

```
:: Program za ispitivanje ispravnosti bicikle za predstojeću sezonu
```

```
(deftemplate bicikla
  (slot gume(type SYMBOL))
  (slot ram(type SYMBOL))
  (slot kocnice(type SYMBOL))
  (slot prednja_srednja_i_zadnja_glava(type SYMBOL))
  (slot svetlo(type SYMBOL))
)
```

```
:: PRAVILA
```

```
:: Unesimo prvo podatke koji omogućavaju proveru minimalne ispravnosti bicikle.
:: Prvo unosimo podatke o ispravnosti guma pomoću prvih pet pravila.
```

```
(defrule pravilo_1
  (initial-fact)
=>
  (assert (task proveraj_guma))
)
```

```
:: Pravilom pravilo_2 unosimo da li su gume dobre ili ne.
```

```
(defrule pravilo_2
  ?t <- (task proveraj_guma)
=>
  (printout t "Unesi podatke o ispravnosti guma: dobre_su ili nisu_dobre." crlf)
  (bind ?g (read))
  (assert (bicikla (gume ?g)))
  (retract ?t)
  (assert (task odgovor_prvi_gume))
)
```

```
:: Ako je odgovor nisu_dobre aktiviraće se pravilo pravilo_3
```

```
(defrule pravilo_3
  ?t <- (task odgovor_prvi_gume)
  (bicikla (gume nisu_dobre))
=>
  (printout t "Gume vam nisu dobre. Izaberite šta im fali:busna_je ili iscepana_je." crlf)
  (bind ?g1 (read))
  (assert (gume ?g1))
  (retract ?t)
  (assert (task odgovor_drugi_gume))
)
```

```
:: Ako je odgovor busna_je biće nam rečeno da odemo kod vulkanizera i biće nam data cena usluge. Sa
pravila pravilo_4 direktno prelazimo na pravilo pravilo_7 koje ispituje ispravnost rama bicikle.
```

```
(defrule pravilo_4
?t <- (task odgovor_drugi_gume)
(gume busna_je)
=>
(printout t "Morate gumu odneti kod vulkanizera.Cena usluge kod vulkanizera iznosi 300din." crlf)
(retract ?t)
(assert (task provera_rama))
)
```

;; Ako je odgovor iscepana\_je bice nam receno da zamenimo gumu i bice nam data cena jedne gume. Sa pravila pravilo\_5 direktno prelazimo na pravilo pravilo\_7 koje ispituje ispravnost rama bicikle.

```
(defrule pravilo_5
?t <- (task odgovor_drugi_gume)
(gume iscepana_je)
=>
(printout t "Morate zameniti gumu.Cena jedne je 1000din." crlf)
(retract ?t)
(assert (task provera_rama))
)
```

;; Ako je odgovor dobre\_su pravilo pravilo\_6 ce aktivirati pravilo pravilo\_7 i bicemo upitani za ispravnost rama bicikle.

```
(defrule pravilo_6
?t <- (task odgovor_prvi_gume)
(bicikla (gume dobre_su))
=>
(retract ?t)
(assert (task provera_rama))
)
```

```
(defrule pravilo_7
?t <- (task provera_rama)
=>
(printout t "Unesi podatke o tome kakav je ram na sledeci nacin: dobar_je ili nije_dobar" crlf)
(bind ?r (read))
(assert (bicikla (ram ?r)))
(retract ?t)
(assert (task odgovor_prvi_ram))
)
```

;; Ako je odgovor za ispravnost rama nije\_dobar aktivirace se pravilo pravilo\_8 koje ce nas upitati u cemu je problem sa ramom.Ovaj odgovor aktivirace jedno od pravila pravilo\_13 ili pravilo\_14 a to ce zavisiti od toga kakav je odgovor bio kada smo bili upitani za ispravnost guma.

```
(defrule pravilo_8
```

```

?t <- (task odgovor_prvi_ram)
(bicikla (ram nije_dobar))
=>
(printout t "U cemu je problem sa vasim ramom:polomljen_je ili problemi_sa_volanom_ili_sedistem." crlf)
(bind ?r1 (read))
(assert (ram ?r1))
(retract ?t)
(assert (task odgovor_drugi_ram))
)

```

:: Ako izaberemo odgovor polomljen\_je aktivirace se pravilo pravilo\_9 koje ce nam reci da odemo kod varioca i dace nam cenu usluge.

```

(defrule pravilo_9
?t <- (task odgovor_drugi_ram)
(ram polomljen_je)
=>
(printout t "Morate ga odneti kod varioca.To ce vas kostati 500din." crlf)
(retract ?t)
(assert (task provera_minimalnosti))
)

```

:: Ako izaberemo odgovor problemi\_sa\_volanom\_ili\_sedistem aktivirace se pravilo pravilo\_10.

```

(defrule pravilo_10
?t <- (task odgovor_drugi_ram)
(ram problemi_sa_volanom_ili_sedistem)
=>
(printout t "Oni su povezani individualnim karakteristikama vozaca.Volan i sediste morate sami podesiti." crlf)
(retract ?t)
(assert (task provera_minimalnosti))
)

```

:: Ako je odgovor na pitanje o ispravnosti rama dobar\_je aktivirace se pravilo pravilo\_11.

```

(defrule pravilo_11
?t <- (task odgovor_prvi_ram)
(bicikla (ram dobar_je))
=>
(retract ?t)
(assert (task provera_minimalnosti))
)

```

:: Pomenuli smo situacije kada ce biti aktivirano jedno od pravila pravilo\_12, pravilo\_13 i pravilo\_14.Ona nam daju obavestenje da nemamo minimalnu ispravnost i da moramo poraditi na njoj.

```

(defrule pravilo_12

```

```

?t <- (task provera_minimalnosti)
?a <- (bicikla (gume nisu_dobre))
?b <- (bicikla (ram dobar_je))
=>
(printout t "Vasa bicikla nema ni minimalnu ispravnost.Ne moze se voziti.Morate poraditi na vasim
gumama.Kada postignete minimalnu ispravnost potrazicemo nove nedostatke." crlf)
(printout t "Da li ste izvorsili neophodne popravke?" crlf)
(retract ?t)
(retract ?a)
(retract ?b)
(assert (task ponovni_unos))
)

```

```

(defrule pravilo_13
?t <- (task provera_minimalnosti)
?a <- (bicikla (gume dobre_su))
?b <- (bicikla (ram nije_dobar))
=>
(printout t "Vasa bicikla nema ni minimalnu ispravnost.Ne moze se voziti.Morate poraditi na vasem
ramu.Kada postignete minimalnu ispravnost potrazicemo nove nedostatke." crlf)
(printout t "Da li ste izvorsili neophodne popravke?" crlf)
(retract ?t)
(retract ?a)
(retract ?b)
(assert (task ponovni_unos))
)

```

```

(defrule pravilo_14
?t <- (task provera_minimalnosti)
?a <- (bicikla (gume nisu_dobre))
?b <- (bicikla (ram nije_dobar))
=>
(printout t "Vasa bicikla nema ni minimalnu ispravnost.Ne moze se voziti.Morate poraditi i na vasim gumama
i na vasem ramu.Kada postignete minimalnu ispravnost potrazicemo nove nedostatke." crlf)
(printout t "Da li ste izvorsili neophodne popravke?" crlf)
(retract ?t)
(retract ?a)
(retract ?b)
(assert (task ponovni_unos))
)

```

:: Pravilo pravilo\_15 se aktivira ako je aktivirano jedno od pravila pravilo\_12,pravilo\_13 ili pravilo\_14.

```

(defrule pravilo_15
?t <- (task ponovni_unos)
=>
(retract ?t)
(assert (task provera_minimalnosti))

```

```
(printout t "Unesi podatke o ispravnosti guma, na sleeci nacin:dobre_su ili nisu_dobre" crlf)
(bind ?g (read))
(assert (bicikla (gume ?g)))
(printout t "Unesi podatke o tome kakav je ram na sledeci nacin:doobar_je ili nije_dobar" crlf)
(bind ?r (read))
(assert (bicikla (ram ?r)))
)
```

:: pravilo\_16 se aktivira kada je zadovoljena minimalna ispravnost (nastavljamo sa proverom ispravnosti ostalih delova bicikle)

```
(defrule pravilo_16
?t <- (task provera_minimalnosti)
(bicikla (gume dobre_su))
(bicikla (ram doobar_je))
=>
(printout t "Vase bicikla ima minimalnu ispravnost.Nastavljamo sa proverom ispravnosti ostalih delova
bicikle." crlf)
(printout t "Unesi podatke o ispravnosti kocnica na sledeci nacin:dobre_su ili nisu_dobre" crlf)
(bind ?k (read))
(assert (bicikla (kocnice ?k)))
(retract ?t)
(assert (task odgovor_prvi_kocnice))
)
```

:: Ako je odgovor na pitanje o ispravnosti kocnica nisu\_dobre tada se aktivira pravilo pravilo\_17 koje nas pita u cemu je problem sa kocnicama.

```
(defrule pravilo_17
?t<-(task odgovor_prvi_kocnice)
(bicikla (kocnice nisu_dobre))
=>
(printout t "U cemu je problem sa vasim kocnicama:pukla_sajla ili izlizane_gumice." crlf)
(bind ?k1 (read))
(assert (kocnice ?k1))
(retract ?t)
(assert (task odgovor_drugi_kocnice))
)
```

:: Ako je nas odgovor pukla\_sajla aktivirace se pravilo pravilo\_18 koje nam kaze da sajlu moramo zameniti, a odmah zatim se aktivira pravilo pravilo\_21 koje pita o ispravnosti glava.

```
(defrule pravilo_18
?t <- (task odgovor_drugi_kocnice)
(kocnice pukla_sajla)
=>
(printout t "Sajlu morate zameniti.Sajla kosta 200 din." crlf)
(retract ?t)
```

```
(assert (task provera_glava))
)
```

:: Ako je nas odgovor izlizane\_gumice aktivirace se pravilo pravilo\_19 koje nam kaze sta da radimo, a odmah zatim se aktivira pravilo pravilo\_21 koje pita o ispravnosti glava.

```
(defrule pravilo_19
?t <- (task odgovor_drugi_kocnice)
(kocnice izlizane_gumice)
=>
(printout t "Njih je lako zameniti a i nisu skupe pa to treba cesce raditi.Cena jednog para je 100din." crlf)
(retract ?t)
(assert (task provera_glava))
)
```

:: Ako je odgovor na pitanje o ispravnosti kocnica dobre\_su tada se aktivira pravilo pravilo\_20 koje aktivira pravilo pravilo\_21 koje pita o ispravnosti glava.

```
(defrule pravilo_20
?t <- (task odgovor_prvi_kocnice)
(bicikla (kocnice dobre_su))
=>
(retract ?t)
(assert (task provera_glava))
)
```

:: Pravilo pravilo\_21 koje pita o ispravnosti glava.

```
(defrule pravilo_21
?t <- (task provera_glava)
=>
(printout t "Unesi podatke o ispravnosti glava na sledeci nacin: dobre_su ili nisu_dobre." crlf)
(bind ?p (read))
(assert (bicikla (prednja_srednja_i_zadnja_glava ?p)))
(retract ?t)
(assert (task odgovor_prvi_glave))
)
```

:: Ako je odgovor na pitanje o ispravnosti glava nisu\_dobre tada se aktivira pravilo pravilo\_22 koje nas pita u cemu je problem sa glavama.

```
(defrule pravilo_22
?t <- (task odgovor_prvi_glave)
(bicikla (prednja_srednja_i_zadnja_glava nisu_dobre))
=>
(printout t "U cemu je problem sa glavama:tesko_se_okrece ili cuje_se_krcanje." crlf)
(bind ?p1 (read))
(assert (prednja_srednja_i_zadnja_glava ?p1))
```

```
(retract ?t)
(assert (task odgovor_drugi_glave))
)
```

:: Ako je nas odgovor tesko\_se\_okrece aktivirace se pravilo pravilo\_23 koje nam kaze sta da radimo, a odmah zatim se aktivira pravilo pravilo\_26 koje pita o ispravnosti svetla.

```
(defrule pravilo_23
?t <- (task odgovor_drugi_glave)
(prednja_srednja_i_zadnja_glava tesko_se_okrece)
=>
(printout t "Uzmite ulje i podmazite glavu." crlf)
(retract ?t)
(assert (task provera_svetla))
)
```

:: Ako je nas odgovor cuje\_se\_krcanje aktivirace se pravilo pravilo\_24 koje nam kaze sta da radimo, a odmah zatim se aktivira pravilo pravilo\_26 koje pita o ispravnosti svetla.

```
(defrule pravilo_24
?t <- (task odgovor_drugi_glave)
(prednja_srednja_i_zadnja_glava cuje_se_krcanje)
=>
(printout t " "Pojeli" su vam se lageri i glavi.Morate zameniti lagere." crlf)
(retract ?t)
(assert (task provera_svetla))
)
```

:: Ako je odgovor na pitanje o ispravnosti glava dobre\_su tada se aktivira pravilo pravilo\_25 koje aktivira pravilo pravilo\_26 koje pita o ispravnosti svetla.

```
(defrule pravilo_25
?t <- (task odgovor_prvi_glave)
(bicikla (prednja_srednja_i_zadnja_glava dobre_su))
=>
(retract ?t)
(assert (task provera_svetla))
)
```

:: Pravilo\_26 pita o ispravnosti svetla.

```
(defrule pravilo_26
?t <- (task provera_svetla)
=>
(printout t "Unesi podatke o ispravnosti svetla na sledeci nacin:dobro_je ili nije_dobro" crlf)
(bind ?s (read))
(assert (bicikla (svetlo ?s)))
(retract ?t)
```

```
(assert (task odgovor_prvi_svetlo))  
)
```

:: Ako je odgovor na pitanje o ispravnosti svetla nije\_dobro tada se aktivira pravilo pravilo\_27 koje nas pita u  
cemu je problem sa svetlom.

```
(defrule pravilo_27  
?t <- (task odgovor_prvi_svetlo)  
(bicikla (svetlo nije_dobro))  
=>  
(printout t "U cemu je problem sa svetlom: pregorela_sijalica ili problem_je_u_dinami." crlf)  
(bind ?s1 (read))  
(assert (svetlo ?s1))  
(retract ?t)  
(assert (task odgovor_drugi_svetlo))  
)
```

:: Ako je nas odgovor pregorela\_sijalica aktivirace se pravilo pravilo\_28 koje nam kaze sta da radimo, a  
odmah zatim se aktivira jedno od pravila koje proverava delimicnu ispravnost bicikle.

```
(defrule pravilo_28  
?t <- (task odgovor_drugi_svetlo)  
(svetlo pregorela_sijalica)  
=>  
(printout t "Morate zameniti novu sijalicu." crlf)  
(retract ?t)  
(assert (task prover_a_delimicne_ispravnosti))  
)
```

:: Ako je nas odgovor problem\_je\_u\_dinami aktivirace se pravilo pravilo\_29 koje nam kaze sta da radimo, a  
odmah zatim se aktivira jedno od pravila koje proverava delimicnu ispravnost bicikle.

```
(defrule pravilo_29  
?t <- (task odgovor_drugi_svetlo)  
(svetlo problem_je_u_dinami)  
=>  
(printout t "Morate otvoriti dinamu.Ukoliko ne vidite gde je problem potrazite pomoc." crlf)  
(retract ?t)  
(assert (task prover_a_delimicne_ispravnosti))  
)
```

:: Ako je odgovor na pitanje o ispravnosti svetla dobro\_je tada se aktivira pravilo pravilo\_30 koje aktivira  
jedno od pravila koje proverava delimicnu ispravnost bicikle.

```
(defrule pravilo_30  
?t <- (task odgovor_prvi_svetlo)  
(bicikla (svetlo dobro_je))  
=>
```



```
(retract ?t)
(assert (task provera_delimicnu_ispravnosti))
)
```

:: Pravilo pravilo\_31 proverava delimicnu ispravnost bicikle. Aktivirace se kada kocnice,glave i svetlo nisu ispravni.

```
(defrule pravilo_31
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice nisu_dobre))
(bicikla (prednja_srednja_i_zadnja_glava nisu_dobre))
(bicikla (svetlo nije_dobro))
=>
(retract ?t)
(printout t "Vasa bicikla ima samo minimalnu ispravnost.Dobili ste savete sta treba da radite. Radi vase
sigurnosti ne bi bilo lose da nas poslusate i popravite kocnice, glave i svetlo." crlf)
(halt)
)
```

:: Pravilo pravilo\_32 proverava delimicnu ispravnost bicikle. Aktivirace se kada kocnice i glave nisu ispravne.

```
(defrule pravilo_32
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice nisu_dobre))
(bicikla (prednja_srednja_i_zadnja_glava nisu_dobre))
(bicikla (svetlo dobro_je))
=>
(retract ?t)
(printout t "Vasa bicikla ima vise od minimalne ispravnosti.Dobili ste savete sta treba da radite. Radi vase
sigurnosti ne bi bilo lose da nas poslusate i popravite kocnice i glave.." crlf)
(halt)
)
```

:: Pravilo pravilo\_33 proverava delimicnu ispravnost bicikle. Aktivirace se kada kocnice nisu ispravne.

```
(defrule pravilo_33
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice nisu_dobre))
(bicikla (prednja_srednja_i_zadnja_glava dobre_su))
(bicikla (svetlo dobro_je))
=>
(retract ?t)
(printout t "Vasa bicikli je skoro potpuno ispravna.Dobili ste savete sta treba da radite. Radi vase sigurnosti
ne bi bilo lose da nas poslusate i popravite kocnice." crlf)
(halt)
)
```

:: Pravilo pravilo\_34 proverava delimicnu ispravnost bicikle. Aktivirace se kada glave i svetlo nisu ispravni.

```

(defrule pravilo_34
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice dobre_su))
(bicikla (prednja_srednja_i_zadnja_glava nisu_dobre))
(bicikla (svetlo nije_dobro))
=>
(retract ?t)
(printout t "Vasa bicikla ima malo vise od minimalne ispravnosti.Dobili ste savete sta treba da radite. Radi vase sigurnosti ne bi bilo lose da nas poslusate i popravite glave i svetlo." crlf)
(halt)
)

```

:: Pravilo pravilo\_35 proverava delimicnu ispravnost bicikle. Aktivirace se kada glave nisu ispravne.

```

(defrule pravilo_35
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice dobre_su))
(bicikla (prednja_srednja_i_zadnja_glava nisu_dobre))
(bicikla (svetlo dobro_je))
=>
(retract ?t)
(printout t "Vasa bicikla je skoro potpuno ispravna.Dobili ste savete sta treba da popravite. Radi vase sigurnosti ne bi bilo lose da nas poslusate i popravite glave." crlf)
(halt)
)

```

:: Pravilo pravilo\_36 proverava delimicnu ispravnost bicikle. Aktivirace se kada kocnice i svetlo nisu ispravni.

```

(defrule pravilo_36
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice nisu_dobre))
(bicikla (prednja_srednja_i_zadnja_glava dobre_su))
(bicikla (svetlo nije_dobro))
=>
(retract ?t)
(printout t "Vasa bicikla ima malo vise od minimalne ispravnosti.Dobili ste savete sta treba da popravite. Radi vase sigurnosti ne bi bilo lose da nas poslusate i popravite kocnice i svetlo." crlf)
(halt)
)

```

:: Pravilo pravilo\_37 proverava delimicnu ispravnost bicikle. Aktivirace se kada svetlo nije ispravno.

```

(defrule pravilo_37
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice dobre_su))
(bicikla (prednja_srednja_i_zadnja_glava dobre_su))
(bicikla (svetlo nije_dobro))

```

```
=>
(retract ?t)
(printout t "Vasa bicikla je skoro potpuno ispravna.Dobili ste savete sta treba da popravite. Radi vase
sigurnosti ne bi bilo lose da nas poslusate i popravite svetlo." crlf)
(halt)
)
```

```
:: Pravilo pravilo_38 proverava celokupnu ispravnost bicikle. Aktivirace se kada su svi elementi bicikle
ispravni.
```

```
(defrule pravilo_38
?t <- (task provera_delimicne_ispravnosti)
(bicikla (kocnice dobre_su))
(bicikla (prednja_srednja_i_zadnja_glava dobre_su))
(bicikla (svetlo dobro_je))
=>
(retract ?t)
(printout t "Vasa bicikla je potpuno ispravna.Vozite oprezno." crlf)
(halt)
)
```

```
:: KRAJ
```