj. pozivom metode `newInstance()`

d) upravljati pristupom ciljnom objektu

⑪ Odaberite konkr. strategije i dorasnu strategiju vrsti:

a) klijent konteksta b) apstr. strategija c) konkr. strat. d) autor izvoruvač koda

⑫ Kako zovemo proxy koji briše doj. koje više nisu ne potrebi:

(a) udaljeni (b) pauchni plassivac (c) garbage collector (d) sigurnosni