

## **Programmēšanas inženiera profesijas standarts**

### **1. Vispārīgie jautājumi**

1. Profesijas nosaukums – programmēšanas inženieris.
2. Profesijas kods – 2512 02.

### **2. Nodarbinātības apraksts**

1. Profesionālās kvalifikācijas līmenis – piektais profesionālās kvalifikācijas līmenis.

2. Profesionālās pamatdarbības kopsavilkums:  
– programmēšanas inženieris izstrādā programmatūru atbilstoši funkcionalitātes, kvalitātes un resursietilpības nosacījumiem, konfigurējot izstrādes vidi un rakstot programmas kodu saskaņā ar projektējuma un kodēšanas vadlīnijām; ievieš un uztur programmatūru un konsultē tās lietotājus; veic vides sagatavošanu programmatūras ieviešanai, kā arī sagatavo programmatūras testēšanas plānu un veic nepieciešamās programmatūras izmaiņas; analizē esošo situāciju; plāno programmatūras projektus; specificē prasības un sagatavo nepieciešamo lietotāja dokumentāciju; organizē un vada programmētāju darba grupu, kā arī sistemātiski pilnveido zināšanas un prasmes.

Programmēšanas inženieris strādā uzņēmumos, kas veic programmatūras izstrādi, ieviešanu vai uzturēšanu.

### **3. Profesionālās darbības veikšanai nepieciešamās profesionālās kompetences**

1. Spēja kodēt, lasot un saprotot programmatūras projektējuma aprakstus.
2. Spēja kodēt, analizējot ieejas un izejas datus.
3. Spēja kodēt, konfigurējot izstrādes vidi.
4. Spēja kodēt, rakstot programmas kodu saskaņā ar projektējumu un kodēšanas vadlīnijām.
5. Spēja kodēt, konstruējot algoritmus.

6. Spēja kodēt, lasot un analizējot svešus programmu tekstus.
7. Spēja kodēt, veidojot lietotāja saskarni.
8. Spēja kodēt, atklūdojot programmas un veicot vienībtestēšanu.
9. Spēja kodēt, analizējot programmas izpildes laiku un to optimizējot.
10. Spēja kodēt, dokumentējot kodu.
11. Spēja kodēt, veidojot programmatūras instalāciju.
12. Spēja kodēt, veidojot iebūvēto palīdzības sistēmu.
13. Spēja kodēt, apstrādājot izmaiņu pieprasījumus un problēmu ziņojumus.
14. Spēja projektēt, lasot un saprotot programmatūras prasību specifikācijas.
15. Spēja projektēt, iepazīstoties ar programmatūras projektējuma apraksta standartiem.
16. Spēja projektēt, veidojot un aprakstot programmatūras arhitektūru.
17. Spēja projektēt, analizējot dažādus tehniskos risinājumus un izvēloties piemērotāko.
18. Spēja projektēt, veidojot datu konceptuālo modeli un fizisko modeli.
19. Spēja projektēt, veidojot realizācijas modeli (klašu un/vai funkciju hierarhiju).
20. Spēja projektēt, konstruējot un aprakstot algoritmus.
21. Spēja projektēt lietotāja saskarnes.
22. Spēja projektēt, sagatavojot programmatūras projektējuma apraksta dokumentu.
23. Spēja uzturēt programmatūru, lasot un saprotot uzturamās sistēmas dokumentāciju un kodu.

24. Spēja uzturēt programmatūru, apstrādājot izmaiņu pieprasījumus un problēmu ziņojumus.
25. Spēja uzturēt programmatūru, veicot izmaiņu ietekmes analīzi.
26. Spēja uzturēt programmatūru, veicot izmaiņas programmatūrā.
27. Spēja uzturēt programmatūru, veicot uzturamās programmatūras konfigurācijas pārvaldību.
28. Spēja uzturēt programmatūru, sistematizējot uzturēšanas gaitā uzkrāto atbalsta informāciju.
29. Spēja uzturēt programmatūru, konsultējot programmatūras lietotājus.
30. Spēja ieviest programmatūru, veicot vides sagatavošanu programmatūras uzstādīšanai.
31. Spēja ieviest programmatūru, veicot datu pārvešanu.
32. Spēja ieviest programmatūru, izpildot programmatūras uzstādīšanu un parametrizēšanu.
33. Spēja ieviest programmatūru, iepazīstoties ar lietotāja dokumentāciju.
34. Spēja ieviest programmatūru, sniedzot konsultācijas programmatūras ieviešanas laikā.
35. Spēja sagatavot programmatūras testēšanas plānu.
36. Spēja sagatavot programmatūras testēšanas specifikāciju.
37. Spēja analizēt programmatūras testēšanas programmas kodu.
38. Spēja sagatavot programmatūras testēšanas testpiemērus.
39. Spēja sagatavot testēšanas vidi programmatūras testēšanai.
40. Spēja izpildīt programmatūras testēšanas testpiemērus.
41. Spēja pierakstīt programmatūras testēšanas gaitu un rakstīt problēmu ziņojumus.

42. Spēja programmatūras testēšanā analizēt kļūdu avotus (prasības specifikācijā, projektējuma aprakstā, u.c.).

43. Spēja reproducēt programmatūras testēšanā konstatētās lietotāja kļūdas.

44. Spēja sagatavot programmatūras testēšanas pārskata dokumentu.

45. Spēja specificēt prasības, iepazīstoties ar esošo pasūtītāja programmatūru.

46. Spēja specificēt prasības, analizējot prasību realizācijas iespējas.

47. Spēja sagatavot lietotāja dokumentāciju, iepazīstoties ar lietotāja dokumentācijas standartiem.

48. Spēja sagatavot lietotāja dokumentāciju, iepazīstoties ar lietotāja darījumu terminoloģiju.

49. Spēja sagatavot lietotāja dokumentāciju, rakstot un noformējot lietotāja dokumentācijas tekstu.

50. Spēja sagatavot lietotāja dokumentāciju, saskaņojot lietotāja dokumentāciju ar iebūvēto palīdzības sistēmu (Help).

51. Spēja plānot programmatūras projektu, prognozējot darba uzdevuma darbietilpību un izpildes laiku.

52. Spēja plānot programmatūras projektu, veicot individuālā darba plānošanu un kontroli.

53. Spēja plānot programmatūras projektu, piedaloties projekta gaitas izpildes apspriešanā.

54. Spēja plānot programmatūras projektu, izstrādājot programmēšanas vadlīnijas.

55. Spēja sazināties valsts valodā un divās svešvalodās.

#### **4. Profesionālās darbības pamatuzdevumu veikšanai nepieciešamās prasmes**

1. Lietot informācijas tehnoloģijas nozares standartus.

2. Lietot informācijas tehnoloģijas terminoloģiju valsts valodā un divās svešvalodās.

3. Lietot operētājsistēmas.

4. Lietot teksta un grafikas redaktorus u.c. biroja lietojumprogrammas.

5. Piedalīties projektu vadīšanā.

6. Kodēt un atklūdot programmas.

7. Pielietot projektējuma shēmas un diagrammas.

8. Projektēt algoritmus un datu struktūras.

9. Izvēlēties uzdevumu risināšanai adekvātus līdzekļus.

10. Veikt datu aizsardzības un drošības pasākumus.

11. Konfigurēt darba vietu un darba rīkus.

12. Lietot programmatūras izstrādes rīkus.

13. Analizēt programmas kodu.

14. Realizēt lietotāja saskarnes.

15. Lietot datu pieprasījumu valodas.

16. Mērīt programmatūras veiktspēju.

17. Lietot labu programmēšanas stilu.

18. Lietot programmatūras testēšanas paņēmienus.

19. Veikt sistēmu projektēšanu.

20. Strādāt komandā.

21. Vadīt citus.

22. Veikt darbu patstāvīgi.

23. Plānot izpildāmos darbus un noteikt to prioritātes.

24. Lietot informācijas meklēšanas un atlasēs līdzekļus.
25. Sagatavot prezentācijas materiālus un pasākumus, tos vadīt.
26. Pārliecināt citus un argumentēt savu viedokli.
27. Noformēt lietišķos dokumentus.
28. Ievērot profesionālās ētikas principus.
29. Ievērot darba aizsardzības prasības.
30. Pārvaldīt valsts valodu.
31. Pārvaldīt divas svešvalodas saziņas līmenī.

### **5. Profesionālās darbības pamatuzdevumu veikšanai nepieciešamās zināšanas**

1. Profesionālās darbības pamatuzdevumu veikšanai nepieciešamās zināšanas priekšstata līmenī:
  - 1.1. darba aizsardzība un ergonomika;
  - 1.2. programmatūras izstrādes projektu vadīšana.
2. Profesionālās darbības pamatuzdevumu veikšanai nepieciešamās zināšanas izpratnes līmenī:
  - 2.1. ekonomika un komercdarbība;
  - 2.2. operētājsistēmu klasifikācija un izmantošana;
  - 2.3. datorsistēmu uzbūve un funkcionēšana;
  - 2.4. datortīklu tehnoloģijas;
  - 2.5. informācijas tehnoloģiju nozares tiesiskā regulējuma pamati un standarti;
  - 2.6. programmatūras inženierija;
  - 2.7. darba tiesiskās attiecības;
  - 2.8. vides aizsardzība;
  - 2.9. profesionālie termini valsts valodā un divās svešvalodās.
3. Profesionālās darbības pamatuzdevumu veikšanai nepieciešamās zināšanas lietošanas līmenī:
  - 3.1. divas svešvalodas saziņas līmenī;
  - 3.2. matemātika;
  - 3.3. saskarsme un profesionālā ētika;
  - 3.4. lietojumprogrammatūras klasifikācija un pielietojums;

- 3.5. programmēšanas valodas;
- 3.6. datu bāzu tehnoloģijas;
- 3.7. programmatūras izstrādes tehnoloģijas;
- 3.8. objektorientētā programmēšana;
- 3.9. datu struktūras un algoritmi;
- 3.10. interneta tehnoloģijas;
- 3.11. valsts valoda.

## Pienākumi un uzdevumi

Pienākumi	Uzdevumi
1. Kodēšana.	1.1. analizēt ieejas un izejas datus; 1.2. konfigurēt izstrādes vidi; 1.3. rakstīt programmas kodu saskaņā ar projektējumu un kodēšanas vadlīnijām; 1.4. konstruēt algoritmus; 1.5. lasīt un analizēt svešus programmu tekstus; 1.6. veidot lietotāja saskarni; 1.7. skatot programmas un veikt vienībtestēšanu; 1.8. analizēt programmas izpildes laiku un to optimizēt; 1.9. dokumentēt koda izmaiņas; 1.10. veidot programmatūras instalāciju; 1.11. veidot iebūvēto palīdzības sistēmu; 1.12. apstrādāt un realizēt izmaiņu pieprasījumus un problēmu ziņojumus.
2. Projektēšana.	2.1. iepazīties ar programmatūras projektējuma apraksta standartiem; 2.2. veidot un aprakstīt programmatūras arhitektūru; 2.3. analizēt dažādus tehniskos risinājumus un izvēlēties piemērotāko; 2.4. veidot datu konceptuālo modeli un fizisko modeli; 2.5. veidot realizācijas modeli (klašu un/vai funkciju hierarhiju.); 2.6. konstruēt un aprakstīt algoritmus; 2.7. projektēt lietotāja saskarnes aprakstu; 2.8. sagatavot programmatūras projektējuma apraksta dokumentu.
3. Programmatūras uzturēšana.	3.1. apstrādāt un realizēt problēmu ziņojumus un izmaiņu pieprasījumus; 3.2. konsultēt programmatūras lietotājus; 3.3. veikt izmaiņu ietekmes analīzi

	<p>3.4 veikt uzturamās programmatūras konfigurācijas pārvaldību;</p> <p>3.5. sistematizēt uzturēšanas gaitā uzkrāto atbalsta informāciju.</p>
4. Programmatūras ieviešana.	<p>4.1. veikt vides sagatavošanu programmatūras uzstādīšanai;</p> <p>4.2. izpildīt programmatūras uzstādīšanu un parametrizēšanu;</p> <p>4.3. iepazīties ar lietotāja dokumentāciju;</p> <p>4.4. veikt datu pārvešanu;</p> <p>4.5. sniegt konsultācijas programmatūras ieviešanas laikā.</p>
5. Programmatūras testēšana.	<p>5.1. sagatavot testēšanas plānu;</p> <p>5.2. sagatavot testēšanas specifikāciju;</p> <p>5.3. analizēt programmas kodu;</p> <p>5.4. sagatavot testa piemēra datus;</p> <p>5.5. sagatavot testēšanas vidi;</p> <p>5.6. izpildīt testa piemērus;</p> <p>5.7. pierakstīt testēšanas gaitu un rakstīt problēmu ziņojumus;</p> <p>5.8. analizēt kļūdu avotus (prasības specifikācijā, projektējuma aprakstā, u.c.);</p> <p>5.9. reproducēt lietotāja konstatētās kļūdas;</p> <p>5.10. sagatavot testēšanas pārskata dokumentu.</p>
6. Prasību specificēšana.	<p>6.1. iepazīties ar programmatūras prasību specifikācijas standartiem;</p> <p>6.2. noskaidrot lietotāja funkcionālās prasības;</p> <p>6.3. noskaidrot prasības lietotāja saskarnei;</p> <p>6.4. novērtēt datu apjomus un noskaidrot veiktspējas prasības;</p> <p>6.5. noskaidrot drošības, drošuma un vides prasības, prasības mijiedarbībai ar citām sistēmām un citas tehniskās prasības;</p> <p>6.6. analizēt prasību realizācijas iespējas;</p> <p>6.7. sagatavot programmatūras prasību specifikācijas dokumentu.</p>
7. Esošās sistēmas analīze.	<p>7.1. intervēt projekta pasūtītāju un apkopot interviju rezultātus;</p> <p>7.2. iepazīties ar pasūtītāja darbību reglamentējošo dokumentāciju;</p> <p>7.3. iepazīties ar esošo programmu nodrošinājumu;</p> <p>7.4. apkopot sistēmas analīzes rezultātus vienotā dokumentā.</p>



8. Lietotāja dokumentācijas sagatavošana.	8.1. iepazīties ar lietotāja dokumentācijas standartiem; 8.2. iepazīties ar lietotāja biznesa terminoloģiju; 8.3. rakstīt un noformēt lietotāja dokumentācijas tekstu; 8.4. saskaņot lietotāja dokumentāciju ar iebūvēto palīdzības sistēmu (help).
9. Programmatūras projekta plānošana.	9.1. novērtēt darba uzdevuma darbietilpību un izpildes laiku un sastādīt kalendāro plānu; 9.2. veikt individuālā darba plānošanu un kontroli; 9.3. piedalīties projekta gaitas izpildes apspriešanā.

### **Programmēšanas inženiera profesijas standarta izstrādes darba grupa:**

- Valdis Lauks – „Fortech” programmatūras izstrādes daļas direktors;  
Ivo Odītis – Latvijas Bankas informācijas sistēmu pārvaldes vadītāja vietnieks;  
Jānis Plūme – SIA „IT Alise” kvalitātes vadītājs;  
Uldis Sukovskis – Rīgas Informācijas tehnoloģijas institūta laboratorijas vadītājs;  
Olga Sabanska – Rīgas Valsts tehnikuma direktora vietniece datorizglītībā;  
Renata Frolkova – Rīgas Tehniskās koledžas mācību metodiskā darba vadītāja;  
Gunārs Matisons – Rīgas Tehniskās universitātes Automātikas un skaitļošanas tehnikas fakultātes mācību prodekāns.

### **Programmēšanas inženiera profesijas standarta eksperti:**

- N.Freivalds – Satiksmes ministrijas valsts sekretārs;  
E.Egle Darba – devēju konfederācijas ģenerāldirektore;  
L.Straujuma – Reģionālās attīstības un pašvaldību lietu ministrijas valsts sekretāre.