

TAD Pila <T> {

obs data: dict <K,V>      ¿quiere a la cabeza sea mas  
obs ultimoEnPila: N/A ← cuando llamado en una  
   secuencia

proc nuevaPila() :- pila <T> {  
    asegura { / res.data = 0 si res.ultimoEnPila = 0 }

proc opilar (inout: p.pila <T>, in v: T) {

    requiere { p = p.0 }

    asegura { p.data = setkey (p.data, p.ultimoEnPila + 1, v) si

        p.ultimoEnPila = p.ultimoEnPila + 1

proc desapilar (inout: p.pila <T>) :- T {

    requiere { / p.data ≠ 0 si p = p.0 }

    asegura { res = p.data [ p.ultimoEnPila ] si

        p.data = delkey (p.data, p.ultimoEnPila) si

        p.ultimoEnPila = p.ultimoEnPila - 1

proc tope (in: p.pila <T>) :- T  
    requiere { / p.data ≠ 0 si p = p.0 }  
    asegura { res = p.data [ p.ultimoEnPila ] }

proc vacia (in: p.pila <T>) :- bool {

    asegura { / p.data = 0 → res = true }

    Nota: 1. 0 es un los booleanos }