

# Algoritamska dekompozicija

```
class Reader {
public:
    Command* read ();
    ...

protected:
    bool readOut(int& out);
    bool readCmd(string& cmd);

private:
    ...
}

Command* Reader::read () {

    int out;
    string cmd;

    bool ret = readOut(out);
    if (!ret) return nullptr;

    ret = readCmd(cmd);
    if (!ret) return nullptr;

    return new Command(out,cmd);

}
```

Pomoćne (*helper*) operacije

```
class StackMachine {
public:
    void perform (Command*);
    ...

protected:
    void push (int);
    int pop ();
    ...
};

void StackMachine::perform (Command* cmd) {
    switch (cmd->getOpCode()) {

        case add: {
            int op1 = this->pop();
            int op2 = this->pop();
            this->push(op1+op2);
            break;
        }

        case sub: ...
        ...
    }
}

int StackMachine::pop () {
    if (this->sp==0) return 0;
    else return this->stack[--this->sp];
}

...
```

Pomoćne (*helper*) operacije

# Relacije i zavisnosti između klasa

- ❖ Da bi ispunila svoje odgovornosti, klasa (npr. *Controller*) najčešće sarađuje sa drugim klasama (npr. *Reader*, *Translator*); preciznije, objekat te klase mora da sarađuje sa objektima drugih klasa, recimo tako što od njih traži usluge, tj. poziva njihove operacije
- ❖ Da bi jedan objekat (klijent) sarađivao sa drugim (serverom), odnosno pozivao njegove operacije, mora da ima strukturnu *vezu* (*link*) sa njim — u OOP jeziku, da poseduje pokazivač / referencu ka njemu
- ❖ Relacije između klasa koje predstavljaju strukturne veze između objekata tih klasa nazivaju se *asocijacije* (*association*); veze su instance asocijacija i povezuju objekte kao instance klasa
- ❖ Tokom izvršavanja programa, objekti nastaju i nestaju, a veze između objekata se uspostavljaju i raskidaju; klase i asocijacije su deo *programa* (ili modela softvera) kao statičkog zapisa, objekti i veze su deo *objektnog prostora* (*object space*) koji postoji u vreme izvršavanja i poseduje dinamiku
- ❖ Veze asocijacija ne moraju služiti samo za “prenos” poziva operacija između objekata; one mogu i samo predstavljati prostu informaciju da su određeni objekti povezani nekom konceptualnom relacijom; na primer, činjenicu da neki *učenik* (*Student*) *pohađa* (*attend*) određenu *školu* (*School*)

