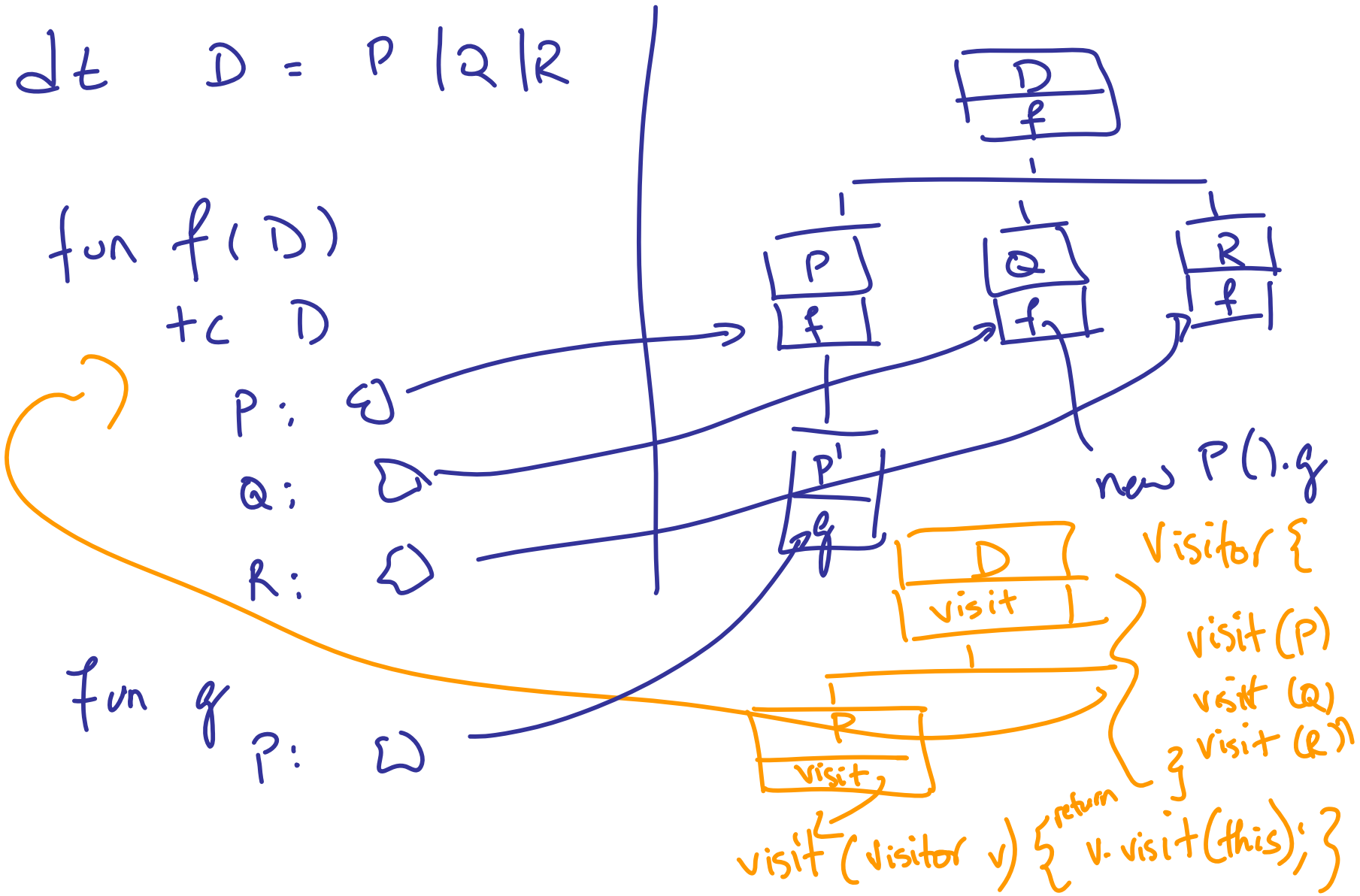
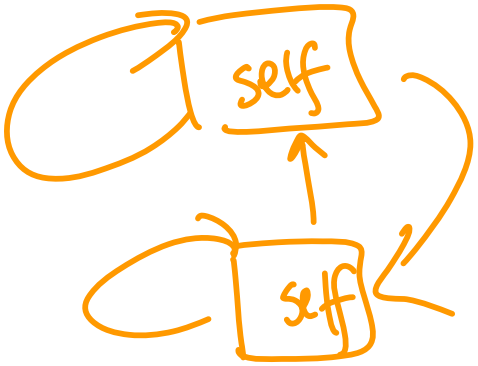
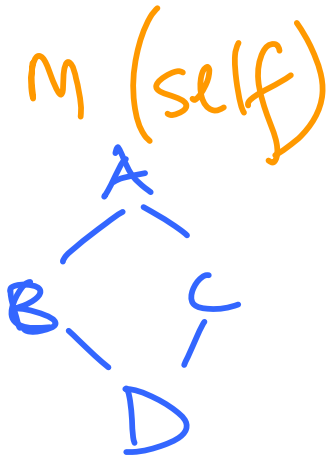
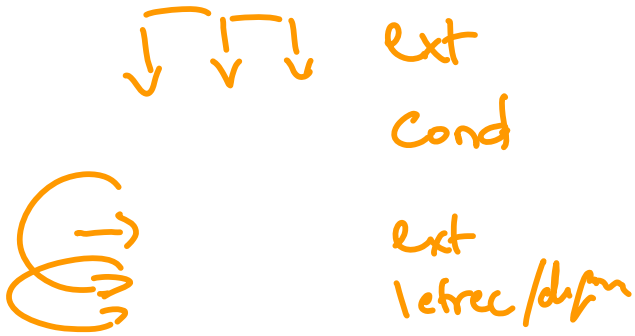

dt D = P | Q | R
fun f(D)
tc D
P:
Q:
R:
fun g
P:
D
f
P
f
Q
f
R
f
P'
g
new P().g
D
visit
R
visit
Visitor {
visit(P)
visit(Q)
visit(R)
}
visit(Visitor v) { return v.visit(this); }





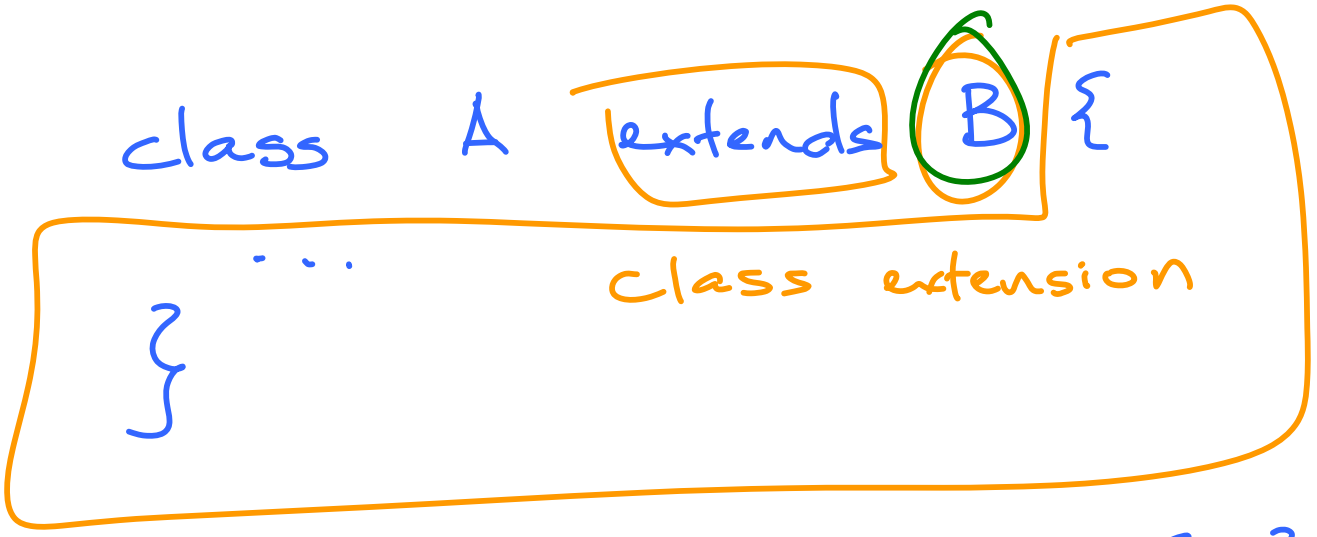



```
(case m
  [(...)  ...]
  [(...)  ...]
  [else ...])
```

classes ⟷ prototypes

si ⟷ mi

super ⟷ inner global



mixins ~~classext~~ Aext extends I {...}

class A = Aext(B) Aext(C)

singleton (proxy (factory (—)))

Program to interfaces, not implementations

Text editors

docs — code

keyboard shortcuts

conf - batch

encoding

visibility

spell checking