

NTP 2025/2026.

Vežba 2.

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

- 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
 - [textX playground](#)
 - VSCode integracija pomoću [textX-LS](#)
 - **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 specifični za domen¹ (engl. *Domain Specific Languages*)? Navesti nekoliko primera.
2. Koja je razlika između JON-a i JSD-a?
3. Koje kriterijume treba da zadovolji dobar JSD?
4. Kakvi sve tehnički JSD-ovi postoji? Navesti nekoliko primera za svaku kategoriju.
5. Koje su prednosti upotrebe JSD-ova?
6. Kako se upotrebom JSD-a obavlja mapiranje konceptualnog rešenja nekog domenskog problema (zadatka) na rešenje iskazano jezikom ciljne platforme?
7. Koje se arhitekture mogu koristiti prilikom implementacije JSD-a? Objasniti svaku od njih.
8. Šta je model, a šta meta-model?
9. 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.

¹U nastavku JSD.

```

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

```

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

```

11. Doraditi *robot* primer tako da omogući *for* petlje oblika:

```

1  for i in 5..10
2      up 2
3      right i
4  end

```

12. Doraditi *robot* primer tako da omogući *if* iskaze. Gde *x* i *y* varijable predstavljaju tekuću poziciju robota. Posle *if* iskaza navodi se Bulov izraz sa standardnim operatorima *<*, *>*, *>=*, *<=*, *!=*, *==*, *and*, *or*, *not*.

```

1  if x > 5
2      up 1
3  end

```

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