

Metodiskie materiāli: tēmas 1.4–1.8

54 materiāli, 6 temati, viena secība

Ejiet secīgi

1.4 → 1.8 veido secīgu posmu: ievade, nosacījumi, pārbaude, saraksti, cikli. Katra tēma balstās uz iepriekšējo.

Lasiet birkas

„2 st.” — plānotais stundu skaits; „Video” — pieejams video skolēnam; „Pārbaudes darbs” — vērtēšanas stunda.

Meklējiet pēc koda

Meklētājā var izmantot gan tēmas kodu („1.8”), gan atslēgvārdu („cikli”).

Mācību materiāli

Sveiki, Admin! Izvēlieties tematu vai meklējiet stundu pēc nosaukuma.

54 stundu materiāli **205** mācību stundas **54/54** jūs apskatījāt

TURPINĀT · ATVĒRTS ŠODIEN
1.8. Cikla konstrukcijas

SEMINĀRS · 13.10.2026. 15.00–16.30
Pirmais metodiskais seminārs — pieteikties

Meklēt pēc nosaukuma vai koda, piem. 3.2 vai cikli

Visi Stundas Pārbaudes darbi (13) ☐ Tikai neskatītie 54 materiāli

☐ Rādīt dzēšanas pogas

TEMATI

- 1 Kā uzrakstīt programmu izvēlētajā programmēšanas valodā? 17
- 2 Kā izplānot un izveidot lietotājam atbilstošu... 7
- 3 Kā automātiski apstrādāt un attēlot datus? 11
- 4 Kā sistemātiski pārbaudīt un atklāt risinājumu? 7
- 5 Kā izvēlēties un savā projektā efektīvi izmantot jau esošu... 6
- 6 Kā īstenot risinājumu? 6

1 Kā uzrakstīt programmu izvēlētajā programmēšanas valodā?
17 materiāli 54 no 60 mācību stundām **apskatīti 17/17**

- 1.1 Ievads programmēšanā**
2 st. Video ✓ Skatīts 114 - 436 **Labot**
- 1.2 Teksta izvade un formatēšana ar operatoru print()**
2 st. Video ✓ Skatīts 85 - 643 **Labot**
- 1.3 Mainīgo definēšana un matemātisko aprēķinu veikšana**
2 st. Video ✓ Skatīts 56 - 434 **Labot**
- 1.4 Lietotāja ievade un tipu pārveidošana**
2 st. Video ✓ Skatīts 42 - 316 **Labot**
- 1.5 Nosacījumu operatori un lēmumu pieņemšana programmēšanas uzdevumos**
3 st. Video ✓ Skatīts 36 - 357 **Labot**
- 1.6 Zināšanu pārbaude Nr. 1**
2 st. Pārbaudes darbs ✓ Skatīts 54 - 356 **Labot**
- 1.7 Sarakstu izmantošana**
37 - 300 **Labot**

Tēma = stundu plāni + datnes + video

Stundu plāns katrai stundai

SR un norise: aktualizācija → apjēgšana → lietošana
→ refleksija. Katru plānu var lejupielādēt PDF formātā.

Rediģējamas datnes

Prezentācijas (.pptx) un uzdevumu lapas (.docx) pielāgojiet savai klasei, nevis izmantojiet bez izmaiņām.

Papildu sadaļas

Špikeris skolēnam, papildu uzdevumi snieguma līmeņiem, pašvērtējuma tabula.

Video

Tēmas skaidrojums skolēniem platformā YouTube — par to nākamajos slaidos.

1.4 1. temats

Lietotāja ievade un tipu pārveidošana

2 mācību st.
3 datnes
1 video
4.3 / 5 · 3 vērt.

Novērtēt materiālu
PDF
Drukāt

2 stundu plāni · 3 papildu sadaļas

Izvērst visuSakļaut visu

Stundu plāns (2 stundas)

1 1. stunda

Lietotāja veikta datu ievade un datu tipu pārveidošana Python vidē (1. daļa. Teorija un pirmais praktiskais darbs)

PDF

Stundas tēma	Lietotāja veikta datu ievade un datu tipu pārveidošana Python vidē (1. daļa. Teorija un pirmais praktiskais darbs)
Stundu skaits	1. no 2
Stundas/-u SR	- Izprot funkcijas <code>input()</code> darbības principu un datu tipu nozīmi programmā. - Prot lietot funkcijas <code>int()</code> un <code>float()</code> tipu pārveidošanai. - Prot lietot <code>try/except</code> kļūdu apstrādei. - Prot formatēt izvadi lietojot <code>f-string</code> un funkciju <code>round()</code>. - Prot izmantot zināšanas praktiskā uzdevumā ar lietotāja veiktu datu ievadi un aprēķinu veikšanu.

STUNDAS NORISE

Aktualizācija	Skolotājs atgādina iepriekšējā tēmā apgūto par funkciju <code>print()</code> un mainīgajiem. Skolotājs uzdod jautājumu klasei: "Kā programma var iegūt informāciju no lietotāja?"
Apjēgšana	Teorijas un demonstrējumu laiks – 15 min. Skolotājs izmanto prezentāciju <i>PRG-LV-1-4.pptx</i> , kurā iekļauti piemēri par - lietotāja veiktu ievadi ar <code>input()</code>; - datu tipu pārveidošanu (<code>int()</code>, <code>float()</code>); - formatētu izvadi ar <code>f-string</code>; - ievades kļūdām.

SATURĀ

1. stunda
2. stunda

- ŠPIKERIS skolēnam – Python komandu lietošana
- Papildu uzdevumu komplekts snieguma līmeņa pārbaudei
- Skolēna pašvērtējuma tabula

LEJUPIELĀDEI

PRG-LV-1-4.pptx

Prezentācija · 65 KB

PRG-LV-1-4-1(2).docx

Dokuments · 109 KB

PRG-LV-1-4-2(2).docx

Dokuments · 39 KB

VIDEO

1

Atvērt YouTube ↗

Video skolēna atbalstam, nevis skolotāja vietā

Pirms stundas

Apdzīvotā klase: skolēns noskatās video mājās, tāpēc stundā atliek vairāk laika praktiskajam darbam.

Pēc stundas

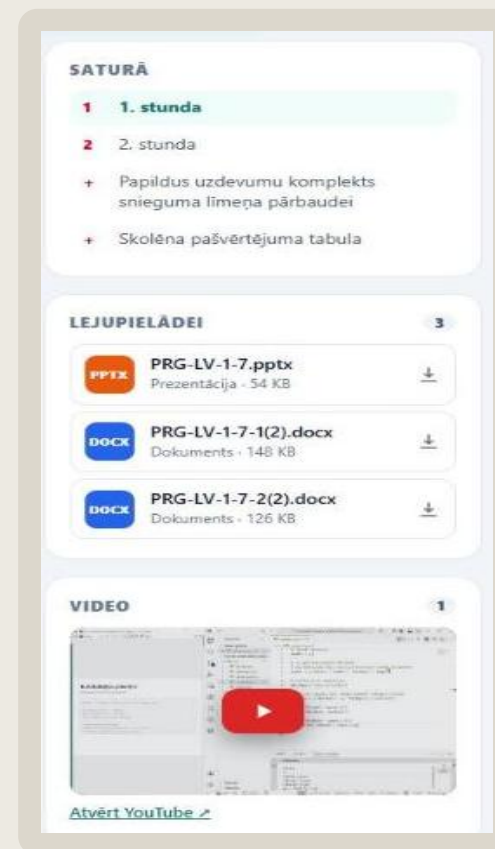
Atkārtotai — īpaši pirms zināšanu pārbaudes 1.6.

Kavētājiem

Patstāvīgai tēmas apguvei kopā ar uzdevumu lapu.

Ar skatīšanās uzdevumu

2–3 jautājumi, uz kuriem jāatbild pēc video noskatīšanās. Bez uzdevuma video var noskatīties, bet saturu neapgūt.



Ja teorija aizņem vairāk nekā 15–20 minūtes, praktiskajam darbam vairs nepietiek laika.

Talantīgajiem — papildu uzdevumi; skolēniem ar grūtībām — atbalsts: gatavs saraksts, fiksēti testdati.

Demonstrējiet tos — tie ir pārbaudīti un saskan ar uzdevumu lapām.

Jautājumi jau ir plānā — izmantojiet tos kā „izejas bileti”.

Piemērs: tēma 1.8, 1. stunda

Lietotāja ievade un tipu pārveidošana

Saturs

`input()`, `int()` un `float()`, `try/except`, `f-string` un `round()`;
praktiskais darbs ar aprēķinu pēc formulas.

Galvenais slazds

`input()` vienmēr atgriež tekstu: „5” + „3” = „53”.
Ļaujiet skolēniem to piedzīvot pašiem.

Diferenciācija

Talantīgajiem — PVN un atlaides aprēķins; skolēniem
ar grūtībām — viens datu tips.

Refleksija

„Kas notiek, ja skaitļa vietā ievada tekstu?” — tilts uz
`try/except`.

1.41. temats

Lietotāja ievade un tipu pārveidošana

2 mācību st.3 datnes1 video4.3 / 5 · 3 vērt.

★ Novērtēt materiāluPDFDrukāt

2 stundu plāni · 3 papildu sadaļasIzvērst visuSakļaut visu

1. stunda

Lietotāja veikta datu ievade un datu tipu pārveidošana Python vidē (1. daļa. Teorija un pirmais praktiskais darbs)

PDF

Stundas tēma	Lietotāja veikta datu ievade un datu tipu pārveidošana Python vidē (1. daļa. Teorija un pirmais praktiskais darbs)
Stundu skaits	1. no 2
Stundas/-u SR	- Izprot funkcijas <code>input()</code> darbības principu un datu tipu nozīmi programmā. - Prot lietot funkcijas <code>int()</code> un <code>float()</code> tipu pārveidošanai. - Prot lietot <code>try/except</code> kļūdu apstrādei. - Prot formatēt izvadi lietojot <code>f-string</code> un funkciju <code>round()</code>. - Prot izmantot zināšanas praktiskā uzdevumā ar lietotāja veiktu datu ievadi un aprēķinu veikšanu.
STUNDAS NORISE	
Aktualizācija	Skolotājs atgādina iepriekšējā tēmā apgūto par funkciju <code>print()</code> un mainīgajiem. Skolotājs uzdod jautājumu klasei: "Kā programma var iegūt informāciju no lietotāja?"
Apjēgšana	Teorijas un demonstrējumu laiks – 15 min. Skolotājs izmanto prezentāciju <i>PRG-LV-1-4.pptx</i> , kurā iekļauti piemēri par - lietotāja veiktu ievadi ar <code>input()</code>; - datu tipu pārveidošanu (<code>int()</code>, <code>float()</code>); - formatētu izvadi ar <code>f-string</code>; - ievades kļūdām.

SATURĀ

1. stunda

2. stunda

- ŠPIKERIS skolēnam – Python komandu lietošana
- Papildu uzdevumu komplekts snieguma līmeņa pārbaudei
- Skolēna pašvērtējuma tabula

LEJUPIELĀDEI

3

PRG-LV-1-4.pptx

Prezentācija · 65 KB

PRG-LV-1-4-1(2).docx

Dokuments · 109 KB

PRG-LV-1-4-2(2).docx

Dokuments · 39 KB

VIDEO

1

Atvērt YouTube ↗

Raķete, ko iznīcināja tipu pārveidošana

1996. gada 4. jūnijs · Kuru, Francijas Gviāna

Jaunās raķetes Ariane 5 pirmais lidojums. Apmēram 37 sekundes pēc starta navigācijas programma mēģināja 64 bitu peldošā komata skaitli — ar horizontālo ātrumu saistītu vērtību — ierakstīt 16 bitu vesela skaitļa mainīgajā. Skaitlis tajā neietilpa, radās kļūda, kas netika apstrādāta. Navigācija izslēdzās, raķete novirzījās no kursa un tika iznīcināta. Kods bija pārņemts no Ariane 4, kuras lidojumos šādas vērtības neradās.

SAIKNE AR STUNDU

`int()` un `float()` nav formalitāte — pārveidošana var neizdoties. Tieši tam ir `try/except`.

```
try:
    x = int(input("Skaitlis: "))
except ValueError:
    print("Tas nav skaitlis!")
```

Jautājums klasei: ko darīs jūsu programma, ja lietotājs ievadīs „pieci”?

Nosacījumu operatori un lēmumi

Saturs

Salīdzināšanas un loģiskie operatori, if/elif/else; praktiskie darbi „Skaitļu salīdzināšana” un „Veselības un fitnesa kalkulators”.

Sāciet no dzīves

„Ja līst, paņem lietussargu.” Pirms koda rakstīšanas uzzīmējiet uz tāfeles lēmumu koku.

Biežākās kļūdas

= lietošana == vietā, aizmirsts kols, nepareizas atkāpes. Refleksijā pārrunājiet šīs kļūdas.

Testējiet robežas

a == b, nulle un negatīvs skaitlis — katram zaram vismaz viens testpiemērs.

1.5 1. temats

Nosacījumu operatori un lēmumu pieņemšana programmēšanas uzdevumos

3 mācību st. 3 datnes 1 video 4,8 / 5 · 4 vērt.

Novērtēt materiālu PDF Drukāt

3 stundu plāni · 2 papildu sadaļas

Izvērst visu Sakļaut visu

Stundu plāns (3 stundas)

1. stunda

Nosacījuma operatori: matemātiskie, salīdzināšanas un loģiskie operatori. if, elif, else nosacījuma vadības plūsmas struktūra.

PDF

Stundas tēma	Nosacījuma operatori: matemātiskie, salīdzināšanas un loģiskie operatori. if, elif, else nosacījuma vadības plūsmas struktūra.
Stundu skaits	1. no 3
Stundas/-u SR	- Prot lietot matemātiskos operatorus (+ , - , * , / , // , % , **). - Prot lietot salīdzināšanas operatorus (== , != , > , < , >= , <=). - Izprot loģiskos operatorus (and , or , not) vienkāršos piemēros. - Prot uzrakstīt vienkāršu nosacījumu ar if un else . - Prot lietot loģiskās vērtības (True / False).
STUNDAS NORISE	
Aktualizācija	Saistība ar iepriekš aplūkotajām tēmām: līdz šim programmās veicām aprēķinus bez izvēles loģikas. Jautājums skolēniem – kā datoram pateikt, lai tas veic darbības atkarībā no situācijas? Piemēri no dzīves: ja ārā līst, paņem lietussargu, ja neliest, tad neņem; ja pulkstenis rāda vairāk nekā 22.00, laiks doties gulēt.
Apjēgšana	Teorija un demonstrējumi – 20 min. Aplūko prezentāciju PRG-LV-1-5.pptx (tikai tajā iekļautos piemērus). Atkārtojums par matemātiskajiem operatoriem + piemēri ar atlikuma noteikšanu (%)

SATURĀ

1. stunda
2. stunda
3. stunda

- Papildu uzdevumu komplekts snieguma līmeņa pārbaudei
- Skolēna pašvērtējuma tabula

LEJUPIELĀDEI

PRG-LV-1-5.pptx
Prezentācija · 77 KB

PRG-LV-1-5-1(2).docx
Dokuments · 149 KB

PRG-LV-1-5-2(2).docx
Dokuments · 195 KB

VIDEO

Nosacījums, kas izglāba pasauli

1983. gada 26. septembris · netālu no Maskavas

PSRS agrīnās brīdināšanas sistēma ziņoja: no ASV palaista raķete, pēc tam vēl četras. Dežurējošajam virsniekam Staņislavam Petrovam bija jāziņo par uzbrukumu. Viņš sprieda citādi: ja tas būtu īsts uzbrukums, raķešu būtu simtiem, nevis piecas, turklāt radari uzbrukumu neapstiprināja. Viņš ziņoja par sistēmas kļūmi. Viņam bija taisnība: satelīti saules atspīdumu mākoņos bija uztvēruši kā raķešu startu.

SAIKNE AR STUNDU

Programma rīkojas tikai pēc nosacījumiem, ko kāds ir paredzējis. Petrovs ņēma vērā vēl vienu nosacījumu — radaru apstiprinājumu.

```
if raketes > 0 and radars:  
    print("Trauksme!")  
else:  
    print("Pārbaudi sistēmu")
```

Jautājums klasei: kuru gadījumu jūsu programma nav paredzējusi?

Zināšanu pārbaude Nr. 1

Ko pārbauda

Tēmas 1.2–1.5: 10 uzdevumi datnē uzdevums.py, 35 minūtes; katrs uzdevums — 1 punkts.

Datnes skolotājam

Uzdevumu lapa, atbildes, paraugrisinājumi (V1 un V2), snieguma līmeņu apraksts.

Tikai tehniska palīdzība

Iepriekš pasakiet klasei: palīdzu saglabāt datni un palaist programmēšanas vidi, bet nepalīdzu rakstīt kodu.

2. stundu neizlaidiet

Rezultātu analīze un kļūdu labošana uzdevumā „Temperatūras analīze” — tieši tur notiek mācīšanās.

1.61. temats

Zināšanu pārbaude Nr. 1

2 mācību st.

Pārbaudes darbs

4 datnes

4,7 / 5 · 3 vērt.

★ Novērtēt materiāluPDFDrukāt

2 stundu plāni · 1 papildu sadaļas

Izvērst visuSakļaut visu

1

1. stunda. Zināšanu pārbaude
Zināšanu pārbaude par 1.2.–1.5. tēmu – Python pamati: `print()`, formatēšana, mainīgie, ievade, datu tipi, matemātiskie un loģiskie operatori, `if/elif/else` nosacījumi

Stundas tēma	Zināšanu pārbaude par 1.2.–1.5. tēmu – Python pamati: <code>print()</code> , formatēšana, mainīgie, ievade, datu tipi, matemātiskie un loģiskie operatori, <code>if/elif/else</code> nosacījumi
Stundu skaits	1. no 2 (40 minūtes)
Stundas mērķis	Pārbaudīt skolēnu zināšanas un praktiskās prasmes rakstīt kodu Python valodā (<code>print()</code> , mainīgo definēšanu, <code>input()</code> , datu tipu pārveidošanu, operatorus un <code>if/elif/else</code> nosacījumus), izstrādājot programmu atbilstoši uzdevumu lapai “Zināšanu pārbaudes uzdevums Nr. 1”.
Stundas sasniedzamie rezultāti (SR)	- Prot lietot <code>print()</code> teksta un rezultātu izvadīšanai, izmantojot formatēšanu (piem., <code>\t</code>, <code>f-string</code>). - Prot ievadīt datus ar <code>input()</code> un korekti pārveidot datu tipus, izmantojot <code>int()</code> un <code>float()</code>. - Prot definēt mainīgos un veikt aprēķinus, izmantojot matemātiskos operatorus. - Prot izmantot loģiskos operatorus (<code>and</code>, <code>or</code>) un nosacījumu konstrukcijas <code>if/elif/else</code>. - Spēj pārbaudīt programmas darbību, atrast kļūdas un veikt kļūdu labošanu (debug) pēc rezultātu salīdzināšanas ar uzdevuma prasībām.

SATURĀ

1

1. stunda. Zināšanu pārbaude

2

2. stunda. Rezultātu analīze un kļūdu labojums

+

Snieguma līmeņu apraksts sasniedzamajiem rezultātiem

LEJUPIELĀDEI

4

DOCX

Praktiska_darba_paugrissinajums_V1.docx

Dokuments · 27 KB

↓

DOCX

Praktiska_darba_paugrissinajums_V2.docx

Dokuments · 27 KB

↓

DOCX

PRG-LV-1-6-1.docx

Dokuments · 45 KB

↓

DOCX

PRG-LV-1-6-atb_LV.docx

Dokuments · 46 KB

↓

Pirmais īstais „bug” datorā

1947 . gada 9. septembris · Hārvarda universitāte, ASV

Datora Harvard Mark II komanda meklēja, kāpēc mašīna strādā nepareizi. F panelī, 70. relejā, atrada iesprūdušu naktstauriņu. To ielīmēja darba žurnālā un pievienoja ierakstu: „First actual case of bug being found.” Vārdu „bug” inženieri lietoja jau agrāk, bet šo stāstu gadiem ilgi stāstīja programmēšanas pioniere Greisa Hopere. Žurnāla lapa ar tauriņu šodien glabājas Smitsona institūta muzejā.

SAIKNE AR STUNDU

Kļūda nav kauns — tā ir informācija. Pārbaudes darba 2. stunda ir mūsu „tauriņa meklēšana”.

```
t = input("Temperatūra: ")
if t > 20:    # TypeError
    print("Silts")
# labojums: t = float(t)
```

Jautājums klasei: kur ir „tauriņš” šajā kodā?

Sarakstu izmantošana

Saturs

Saraksts (list), metode append(), piekļuve elementiem pēc indeksa; uzdevums „Draugu saraksts”; 2. stundā — datu apstrāde.

Aktualizācija no ikdienas

„Kā vairākas vērtības glabāt vienā mainīgajā?” — iepirkumu saraksts, klases saraksts.

Galvenais slazds

Indeksēšana sākas ar 0. Ļaujiet skolēniem apzināti izraisīt IndexError — tā šo kļūdu labāk atcerēsies.

Tilts uz 1.8

Talantīgajiem — četru vārdu ievade ar for ciklu, nevis četrām append() rindām.

1.7 1. temats

Sarakstu izmantošana

2 mācību st. 3 datnes 1 video 4.0 / 5 · 4 vērt.

★ Novērtēt materiālu PDF Drukāt

2 stundu plāni · 2 papildu sadaļas Izvērst visu Sakļaut visu

Stundu plāns (2 stundas)

1 1. stunda

ievads par Python sarakstiem (list): saraksta jēdziens, darbība un praktiska izmantošana

PDF

Stundas tēma	ievads par Python sarakstiem (list): saraksta jēdziens, darbība un praktiska izmantošana
Stundu skaits	1. no 2
Stundas/-u SR	- Izprot, kas ir saraksts (list) un kādēļ to izmanto programmēšanā. - Prot izveidot sarakstu un pievienot tajā jaunus elementus, izmantojot append() . - Prot izvadīt datus no saraksta, izmantojot indeksu (list[index]). - Spēj analizēt vienkāršu programmas piemēru un pielāgot to savām vajadzībām.
STUNDAS NORISE	
Aktualizācija	Jautā skolēniem: “Kā varētu saglabāt vairākas vērtības vienā mainīgajā?” Skolēni min piemērus no ikdienas – draugu saraksts, iepirkumu saraksts, skolēnu vārdi klasē.
Apjēgšana	Teorija un demonstrējumi – 20 min. Prezentācija PRG-LV-1-7.pptx: - saraksta jēdziens un tā pieraksts: <code>vardi = ["Anna", "Jānis", "Ilze"]</code>; - pieeja elementiem: <code>vardi[0]</code> – pirmais elements; numurēšana sākas ar 0; - jaunu elementu pievienošana: <code>vardi.append("Mārtiņš")</code>; - elementu nomaiņa un dzēšana – ievads.

SATURĀ

1. stunda
2. stunda

- Papildus uzdevumu komplekts snieguma līmeņa pārbaudei
- Skolēna pašvērtējuma tabula

LEJUPIELĀDEI

PRG-LV-1-7.pptx

Prezentācija · 54 KB

↓

PRG-LV-1-7-1(2).docx

Dokuments · 146 KB

↓

PRG-LV-1-7-2(2).docx

Dokuments · 126 KB

↓

VIDEO

1

Atvērt YouTube ↗

Kāpēc saraksts sākas ar nulli?

1982. gada 11. augusts · Nīderlande

Datorzinātnieks Edsgers Deikstra ar roku rakstīja un numurēja savas piezīmes. Viņa piezīme EWD 831 saucas „Kāpēc numerācija jāsāk ar nulli?”. Viņa arguments: ja elementus apzīmē ar $0 \leq i < N$, tad elementu skaits ir tieši N , un robeža sakrīt ar saraksta garumu. Python dara tieši tā: `vardi[0]` ir pirmais elements, bet `range(len(vardi))` beidzas tieši pirms saraksta garuma.

SAIKNE AR STUNDU

Indekss ir attālums no sākuma, nevis kārtas numurs. Pirmajam elementam attālums ir 0.

```
vardi = ["Anna", "Jānis", "Ilze"]
print(vardi[0])      # Anna
print(len(vardi))    # 3
print(vardi[3])      # IndexError
```

*Jautājums klasei: kas notiks, izpildot `print(vardi[3])`?
Aiciniet skolēnus izteikt minējumu pirms programmas
palaišanas.*

Cikla konstrukcijas

Saturs pa stundām

1. stunda — `for` un `range()`; 2. stunda — `range(start, stop, step)` aprēķinos; 3. stunda — `while`, „Temperatūras kontrole” un FizzBuzz.

Turpinājums 1.7. tēmai

Uzdevumā „Soļu skaitītājs” tiek apstrādāti saraksta elementi — saraksti un cikli jāmāca kopā.

Slazdi

`range(5)` beidzas ar 4; `while` cikls bez nosacījuma izmaiņām var kļūt bezgalīgs. Iemāciet `Ctrl+C`.

Refleksija

„Kāpēc cikls ir labāks nekā septiņas reizes pārrakstīta komanda?”

1.8 1. temats

Cikla konstrukcijas

3 mācību st. 4 datnes 1 video 4,0 / 5 - 4 vērt.

Novērtēt materiālu PDF Drukāt

3 stundu plāni 2 papildu sadaļas

Izvērst visu Sakļaut visu

Stundu plāns (3 stundas)

1 1. stunda

levads par cikla jēdzienu: atkārtotāns programmēšanā un cikla `for` darbības principi

PDF

Stundas tēma	levads par cikla jēdzienu: atkārtotāns programmēšanā un cikla <code>for</code> darbības principi
Stundu skaits	1. no 3
Stundas/-u SR	- Izprot, kas ir cikls un kāpēc programmās nepieciešama atkārtotu darbību automatizācija. - Prot lietot ciklu <code>for</code>, lai atkārtotu darbību noteiktu skaitu reižu. - Izprot funkcijas <code>range()</code> un tās parametru (sākums, beigas, solis) nozīmi. - Spēj uzrakstīt vienkāršu programmu, kurā izmanto ciklu darbības atkārtotānai.
STUNDAS NORISE	
Aktualizācija	Skolotājs uzdod jautājumu: “Kādās situācijās ikdienā kaut ko atkārtojam daudzas reizes (solji, vingrojumā, dienas ieradumi)?” Skolēni min piemērus, skolotājs stāsta, kā to var attēlot programmā.
Apjēgšana	Teorija un demonstrējumi – 20 min. Prezentācija <i>PRG-LV-1-8.pptx</i> : - cikla <code>for</code> sintakse un darbības loģika; - funkcijas <code>range()</code> piemēri: <code>range(5)</code>, <code>range(2, 8)</code>, <code>range(0, 10, 2)</code>; - cikla ķermeņa atkārtojums un izpildes secība.

SATURĀ

1. stunda
2. stunda
3. stunda

- Papildus uzdevumu komplekts
- Skolēna pašvērtējuma tabula

LEJUPIELĀDEI

PRG-LV-1-8.pptx
Prezentācija - 124 KB

PRG-LV-1-8-1(3).docx
Dokuments - 108 KB

PRG-LV-1-8-2(3).docx
Dokuments - 111 KB

PRG-LV-1-8-3(3).docx
Dokuments - 88 KB

VIDEO

1

Mazais Gauss un skaitļi no 1 līdz 100

~1786 · Braunšveiga, Vācija

Leģenda vēsta: skolotājs, gribēdams klasi ilgi nodarbināt, lika saskaitīt visus skaitļus no 1 līdz 100. Pēc brīža mazais Karls Frīdrihs Gauss nolika tāfelīti ar atbildi 5050. Viņš bija pamanījis, ka $1 + 100$, $2 + 99$, $3 + 98$... katrs pāris dod 101, un pāru ir 50. Stāsts laika gaitā izpušķots, taču matemātiskā ideja ir pareiza — tā ir lieliska ievada mīkla ciklam.

SAIKNE AR STUNDU

Klase skaitīja ar „ciklu”, Gauss — ar formulu. Dators izdara abus; salīdziniet rezultātus.

```
summa = 0
for i in range(1, 101):
    summa = summa + i
print(summa)           # 5050
print(100 * 101 // 2)  # 5050
```

Jautājums klasei: kāpēc `range(1, 101)` beigu robeža ir 101, nevis 100?

Pieci ieteikumi darbam ar materiāliem

- 1 Izlasiet metodiskos komentārus** pirms stundas — laiks, diferenciācija, datnes.
- 2 Demonstrējiet prezentācijas piemērus** — tie saskan ar uzdevumu lapām.
- 3 Ļaujiet kļūdīties kontrolēti:** „53”, `IndexError`, bezgalīgs cikls.
- 4 Video — mājās un atkārtšanai**, ar skatīšanās uzdevumu, nevis skolotāja skaidrojuma vietā.
- 5 Stāsts pie ugunsкура** — **3 minūtes** stundas sākumā, ar saikni uz stundā apgūstamo kodu.

Ko pilnveidot materiālos?

Vai 40 minūšu laika sadalījums atbilst jūsu klasei?

Kuri uzdevumi bija par grūtu vai par vieglu?

Kā jūsu skolēni izmanto video?

Vai pamanījāt kļūdas, neprecizitātes vai trūkstošu informāciju?

JAU PAMANĪTS

1.5, 1. stunda: tēmā rakstīts „`elif`” — Python valodā ir `elif`.

1.8, 1. stunda: resursos norādīts PRG-LV-1-8-1(2).docx, bet lejupielādēs — PRG-LV-1-8-1(3).docx.

Rakstiet ierosinājumus tērzētavā — tos ņemsim vērā nākamajā materiālu versijā.