

TEMATISKAIS PLĀNS

5. TEMATS

Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu? (25 stundas)

Attīstīt prasmi analizēt algoritmu efektivitāti, izmantot datu struktūras un pielietot meklēšanas un kārtošanas algoritmus, kā arī strukturēt programmas kodu, izmantojot objektorientētās programmēšanas pamatprincipus.

Tematā realizētie valsts pamatizglītības standarta sasniedzamie rezultāti

- T.O.2.4.14. Izmanto dažādas datu struktūras un gatavus algoritmus to apstrādei.
- T.O.2.4.15. Skaidro ar piemēriem jēdzienus funkcija, procedūra, klase, objekts, metode, lauks (atribūts) un notikums.
- T.O.2.4.16. Skaidro ar piemēriem jēdzienu algoritma sarežģītība. Veic vienkāršotu izstrādātās programmatūras ātrdarbības aptuvenu novērtējumu.
- T.O.2.4.19. Veido dotā uzdevuma (problēmas) risinājumu, izmantojot gatavus algoritmus un/vai pielāgojot vai kombinējot tos. Skaidro un salīdzina dažādus datu kārtošanas un meklēšanas algoritmus.

5. TEMATS. Kā izvēlēties un efektīvi savā projektā izmantot jau esošu risinājumu?		25 stundas
1	Objektorientētās programmēšanas pamati	5
2	Big O notācija	2
3	Dati šifrēšanas metodes	4
4	Zināšanu pārbaude	1
5	Meklēšanas algoritmi	5
6	Kārtošanas algoritmi	6
7	Zināšanu pārbaude	2

Atbilstība valsts vispārējās vidējās izglītības standartam

Apakštēmas Nr.	Apakštēmas nosaukums	Standarta punkti
1	Objektorientētās programmēšanas pamati	2.4.14, 2.4.15
2	Big O notācija	2.4.16
3	Zināšanu pārbaude	–
4	Meklēšanas algoritmi	2.4.16, 2.4.19
5	Kārtošanas algoritmi	2.4.16, 2.4.19
6	Zināšanu pārbaude	–

Sasniedzamie rezultāti

Zināšanas

- Zinot algoritma sarežģītību, iespējams izvērtēt programmatūras izveides iespējamību un lietojamību atkarībā no izpildāmo darbību apjoma. (T.Li.2.)
- Kārtošanas un meklēšanas algoritmi lietotājam atvieglo programmatūras lietojamību. (T.Li.2.)
- Algoritma efektivitāti raksturo izpildes laika un atmiņas patēriņa atkarība no ievaddatu apjoma. (T.Li.3.)
- Datu struktūru un algoritmu izvēle ietekmē programmas veikspēju. (T.Li.3.)
- Objektorientētā programmēšana atvieglo darbu ar kompleksām datu struktūrām un apkārtējās pasaules modelēšanu. (T.Li.2.)
- Informācijas drošība ir būtiska datu apstrādes sistēmu sastāvdaļa. (T.Li.4.)
- Datu šifrēšana tiek izmantota informācijas aizsardzībai pārsūtīšanas un glabāšanas laikā. (T.Li.4.)
- Algoritmi tiek izmantoti problēmu risināšanai un datu apstrādei. (T.Li.3.)
- Datu sakārtošana atvieglo informācijas apstrādi un analīzi. (T.Li.3.)
- Darba un programmatūras drošība ir svarīga. (T.Li.3.)

Prasmes

- Izmanto dažādas datu struktūras. (T.O.2.4.14.)
- Veic vienkāršotu programmatūras ātrdarbības novērtējumu. (T.O.2.4.16.)
- Risina uzdevumu, izmantojot gatavu algoritmu. (T.O.2.4.16.)
- Izvēlas uzdevuma izpildei atbilstošu datu kārtošanas un meklēšanas algoritmu. (T.O.2.4.19.)
- Izmanto dažādas datu struktūras. (T.O.2.4.14.)
- Strukturē programmas kodu, izmantojot klases un objektus. (T.O.2.4.14.)
- Veic datu pārstrukturizāciju un to attēlošanu, izmantojot vizualizācijas. (T.V.2.3.6.)
- Veic datu apstrādi gatavā datubāzē, izmantojot vaicājumus, formas un pārskatus. (T.V.2.3.7., T.O.2.3.4.)
- Izmantojot objektorientēto pieeju, izstrādā programmu, kura spēj šifrēt un atšifrēt datnes sensitīvu informāciju, lai nodrošinātu datu drošību. (T.V.3.1.3., T.O.2.4.3.)
- Atbilstoši situācijai izvēlas atvērtu vai šifrētu datu apmaiņu. (T.V.3.1.3.)
- Ievēro drošības nosacījumus, lietojot atvērtu datu apmaiņu. (T.V.3.1.3.)

Komplekss sniedzamais rezultāts

Izstrādā programmu, kurā tiek izmantotas piemērotas datu struktūras, izvēlēti un lietoti atbilstoši kārtošanas un meklēšanas algoritmi, novērtēta algoritmu sarežģītība un programmatūras ātrdarbība, kā arī nodrošināta datu drošība, izmantojot šifrēšanu un drošas datu apmaiņas principus. (T.O.2.4.14., T.O.2.4.16., T.O.2.4.19., T.O.2.4.3., T.V.3.1.3.)

Ierādumi

- Attīsta ieradumu sistemātiski organizēt, pārbaudīt un analizēt datus, izvērtējot to ticamību un kvalitāti. (*Tikums – gudrība; vērtība – darbs*)
- Attīsta ieradumu izmantot programmēšanu un automatizāciju ikdienas uzdevumu risināšanai un datiem pamatotu lēmumu pieņemšanai. (*Tikums – atbildība; vērtība – darbs*)

- Attīsta ieradumu atbildīgi, ētiski un droši izmantot digitālos rīkus, datu avotus un automatizācijas risinājumus. (*Tikums – atbildība; vērtība – drošība*)

Temata saturs un tā realizācija

Temata apakšdaļa	Saturs
Objektorientētās programmēšanas pamati	Skolēni apgūst objektorientētās programmēšanas pamatprincipus, veidojot izpratni par to, kā programmā strukturēt datus un darbības, izmantojot klases un objektus. Tēmas ietvaros skolēni iepazīstas ar jēdzieniem klase, objekts, īpašība un metode, kā arī apgūst šo elementu izmantošanu vienkāršu programmēšanas risinājumu izstrādē. Aktualizācijas posmā skolēni analizē ikdienā sastopamus objektus, nosauc to īpašības un raksturīgās darbības. Skolēni secina, ka līdzīgu pieeju var izmantot arī programmēšanā, lai strukturēti aprakstītu datus un to apstrādi. Apjēgšanas posmā skolēni apgūst objektorientētās programmēšanas pamatjēdzienus un veido izpratni par to, kā definēt klasi, izveidot objektu, piešķirt tam īpašības un izmantot metodes. Skolēni analizē, kā šāda pieeja palīdz organizēt programmas kodu un veidot pārskatāmākus risinājumus. Praktiskajās darbībās skolēni izstrādā vienkāršas klases, izveido objektus un izmanto tos uzdevumu risināšanā. Skolēni nosaka objektu īpašības, realizē metodes un izmanto objektus datu glabāšanai un apstrādei, pakāpeniski attīstot prasmi strukturēt programmas risinājumu. Refleksijas posmā skolēni analizē izveidotos risinājumus un izvērtē, kā klašu un objektu izmantošana palīdz padarīt programmu pārskatāmāku un vieglāk saprotamu. Skolēni formulē secinājumus par objektorientētās programmēšanas nozīmi sarežģītāku risinājumu izstrādē.
Big O notācija	Skolēni apgūst algoritmu sarežģītības jēdzienu un Big O notācijas pamatprincipus, veidojot izpratni par to, kā algoritma izpildāmo darbību skaits pieaug, palielinoties ievaddatu apjomam. Tēmas ietvaros skolēni iepazīstas ar principu, ka programmas efektivitāti būtiski ietekmē izvēlētais algoritms un ka dažādi algoritmiskie risinājumi var būt piemēroti atšķirīgiem datu apjomiem. Aktualizācijas posmā skolēni analizē situācijas, kurās darbību skaits ietekmē rezultāta iegūšanas laiku, piemēram, informācijas meklēšanu sarakstā. Skolēni secina, ka programmēšanā svarīga ir ne tikai programmas pareizība, bet arī tās efektivitāte. Apjēgšanas posmā skolēni iepazīstas ar Big O notāciju un tās izmantošanu algoritmu salīdzināšanā. Skolēni apgūst, kā raksturot algoritma sarežģītību, ņemot vērā ievaddatu apjoma pieaugumu. Praktiskajās darbībās skolēni salīdzina vienkāršus algoritmus, analizē darbību skaitu dažādos ievaddatu apjomos un sasaista iegūtos rezultātus ar Big O notāciju. Refleksijas posmā skolēni izvērtē, kā algoritmu sarežģītības izpratne palīdz izvēlēties efektīvākus risinājumus programmēšanas uzdevumos.
Meklēšanas algoritmi	Skolēni apgūst meklēšanas algoritmu pamatprincipus, veidojot izpratni par to, kā programmās tiek atrasts nepieciešamais elements datu kopā. Tēmas ietvaros skolēni iepazīstas ar principu, ka dažādi meklēšanas algoritmi ir piemēroti atšķirīgām datu struktūrām un ievaddatu apjomiem, un ka algoritma izvēle ietekmē programmas efektivitāti. Aktualizācijas posmā skolēni analizē situācijas, kurās nepieciešams atrast konkrētu elementu lielākā datu kopā, piemēram, kontaktu sarakstā vai katalogā. Skolēni secina, ka informācijas meklēšanai var izmantot dažādas pieejas.

	Apjēgšanas posmā skolēni iepazīstas ar secīgās meklēšanas un binārās meklēšanas darbības principiem. Skolēni analizē, kā datu sakārtojums ietekmē meklēšanas algoritma efektivitāti. Praktiskajās darbībās skolēni realizē meklēšanas algoritmus vienkāršos programmēšanas uzdevumos un salīdzina to darbību dažādos ievaddatu apjomos. Skolēni analizē darbību skaitu un sasaista rezultātus ar algoritmu sarežģītības novērtējumu. Refleksijas posmā skolēni izvērtē, kā meklēšanas algoritma izvēle ietekmē programmas veikspēju un secina, ka efektīvs algoritms ļauj ātrāk apstrādāt lielākas datu kopas.
Kārtošanas algoritmi	Skolēni apgūst kārtošanas algoritmu pamatprincipus, veidojot izpratni par to, kā datu sakārtošana ietekmē informācijas apstrādi un programmu efektivitāti. Tēmas ietvaros skolēni iepazīstas ar principu, ka dažādi kārtošanas algoritmi var būt piemēroti atšķirīgiem ievaddatu apjomiem un uzdevumiem. Aktualizācijas posmā skolēni analizē situācijas, kurās informācija tiek sakārtota noteiktā secībā. Skolēni secina, ka sakārtoti dati atvieglo informācijas pārskatīšanu un meklēšanu. Apjēgšanas posmā skolēni iepazīstas ar kārtošanas algoritmu darbības principiem, analizējot, kā algoritmi salīdzina un pārkārto elementus datu kopā. Skolēni veido izpratni par saistību starp kārtošanas algoritmu darbību un algoritmu sarežģītību. Praktiskajās darbībās skolēni realizē vienkāršus kārtošanas algoritmus programmēšanas uzdevumos un salīdzina to darbību dažādos ievaddatu apjomos. Skolēni novēro darbību skaitu un analizē algoritmu efektivitāti. Refleksijas posmā skolēni izvērtē, kā kārtošanas algoritmu izmantošana palīdz efektīvāk organizēt datus un uzlabot programmas veikspēju.