

Vežba 3.

U trećem terminu vežbi koje se tiču naprednih koncepata programskog jezika *Python*, studenti bi trebalo da se upoznaju sa sledećim tematskim celinama samostalno proučavajući dolenađene materijale:

- Programski prevodioci:
 - [Kratko podsećanje na terminologiju programskih prevodilaca](#)
- Jezici specifični za domen:
 - [Uvod u jezike specifične za domen](#)
 - [Predavanje 3 \(00:00-33:00\)](#)
- *textX*
 - [Predavanje 3 \(33:01-58:33\)](#)
 - [Introduction to textX](#)
 - [Uputstvo za instalaciju textX biblioteke](#)
 - **Primeri jednostavnih JSD-ova:**
 - *Hello World* primer:
 - [Hello World tutorial](#)
 - [paket hello example](#)
 - Robot primer:
 - [Robot tutorial](#)
 - [paket robot example](#)

Zadaci za samostalni rad

Nakon prolaska kroz gorenavedene materijale, ali i uz dodatno samostalno istraživanje, potrebno je uraditi sledeće zadatke (teorijske i praktične):

1. Šta su to jezici opšte namene¹ (engl. *General Purpose Languages*)? Navesti nekoliko primera.
2. Koja je razlika između kompajliranog i interpretiranog programskog jezika?
3. Šta je programski prevodilac (engl. *compiler*)?
4. Od koliko delova se najčešće sastoje programski prevodioci? Koji su to delovi?
5. Koji su najčešće delovi prednjeg, a koji zadnjeg dela programskog prevodioca?
6. Objasniti pojmove leksičke, sintaksne i semantičke analize.
7. Šta predstavlja pojam parsiranja u kontekstu programskih prevodilaca?
8. Kakvo parsiranje može biti? U čemu je razlika?
9. Šta je Bakus-Naurova forma?
10. Šta je stablo parsiranja?
11. Šta je apstraktno stablo sintakse?
12. Po kom principu rade silazni, a po kom uzlazni parseri?
13. Kako silazni parseri rešavaju problem leve rekurzije?
14. Šta su to *LR* parseri?
15. Šta je *PEG* gramatika, a šta *PEG* parser?
16. Da li *PEG* gramatike karakteriše deklarativna priroda? Objasniti.
17. Šta su to jezici specifični za domen² (engl. *Domain Specific Languages*)? Navesti nekoliko primera.

¹ U nastavku JON.

² U nastavku JSD.

18. Koja je razlika između JON-a i JSD-a?
19. Koje kriterijume treba da zadovolji dobar JSD?
20. Kakvi sve tehnički JSD-ovi postoji? Navesti nekoliko primera za svaku kategoriju.
21. Koje su prednosti upotrebe JSD-ova?
22. Kako se upotrebom JSD-a obavlja mapiranje konceptualnog rešenja nekog domenskog problema (zadatka) na rešenje iskazano jezikom ciljne platforme?
23. Koje se arhitekture mogu koristiti prilikom implementacije JSD-a? Objasniti svaku od njih.
24. Šta je model, a šta meta-model?
25. Dat je isečak koda iskazan *robot2* jezikom, za koji je potrebno napisati *textX* gramatiku. Voditi računa da elementi jezika budu dostupni u rezultujućem objektnom modelu koji dobijete od *textX* parsera.

```
1
2  def p1 = {
3      init (3, 1);
4      move (up, 4);
5      move (left, 9);
6      move (down);
7      move (right, 1);
8  };
9
10 run(p1);
11
```

26. Dat je isečak koda iskazan *cs* jezikom, za koji je potrebno napisati *textX* gramatiku. Voditi računa da elementi jezika budu dostupni u rezultujućem objektnom modelu koji dobijete od *textX* parsera.

```
1
2 log_in
3   take GUN,
4   run left 10,
5   run up 5,
6   aim,
7   fire 3,
8   reload,
9   drop GUN,
10  take PISTOL,
11  walk right 3,
12  walk down,
13  aim,
14  fire,
15  reload,
16  drop PISTOL
17 log_out
18
```

Teme za dodatno istraživanje

Studenti koji su zainteresovani za oblast programskih prevodilaca i jezika specifičnih za domen mogu istražiti i sledeće materijale:

- Implementacija JSD-a za iskazivanje jednostavnih matematičkih izraza
 - [upotrebom silaznog parsera](#)
 - [upotrebom uzlaznog parsera](#)
- Upoznati se sa mogućnošću *textX*-a da razrešava reference na instance drugih pravila meta-modela, proučavajući primer jednostavnog *Entity* JSD-a
 - [Entity tutorial](#)
 - [paket entity example](#)
- Jedna implementacija generatora *LR* parsera u programskom jeziku *Python*:
 - [parglare \$\approx\$ Flex + Bison](#)
- Bogata [dokumentacija textX-a](#) sa (ostalim, nešto kompleksnijim) primerima