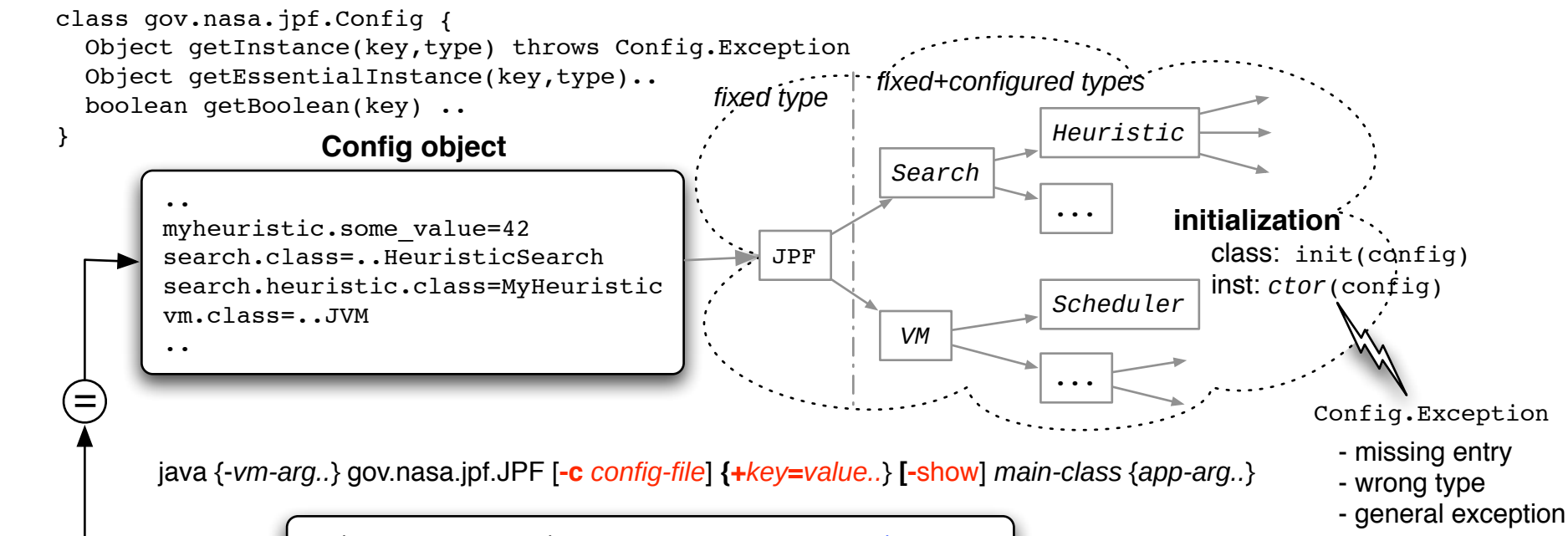
```
class gov.nasa.jpf.Config {
  Object getInstance(key,type) throws Config.Exception
  Object getEssentialInstance(key,type)..
  boolean getBoolean(key) ..
}
```



java {*-vm-arg..*} gov.nasa.jpf.JPF [**-c** *config-file*] {**+***key*=*value..*} [**-show**] *main-class* {*app-arg..*}

command line properties

```
> java gov.nasa.jpf.JPF  -c bfs.properties
      +search.heuristic.class=MyHeuristic
      +myheuristic.some_value=42
         MyTestApp
```

mode properties

```
# breadth first JPF configuration
search.class = \
  gov.nasa.jpf.search.heuristic.HeuristicSearch
search.heuristic.class = \
  gov.nasa.jpf.search.heuristic.BFSHeuristic
```

default properties

```
# section 1: general properties
log = warning
..
# section 2: Search properties
search.class = gov.nasa.jpf.search.DFSearch
..
# section 3: JVM properties
vm.class = gov.nasa.jpf.jvm.JVM
..
```

lookup

1. command-line specified file (**`-c file`**)
2. **`jpf.properties`** in current dir
3. **`jpf.properties`** in JPF root dir
   (can be set via **`+jpf.basedir=dir`**
   command line option)
4. **`jpf.properties`** resource in jar
   (loaded via `gov.nasa.jpf.JPF`)

1. **`default.properties`** in current dir
2. **`default.properties`** in JPF root dir
   (can be set via **`+jpf.basedir=dir`**
   command line option)
3. **`default.properties`** resource in jar
   (loaded via `gov.nasa.jpf.JPF`)

debug

`-show` prints used property values
`+log.level=config` prints used file sources