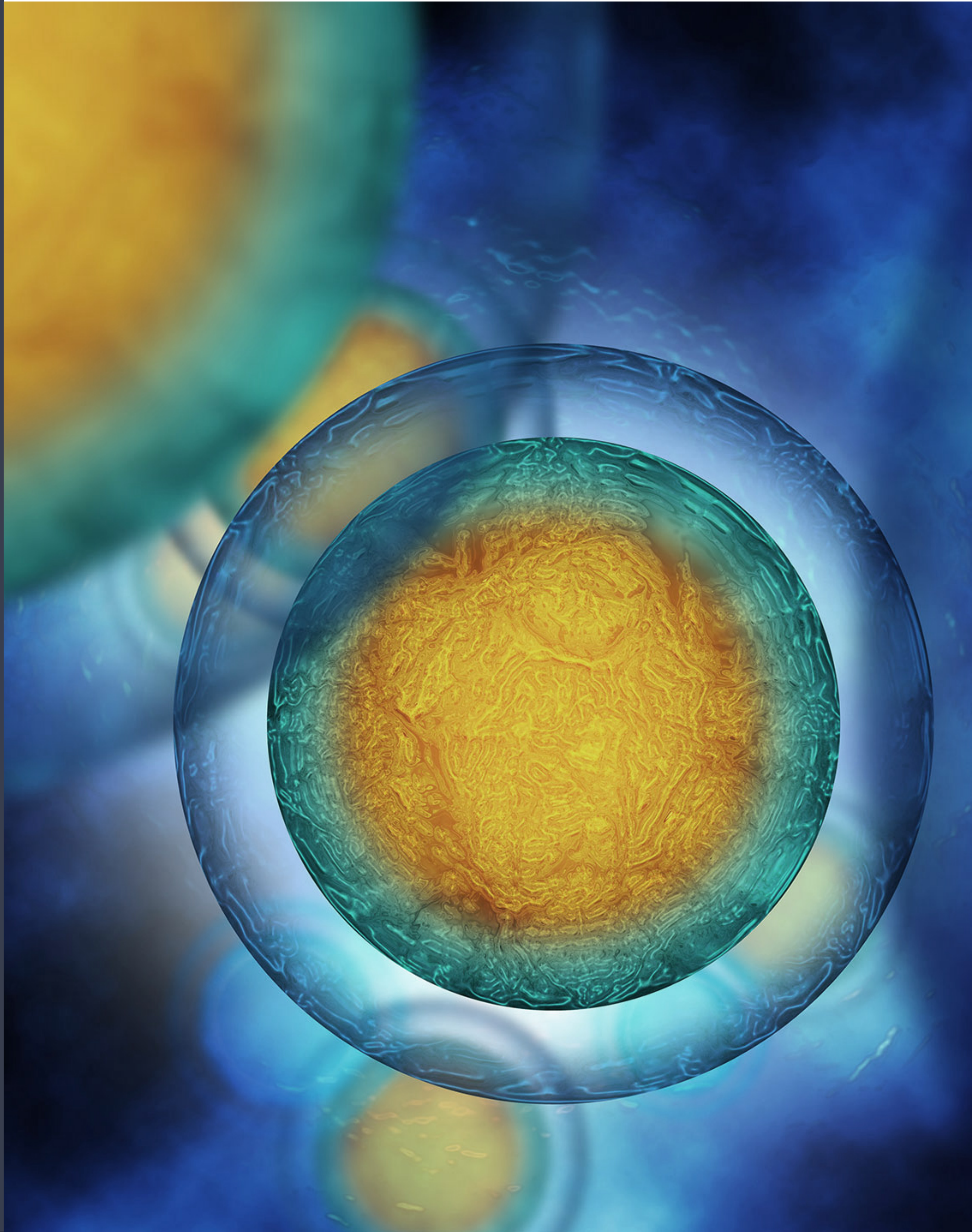
FE3CWS

OBUKA ZA NASTAVNIKE U NIJMEGENU

Intelektualni doprinos O3 iz ERASMUS+ projekta 2017-1-SK01-KA203-035402



Nekoliko riječi o

SADRŽAJU

- Jedinstvena, ali snažna tema koja se odnosi na sastav softvera, razumijevanje i ispravnost
- Dostupan na 7 jezika: engleski, mađarski, slovački, hrvatski, rumunjski, bugarski i portugalski

Co-funded by the
Erasmus+ Programme
of the European Union

Second Teacher Training Material - "Functional Programming in the New Devices Lab".

Materijal za drugu obuku nastavnika - "Funkcionalno programiranje u laboratoriju novih uređaja".

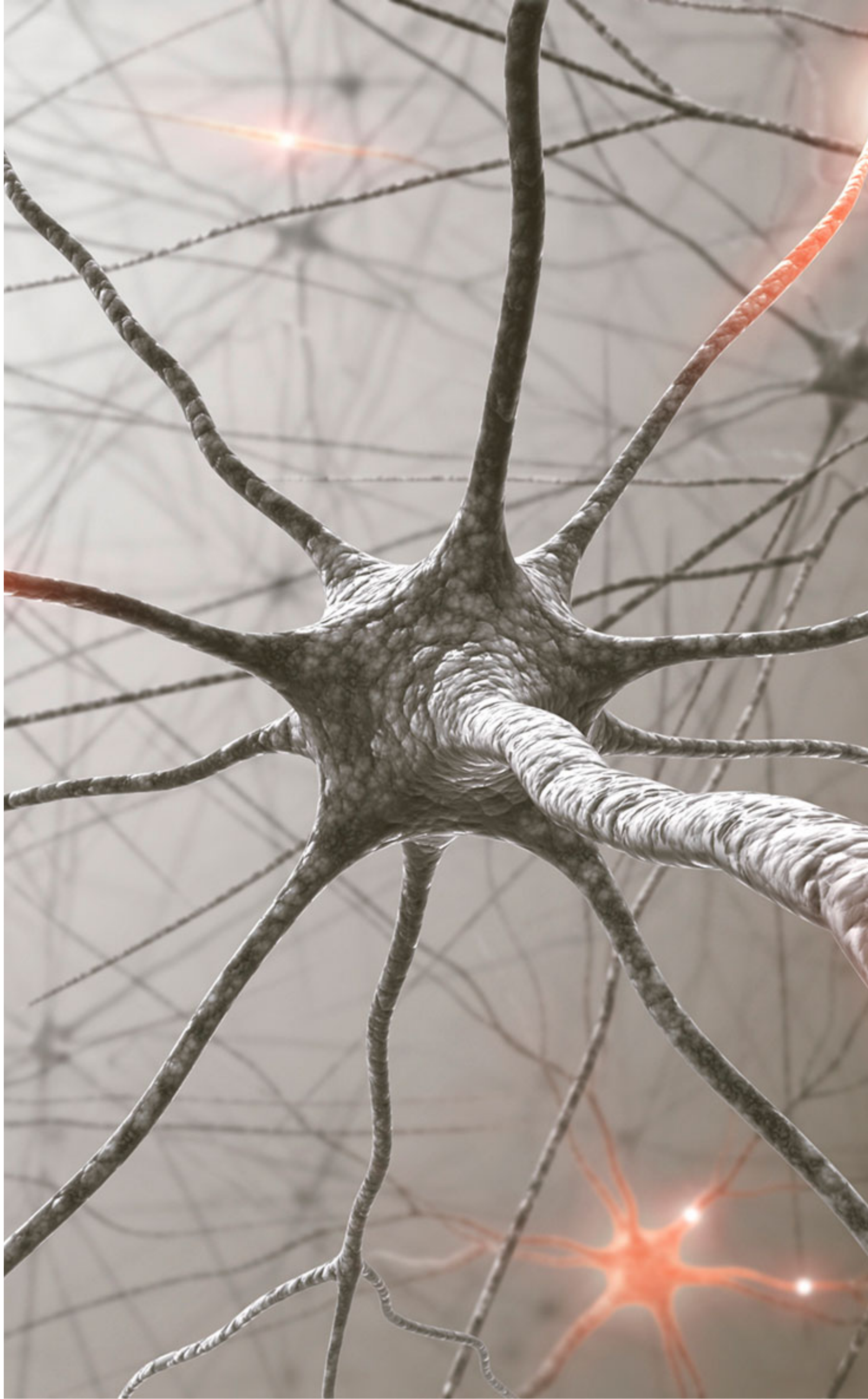
Informacije i stavovi izneseni u ovoj publikaciji su oni autora i ne moraju odražavati službeno mišljenje Europske unije. Niti institucije ni tijela Europske unije, ni bilo koja osoba koja djeluje u njihovo ime, ne može se smatrati odgovornom za uporabu koja može biti izvedena iz informacija sadržanih u njima.

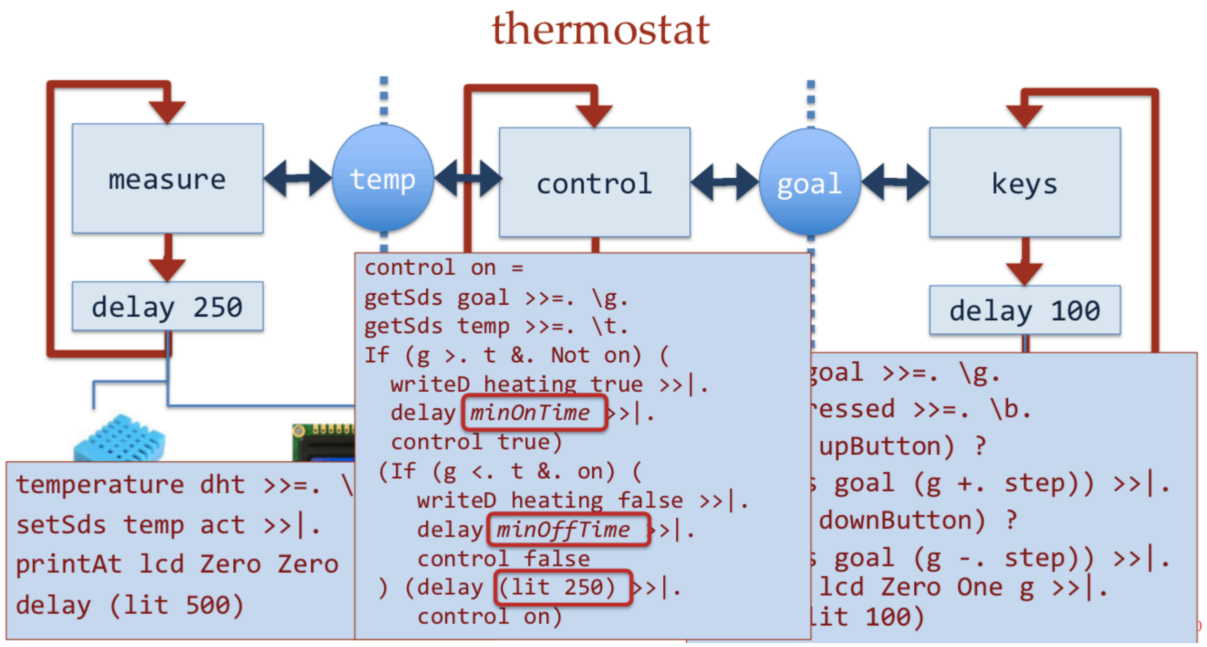
KARAKTERISTIKE

Obuka nastavnika 3COWS u Nijmegenu u Nizozemskoj obuhvaća pet dana intenzivnog školovanja i suradnje na Sveučilištu Radboud u Nijmegenu. Ciljevi ove obuke za nastavnike u sklopu projekta FE3CWS bili su upoznati polaznike sa trenutnim stanjem funkcionalnog programiranja i programiranja orijentiranog na zadatke (eng. Task Oriented Programming, TOP).

Obrađene su sljedeće teme:

- Sažetak čistog funkcionalnog programiranja u programu Clean (https://clean.cs.ru.nl/Clean)
- Generičko programiranje: kako definirati manipulacije koje se rade za bilo koji tip podataka prvog reda, čak i za tipove koji još nisu definirani.
- TOP za određivanje suradnje ljudi i automatiziranih zadataka za postizanje ciljeva. Sustav iTask podržava i nadzire izvršenje zadataka (https://clean.cs.ru.nl/ITasks). Interakcija s korisnicima uglavnom se vrši putem web-uređivača koji se generiraju iz vrsta korištenih u definicijama zadataka na visokoj razini.

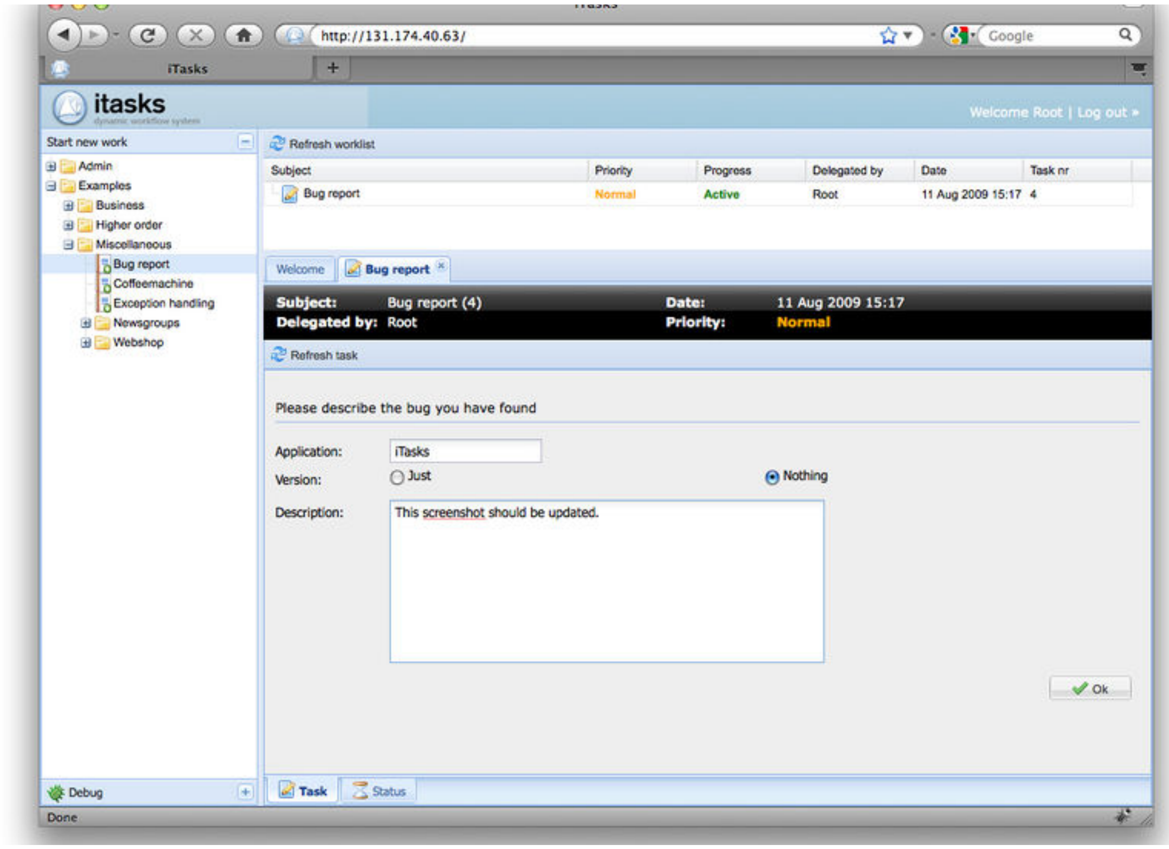




```
temperature dht >>=. \
setSds temp act >>|.
printAt lcd Zero Zero
delay (lit 500)
```

```
control on =
getSds goal >>=. \g.
getSds temp >>=. \t.
If (g >. t &. Not on) (
  writeD heating true >>|.
  delay minOnTime >>|.
  control true)
 (If (g <. t &. on) (
    writeD heating false >>|.
    delay minOffTime >>|.
    control false
 ) (delay (lit 250) >>|.
    control on)
```

```
goal >>=. \g.
ressed >>=. \b.
 upButton) ?
s goal (g +. step)) >>|.
 downButton) ?
s goal (g -. step)) >>|.
 lcd Zero One g >>|.
lit 100)
```



Subject	Priority	Progress	Delegated by	Date	Task nr
Bug report	Normal	Active	Root	11 Aug 2009 15:17	4

Subject: Bug report (4) **Date:** 11 Aug 2009 15:17
Delegated by: Root **Priority:** Normal

Please describe the bug you have found

Application: iTasks
Version: ○ Just ● Nothing
Description: This screenshot should be updated.

TOP formalizam sadrži mali broj fleksibilnih osnovnih zadataka (poput web-uređivača i kontrole IoT perifernih uređaja) i skup sustava za kombiniranje zadataka za paralelni i sekvencijalni sastav.

TOP za kontrolu Interneta stvari (eng. Internet of Things, IoT)

TOP je vrlo pogodan za kontrolu IoT-a i oslobađa programere od tereta mnogih programskih jezika, protokola, sučelja i njihove interoperabilnosti. Kako bi se nosilo s ograničenom računalnom snagom i energetskim ograničenjima uređaja za IoT, izrađen je poseban jezik specifične domene (eng. Domain Specific Language, DSL) nazvan mTasks za programiranje takvih uređaja. Sustavi iTask i mTask neprimjetno rade zajedno kako bi povezali svijet mrežnih zadataka s malenim zadacima koji se izvode na IoT-u.

Podučavanje ovih tema odvijalo se kombinacijom interaktivnih pokaznih vježbi i praktičnog programiranja s polaznicima.

Teme ove obuke za nastavnike su u središtu projekta FE3CWS/ tri CO: razumljivost (eng. COmprehensibility), mogućnost sastavljanja (eng. COmposability) i ispravnost (eng. COrrectness). Sažeti programi na visokom nivou apstrakcije izravno doprinose njihovoj razumljivosti. Sustavi za kombiniranje zadataka vrlo su korisna ilustracija mogućnosti sastavljanja složenijeg softvera. Sustav statičkog tipa čistog funkcionalnog jezika Clean onemogućava pojavu pogrešaka za vrijeme izvršavanja. Zajedno sa sažetim zapisom u ovim DSL-ovima ovo doprinosi ispravnosti programa pisanih u TOP-u.

VEZE ZA PREUZIMANJE

https://fe3cws.kpi.fei.tuke.sk/O3CRO.html

TREPEREĆI LED-OVI = ZDRAVO SVIJETU

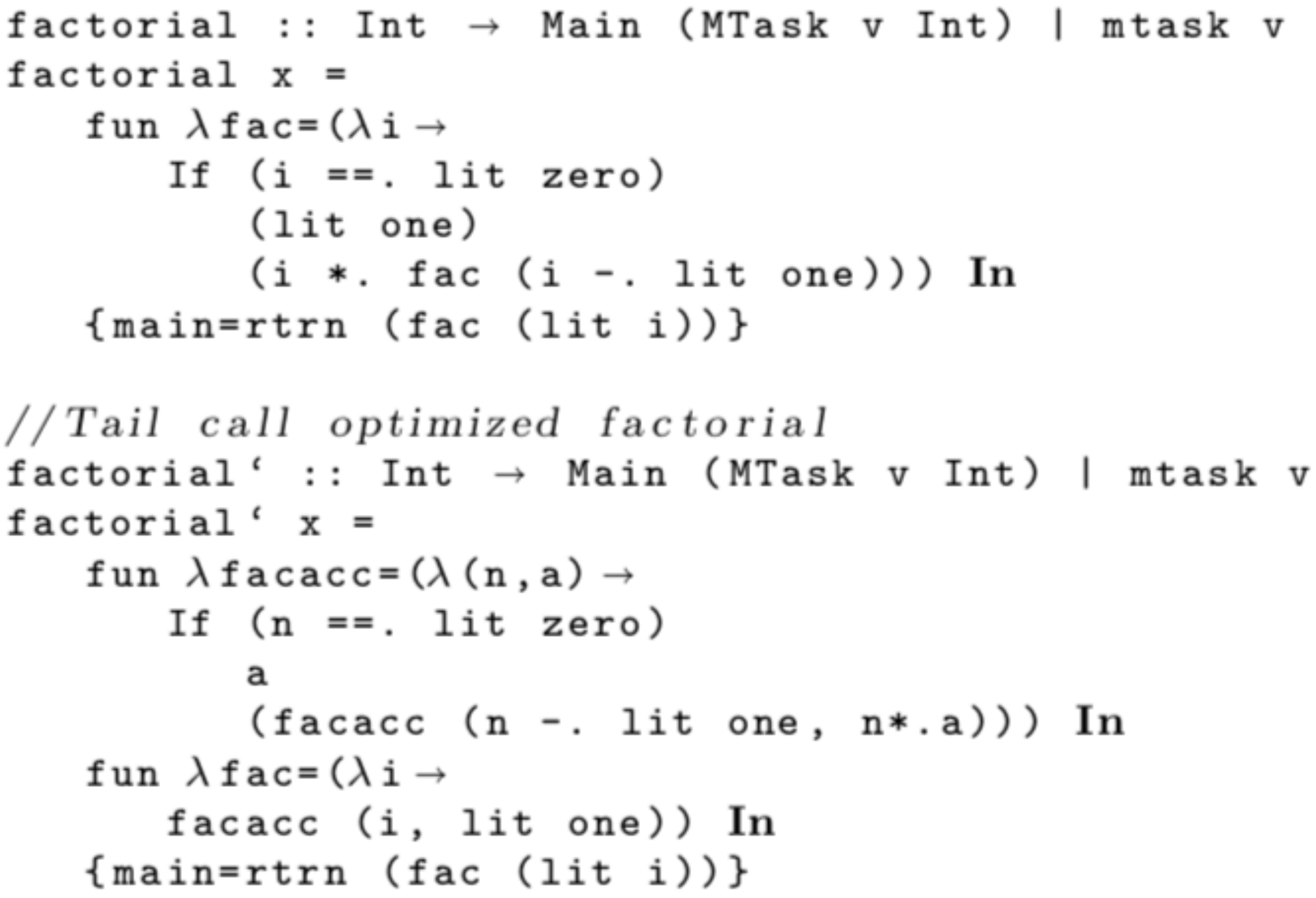
Arduino kod

```
void setup () {
        pinMode(D4, OUTPUT);
}
void loop() {
        digitalWrite(D4, HIGH);
        delay(500);
        digitalWrite(D4, LOW);
        delay(500);
}
```

mTask (Clean) kod

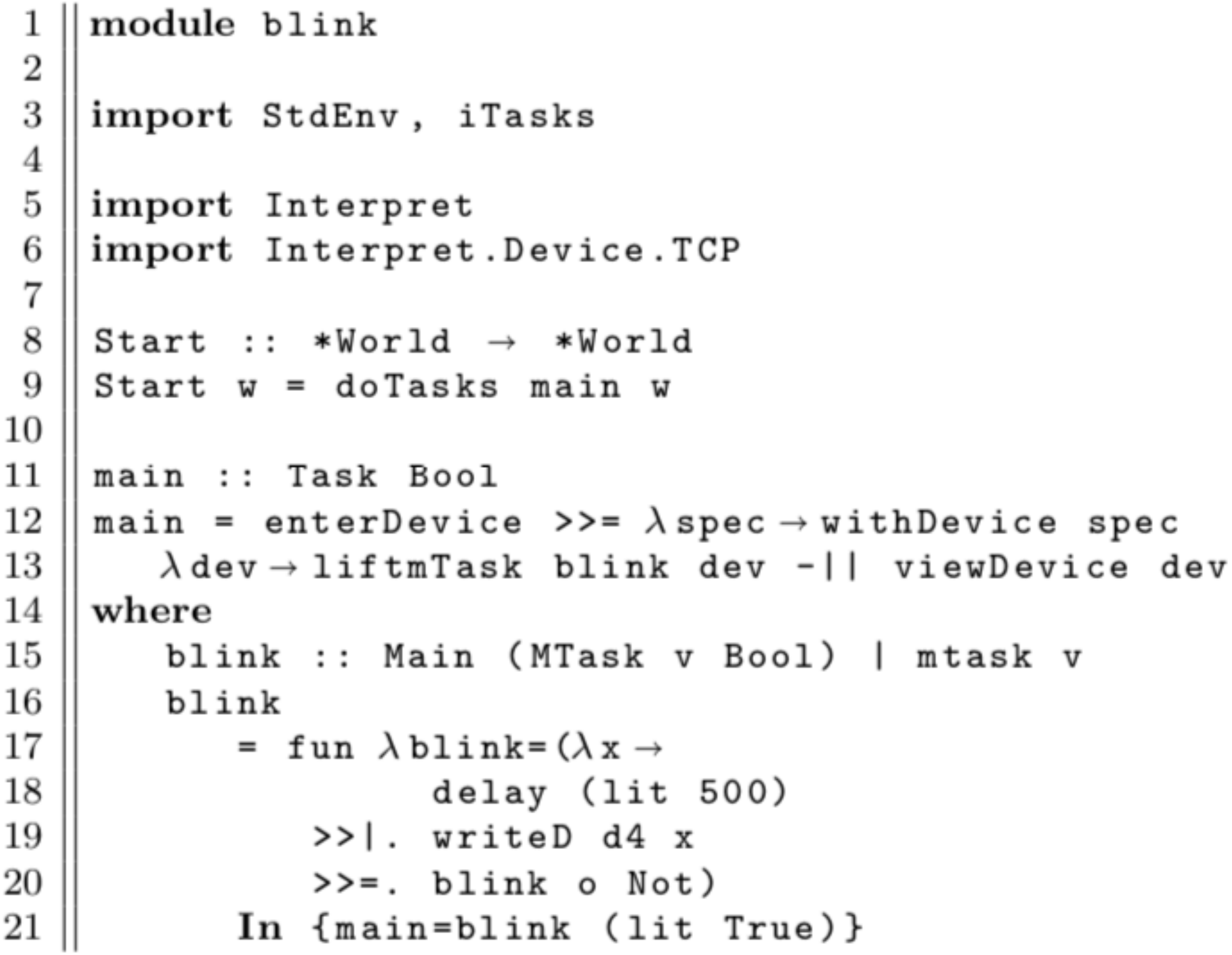
```
blink :: Main (MTask v ()) | mtask v
blink = { main=rpeat (
        writeD d4 (lit True)
>>|.  delay (lit 500)
>>|.  writeD d4 (lit False)
>>|.  delay (lit 500)
)}
```

DVA FAKTORIAL PRIMJERA



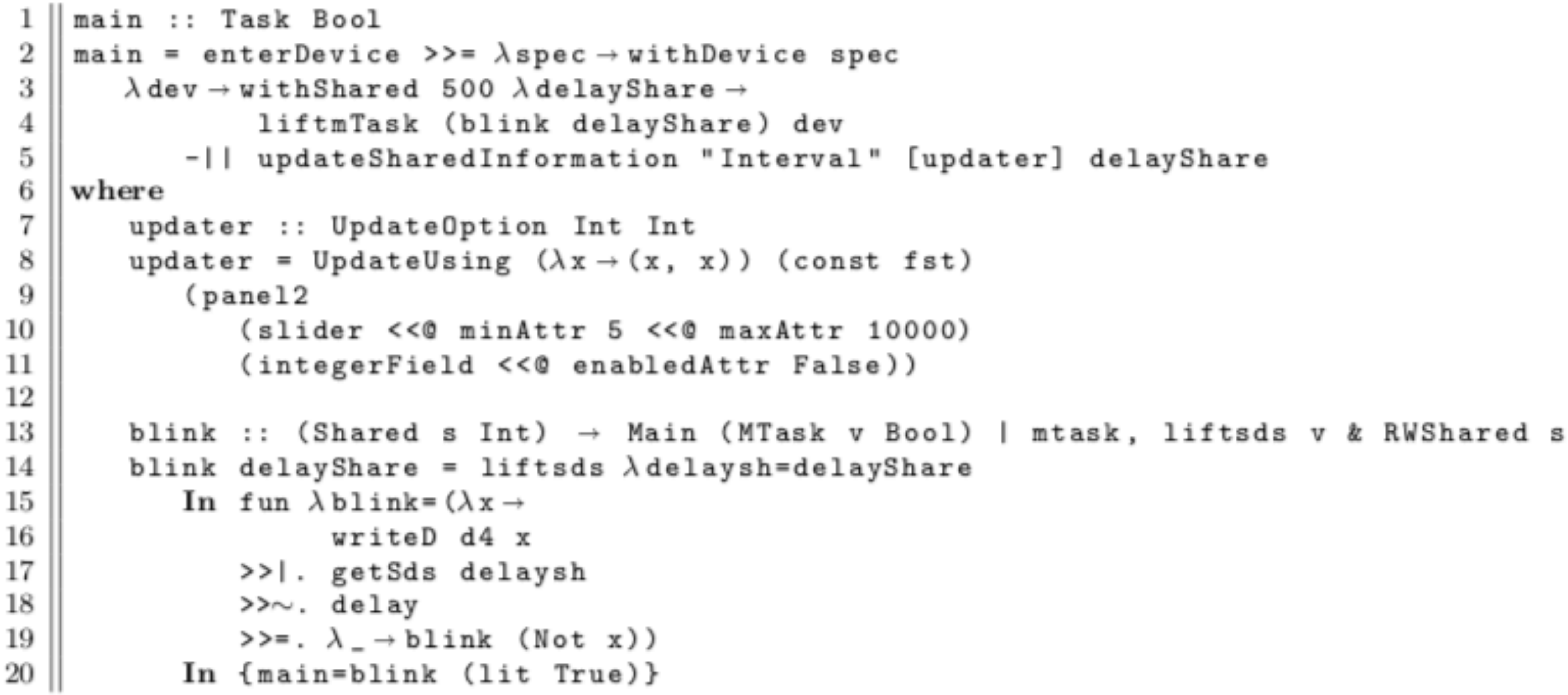
```
factorial :: Int → Main (MTask v Int) | mtask v
factorial x =
   fun λfac=(λi →
      If (i ==. lit zero)
         (lit one)
         (i *. fac (i -. lit one))) In
   {main=rtrn (fac (lit i))}

//Tail call optimized factorial
factorial‘ :: Int → Main (MTask v Int) | mtask v
factorial‘ x =
   fun λfacacc=(λ(n,a) →
      If (n ==. lit zero)
         a
         (facacc (n -. lit one, n*.a))) In
   fun λfac=(λi →
      facacc (i, lit one)) In
   {main=rtrn (fac (lit i))}
```

TREPEREĆI LED-OVI NA FUNKCIONALAN NAČIN U MTASKU



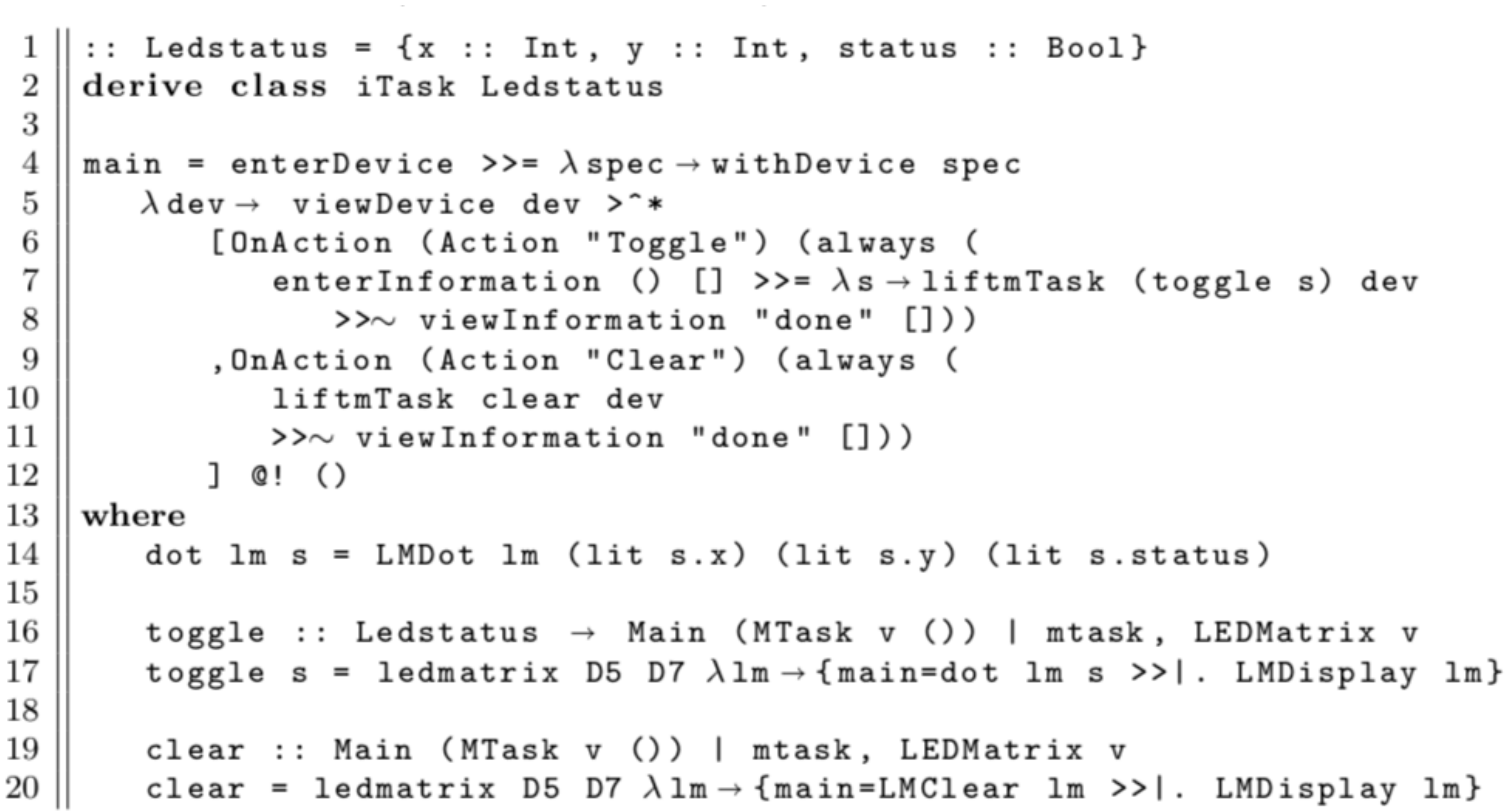
```
module blink

import StdEnv, iTasks

import Interpret
import Interpret.Device.TCP

Start :: *World → *World
Start w = doTasks main w

main :: Task Bool
main = enterDevice >>= λspec → withDevice spec
   λdev → liftmTask blink dev -|| viewDevice dev
where
   blink :: Main (MTask v Bool) | mtask v
   blink
      = fun λblink=(λx →
                 delay (lit 500)
            >>|. writeD d4 x
            >>=. blink o Not)
      In {main=blink (lit True)}
```

INTERAKTIVNO TREPTANJE



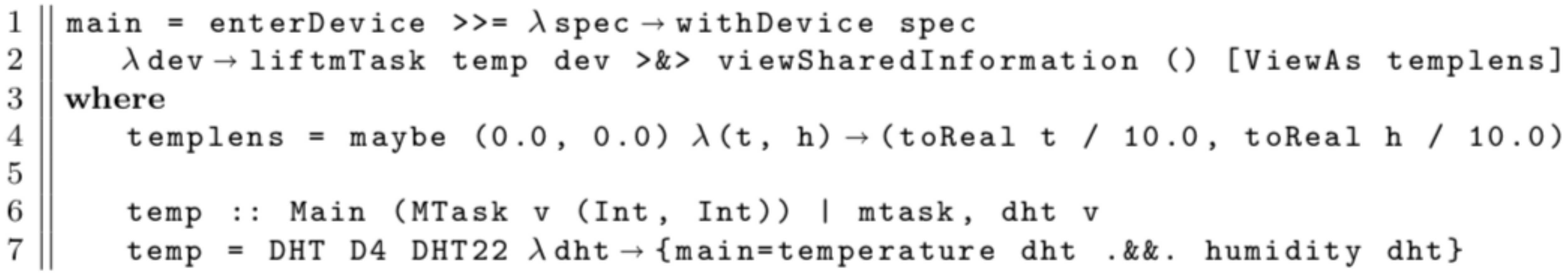
```
main :: Task Bool
main = enterDevice >>= λspec → withDevice spec
   λdev → withShared 500 λdelayShare →
          liftmTask (blink delayShare) dev
      -|| updateSharedInformation "Interval" [updater] delayShare
where
   updater :: UpdateOption Int Int
   updater = UpdateUsing (λx → (x, x)) (const fst)
      (panel2
         (slider <<@ minAttr 5 <<@ maxAttr 10000)
         (integerField <<@ enabledAttr False))

   blink :: (Shared s Int) → Main (MTask v Bool) | mtask, liftsds v & RWShared s
   blink delayShare = liftsds λdelaysh=delayShare
      In fun λblink=(λx →
               writeD d4 x
         >>|. getSds delaysh
         >>~. delay
         >>=. λ_ → blink (Not x))
      In {main=blink (lit True)}
```

INTERAKTIVNI PROGRAM MTASK ZA INTERAKCIJU S LED MATRICOM



```
:: Ledstatus = {x :: Int, y :: Int, status :: Bool}
derive class iTask Ledstatus

main = enterDevice >>= λspec → withDevice spec
   λdev →  viewDevice dev >^*
      [OnAction (Action "Toggle") (always (
         enterInformation () [] >>= λs → liftmTask (toggle s) dev
            >>~ viewInformation "done" []))
      ,OnAction (Action "Clear") (always (
         liftmTask clear dev
         >>~ viewInformation "done" []))
      ] @! ()
where
   dot lm s = LMDot lm (lit s.x) (lit s.y) (lit s.status)

   toggle :: Ledstatus → Main (MTask v ()) | mtask, LEDMatrix v
   toggle s = ledmatrix D5 D7 λlm → {main=dot lm s >>|. LMDisplay lm}

   clear :: Main (MTask v ()) | mtask, LEDMatrix v
   clear = ledmatrix D5 D7 λlm → {main=LMClear lm >>|. LMDisplay lm}
```

MJERENJE TEMPERATURE I VLAŽNOSTI



```
main = enterDevice >>= λspec → withDevice spec
   λdev → liftmTask temp dev >&> viewSharedInformation () [ViewAs templens]
where
   templens = maybe (0.0, 0.0) λ(t, h) → (toReal t / 10.0, toReal h / 10.0)

   temp :: Main (MTask v (Int, Int)) | mtask, dht v
   temp = DHT D4 DHT22 λdht → {main=temperature dht .&&. humidity dht}
```

Literatura

Peter Achten. Clean for Haskell98 Programmers. 13th July 2007.

Peter Achten, Pieter Koopman and Rinus Plasmeijer. 'An Introduction to Task Oriented Programming'. In: Central European Functional Programming School. Springer, 2015, pp. 187– 245.

Douglas Adams. The Hitchhiker's Guide to the Galaxy Omnibus: A Trilogy in Four Parts. Vol. 6. Pan Macmillan, 2017.

Matheus Amazonas Cabral De Andrade. 'Developing Real Life, Task Oriented Applications for the Internet of Things'. Master's Thesis. Nijmegen: Radboud University, 2018. 60 pp.

Tom Brus et al. 'Clean – a language for functional graph rewriting'. In: Conference on Functional Programming Languages and Computer Architecture. Springer, 1987, pp. 364– 384.

Jacques Carette, Oleg Kiselyov and Chung-Chieh Shan. 'Finally tagless, partially evaluated: Tagless staged interpreters for simpler typed languages'. In: Journal of Functional Programming 19.5 (Sept. 2009), p. 509. issn: 0956-7968, 1469-7653. doi: 10.1017/S0956796809007205.

James Cheney and Ralf Hinze. First-class phantom types. Cornell University, 2003.

Li Da Xu, Wu He and Shancang Li. 'Internet of things in industries: a survey'. In: Industrial Informatics, IEEE Transactions on 10.4 (2014), pp. 2233–2243.

L. M. G. Feijs. 'Multi-tasking and Arduino : why and how?' In: Design and semantics of form and movement. 8th International Conference on Design and Semantics of Form and Movement (DeSForM 2013). Ed. by L. L. Chen et al. Wuxi, China, 2013, pp. 119–127. isbn: 978-90-386-3462-3.

Patrick C. Hickey et al. 'Building embedded systems with embedded DSLs'. In: ACM Press, 2014, pp. 3–9. isbn: 978-1-4503-2873-9. doi: 10.1145/2628136.2628146.

Pieter Koopman, Mart Lubbers and Rinus Plasmeijer. 'A Task-Based DSL for Microcomputers'. In: Proceedings of the Real World Domain Specific Languages Workshop 2018 on - RWDSL2018. Vienna, Austria: ACM Press, 2018, pp. 1–11. isbn: 978-1-4503-6355-6. doi: 10.1145/3183895.3183902.

Mart Lubbers. 'Task Oriented Programming and the Internet of Things'. Master's Thesis.
Nijmegen: Radboud University, 2017. 69 pp.

Mart Lubbers, Pieter Koopman and Rinus Plasmeijer. 'Multitasking on Microcontrollers using Task Oriented Programming'. In: 2019 42st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). COnference on COmposability, COmprehensibility and COrrectness of Working Software. Opatija, Croatia: IEEE, 2019, pp. 1842–1846.

Mart Lubbers, Pieter Koopman and Rinus Plasmeijer. 'Task Oriented Programming and the Internet of Things'. In: Proceedings of the 30th Symposium on the Implementation and Application of Functional Programming Languages. International Symposium on Implementation and Application of Functional Languages. Lowell, MA: ACM, 2018, p. 12. isbn: 978-1-4503-7143-8. doi: 10.1145/3310232.3310239.

Rinus Plasmeijer, Peter Achten and Pieter Koopman. 'iTasks: executable specifications of interactive work flow systems for the web'. In: ACM SIGPLAN Notices 42.9 (2007), pp. 141–152.

Rinus Plasmeijer and Pieter Koopman. 'A Shallow Embedded Type Safe Extendable DSL for the Arduino'. In: Trends in Functional Programming. Vol. 9547. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016. isbn: 978-3-319-39109-0 978-3-319-39110-6. doi: 10.1007/978-3-319-39110-6.

Camil Staps and Mart Lubbers. The Clean language search engine. 2017. url: https: //cloogle.org.

Jurrien Stutterheim, Peter Achten and Rinus Plasmeijer. 'Maintaining Separation of Concerns Through Task Oriented Software Development'. In: Trends in Functional Programming. Ed. by Meng Wang and Scott Owens. Vol. 10788. Cham: Springer International Publishing, 2018, pp. 19–38. isbn: 978-3-319-89718-9 978-3-319-89719-6. doi: 10.1007/978- 3-319-89719-6_2.