

Bridge Pattern

Hurece's Sour Man

```
interface JobRequirement {
    boolean accept(Candidate c);
}
```

```
class JavaRequirement implements JobRequirement {
    public boolean accept(Candidate c) {
        return c.hasSkill("Java");
    }
}

class Experience10YearsRequirement implements JobRequirement {
    public boolean accept(Candidate c) {
        return c.getExperience() >= 10;
    }
}
```

```
abstract class Job {
    protected List<? extends JobRequirement> requirements;
```

```

    public Job(List<? extends JobRequirement> requirements) {
        this.requirements = requirements;
    }

    protected boolean accept(Candidate c) {
        for (JobRequirement j : requirements) {
            if (!j.accept(c)) {
                return false;
            }
        }
        return true;
    }
}

class CognitectClojureDeveloper extends Job {
    public CognitectClojureDeveloper() {
        super(Arrays.asList(
            new ClojureJobRequirement(),
            new Experience10YearsRequirement()
        ));
    }
}

```

```

:                               ?

: JobRequirement, JavaRequirement, ExperienceRequirement      ?

: .

: Job, CongnitectClojureDeveloperJob, OracleJavaDeveloperJob .

: , .Job JobRequirement 가 .

: ! 가 .

```

```

Candidate joshuaBloch = new Candidate();
(new CognitectClojureDeveloper()).accept(joshuaBloch);
(new OracleSeniorJavaDeveloper()).accept(joshuaBloch);

```

```

: 가 . Job , JobRequirement .
(job) , JobRequirement

: .

: 가 defprotocol defrecord

: ,

: 가 ?

```

```

:      가      .      job      ,      job      .

:      .

:      .      가

:      ,      .      가

:      ?

:      Job      ,

:      ,      ,      .      가

:      .

:      . adhoc

:      가 ?

:      , job      ,      가      .

```

```

;; abstraction
(derive ::clojure-job ::job)
(derive ::java-job ::job)
(derive ::senior-clojure-job ::clojure-job)
(derive ::senior-java-job ::java-job)

```

```

:      가      ,

```

```

;; implementation
(defmulti accept :job)

(defmethod accept :java [candidate]
  (and (some #{:java} (:skills candidate))
    (> (:experience candidate) 1)))

```

```

:      ,      가      ,      job
accept      , adhoc      accept      .

:      ?

:      가      ::senior-java :java      .

:      ,      ::senior-java accept      ,
::java      가      가 ?

```

: .
: 가 ?
: 가 , 가 .

Plugin Backlinks:

From:
<http://jace.link/> - **Various Ways**

Permanent link:
<http://jace.link/open/bridge-pattern>

Last update: **2021/11/22 14:15**

