

## PROGRAMAVIMO MOKYMAS

### Informacinių technologijų valstybinio brandos egzamino praktinių užduočių ypatumai

#### Jonas Blonskis

Kauno technologijos universiteto docentas,  
daktaras  
Kaunas University of Technology, Assoc. Prof.  
Sukilėlių g. 112-34, LT-49240 Kaunas  
Tel. +370 615 57421  
El. paštas: jonas.blonskis@ktu.lt

#### Renata Burbaitė

Panevėžio Juozo Balčikonio gimnazijos informa-  
tikos mokytoja  
Panevėžys Juozas Balčikonis Gymnasium,  
Informatics teacher  
Respublikos g. 47, LT-35170 Panevėžys  
Tel. +370 615 92257  
El. paštas: burbaite@delfi.lt

#### Valentina Dagienė

Vilniaus universiteto profesorė, habil. daktarė  
Vilnius University, Professor, Habil. doctor  
Akademijos g. 4, LT-08663 Vilnius  
Tel. +370 698 05448  
El. paštas: dagiene@ktl.mii.lt

*Informacinių technologijų valstybinio brandos egzamino sudėtis: IT ir programavimo testas sudaro 50 %, praktinės programavimo užduotys – 50 %. Egzamino programoje numatyta, kad egzamino medžiaga turi tenkinti tokius reikalavimus: žinios ir supratimas – 30 %, praktiniai įgūdžiai – 30 % ir problemų sprendimas – 40 %. Testu tikrinamos žinios ir praktiniai įgūdžiai, praktinėmis užduotimis tikrinamas mokinių gebėjimas savarankiškai, priimti sprendimus, parašyti racionalią programą. Straipsnyje nagrinėjama užduoties sprendimo būdo parinkimo, duomenų struktūrų sukūrimo ir gebėjimų taikyti algoritmus svarba rengiant mokinius informacinių technologijų valstybiniam brandos egzaminui.*

Valstybinis informacinių technologijų egzaminas atlieka dvi esmines funkcijas: vertina mokinių pasiektus rezultatus ir stiprina jų motyvaciją mokytis programuoti (Blonskis; Dagienė, 2008). Egzamino pageidauja aukštosios mokyklos, jos suinteresuotos, kad mokiniai norėtų programuoti ir būtų įgiję bent minimalių programavimo pradmenų (Blonskis; Dagienė, 2003). Praktinių užduočių sprendimo rezultatai nusako mokinių programavimo gebėjimus. Egzaminas ne tik matuo-

ja mokinių žinias ir gebėjimus, ne tik surikiuoja mokinius stoti į aukštąsias mokyklas, bet ir iš egzamino užduočių mokosi jaunesni mokiniai, jas analizuoja mokytojai. Egzamino turinys apima duomenų struktūras, algoritmus, programavimo kalbos pamatines konstrukcijas, programavimo technologijos elementus (Informacinių technologijų brandos egzaminų programa; Informacinių technologijų valstybinių brandos egzaminų užduotys, Blonskis, Dagienė, 2006). Daug dėmesio

skiriama duomenų tipams ir duomenų struktūroms, taip pat algoritmams ir jų taikymui dirbti su duomenimis. Kad išsiugdyti kokybiškus programavimo įgūdžius, mokiniai turi projektuoti ir rašyti programas, jas derinti, testuoti. Parengtas programavimo modulis, kuriuo apibrėžiami pagrindiniai reikalavimai (1 lentelė).

1 lentelė. Programavimo modulio struktūrinė sudėtis

Programavimo mokymo dedamosios	Apimtis %
Duomenų tipai ir duomenų struktūros, duomenų struktūrų analizė ir kūrimas	20
Duomenų apdorojimo algoritmai	40
Programavimo kalbos konstrukcijos	10
Programavimo kalbos priemonės ir aplinkos	10
Programavimo technologijos	10
Programų derinimas, testavimas	5
Programavimo stilius, kultūra	5

Kartu su Anglijos ekspertais (Projektas, 2006–2008) 2007 m. buvo atlikta egzaminų kokybinė ir kokybinė analizė. Remiantis analizės išvadomis ir pasiūlymais buvo parengtos bandomojo egzamino, vykusio 2007 m. gruodžio 8 d., užduotys (2007 m. informacinių technologijų ..., 2007). Itin daug dėmesio buvo skiriama praktinių uždavinių kokybei. Jie turėjo tenkinti tris pagrindinius reikalavimus: tikrinti žinias, praktinius įgūdžius ir gebėjimą spręsti iškilusias problemas. Vertinti pirmus du punktus nėra sudėtinga ir galima formalizuoti, o trečiąjį – kūrybinį – aspektą vertinti problemiška.

Dauguma mokinių gana gerai supranta programavimo kalbos konstrukcijas ir sintaksę, tačiau sunkiai suvokia, kada ir kaip tas žinias pritaikyti. Pastebėta, kad mokiniai, ketinantys laikyti programavimo egzaminą, pasirinkę programavimo modulį mokydami stengiasi sekti egzamino modeliu: jei egzamine daug dėmesio skiriama programoms rašyti, jei yra užduočių iš algoritmų, duomenų struktūrų parinkimo, tai mokiniai ir mokytojai tam skiria daug dėmesio.

Taigi egzaminas turi mokomąją funkciją. Po kiekvieno egzamino atliekama kiekybinė anali-

zė, kuri išryškina egzaminų užduočių sunkumą ir skiriamąją gebą. Uždavinio sunkumo skaitinė reikšmė yra procentinis visų mokinių surinktų taškų santykis su visa teoriškai galimų surinkti taškų suma. Remiantis statistine testų teorija geriausi yra tie uždaviniai, kurių sunkumas apie 50 %, labai lengvų uždavinių sunkumas turi būti didesnis nei 80 %, o labai sunkių – apie 20 %. Uždavinio skiriamoji geba rodo, kaip atskiras uždavinys išskiria geriausius ir blogiausius mokinius. Jei uždavinys labai lengvas ir jį beveik vienodai sėkmingai sprendė ir geriausieji, ir blogiausieji, tai tokio uždavinio skiriamoji geba maža. Panaši skiriamoji geba gali būti ir labai sunkaus uždavinio, kurio neišsprendė taip pat beveik visi. Neigiama skiriamosios gebos reikšmė rodo, kad blogesnieji (sprendžiant pagal visą egzamino užduotį) už tą klausimą surinko daugiau taškų nei geresnieji. Tai tikrai blogos užduoties požymis.

Atliekant egzamino užduočių kokybinę analizę išskiriami mokinių pasirinkti sprendimo metodai, duomenų struktūros, gebėjimai algoritmuoti, kiekvienu atveju stengiamasi išryškinti silpnybes ir stiprybes. Pateikiami mokinių sukurti ir siūlomi kiti racionalūs uždavinio sprendimo būdai, analizuojamos tipinės klaidos ir nurodoma, kaip jų išvengti. Kokybinė egzamino užduočių analizė yra pagrindinė mokomoji priemonė mokiniams, besirengiantiems laikyti egzaminą, ir mokytojams kvalifikacijai kelti.

Mokymuisi efektyviai taikomos dr. J. Kazicko mokinių programavimo konkursinės užduotys (2009), nes daugeliui jų spręsti pakanka mokyklinių žinių, jos suteikia puikią galimybę pasirinkti sprendimo būdus ir realizacijas.

Egzamine yra dvi praktinės užduotys. Pirmoji praktinė užduotis skirta programavimo pradmenims. Ji apima veiksmus su skaičiais vienmačiame masyve, kai duomenys yra tekstiname faile ir rezultatai pateikiami tekstiname faile. Antroji praktinė užduotis skirta veiksmams su įrašo tipo duomenimis, kai duomenys skaitomi iš tekstinio failo į masyvą su įrašo tipo elementais, o rezultatai pateikiami tekstiname faile. Veiksmai atliekami tik su skaičiais.

Mokinių gebėjimą savarankiškai mąstyti geriausiai atspindi realių uždavinių sprendimo

programavimas. Čia esminiai keturi elementai: 1) sprendimo būdo parinkimas, 2) algoritmo sudarymas, 3) duomenų struktūrų sukūrimas, 4) algoritmo pritaikymas dirbti su duomenimis sukurtose duomenų struktūrose. Sprendimo būdo parinkimas svarbus tuomet, kai uždavinį galima išspręsti keliais būdais ir reikia parinkti priimtinausią. Čia turi įtakos mokinio gebėjimas susieti įvairių mokomųjų dalykų (matematikos, fizikos ir kt.) žinias ir įgūdžius. Algoritmavimo etapas yra vienas svarbesnių, nes jis lemia ne tik programos apimtį, bet ir sudėtingumą. Nuo uždavinio analizės (pradinių duomenų ir rezultatų atskyrimo) priklauso duomenų struktūrų sudėtingumas. Kuriant duomenų struktūras reikia įvertinti ne tik tai, ar jos bus patogios pradiniam duomenims ir skaičiavimų rezultatams laikyti, bet ir tai, ar bus patogiu atlikti veiksmus įgyvendinant sukurtą algoritmą. Visuose programos kūrimo etapuose pagrindinis veiksnys yra mokinio savarankiškumas, tinkamo uždavinio sprendimo būdo paieška. Čia atsiskleidžia mokinio kūrybiškumas – gebėjimas ieškoti netiesioginių algoritmavimo būdų. Tai galima pademonstruoti užduotimi, kuri pateikiama 1 priede (Dr. J. Kazicko ..., 2009).

Akcentuojant, kad tikrinamos mokinio žinios ir gebėjimai rašyti programas, matematinis uždavinio sprendimas nėra aktualus. Svarbiausia įvertinti mokinio galimybę sukurti tinkamas duomenų struktūras, atsižvelgiant į pasirinktą sprendimo algoritmą ir pritaikant algoritmą dirbti su duomenimis. Užduotis numato galimybę rinktis vieną iš kelių alternatyvių, nors ir nevienareikšmių, uždavinio sprendimo būdų. Daug sudėtingiau įvertinti parąšytą programą, užduoties vertinimo instrukcijoje rengėjams neįmanoma numatyti visų galimų mokinių pasirinkimų.

Rengiant egzaminą užduotį labai svarbu, kad mokiniai turėtų galimybę rinktis sprendimo būdą iš kelių galimų. Negali būti vienintelio racionalaus sprendimo būdo. Turi būti galimybė pasireikšti mokinio kūrybiškumui.

Panagrinėkime vieną iš užduočių (2 priedas) ir mokinių pasirinktus sprendimo būdus (2007 m. informacinių technologijų...).

Dauguma mokinių rinkosi nuoseklų užduoties sprendimo kelią. Pradinius duomenis (žaisliukų numerius) surašė į du atskirus masyvus ir surikiavo žaisliukų numerių didėjimo tvarka. Po to suformavo du masyvus, kuriuose surašė kiekvieno asmens žaisliukų numerius mainams. Čia įrašė tuos numerius, kurie duomenų masyve bent du kartus pasikartojo. Paskui iš tų masyvų pašalino tuos numerius, kurie buvo kito asmens duomenų sąrašė. Tai rezultatų sąrašai mainams. Toliau sudarė bendros kolekcijos sąrašą: abiejų pradinių duomenų masyvų reikšmes surašė į naują masyvą, surikiavo masyvo reikšmes ir pašalino pasikartojančius žaisliukų numerius. Tai paprasta veiksmų seka, kuri pareikalavo net penkių masyvų. Čia reikėjo panaudoti tris algoritmus: rikiavimo, pasikartojančių reikšmių paieškos ir reikšmės iš masyvo šalinimo.

Turintys didesnę programavimo patirtį rinkosi racionalią ir trumpesnę kelią. Duomenims saugoti skyrė du masyvus A ir B, tačiau juose surašė ne žaisliukų numerius, bet kiek kokio numerio žaisliukų turi asmuo: masyvo elemento indeksas žymėjo žaisliuko numerį, o indeksuoto kintamojo reikšmė – kiek to numerio žaisliukų yra (skaitant duomenis iš failo taikytas veiksmas:  $A[nr] := A[nr] + 1$ ; čia nr – žaisliuko numeris, A – Linos kolekcijos sąrašas; analogiškas B masyvas – Jurgio kolekcijos sąrašas):

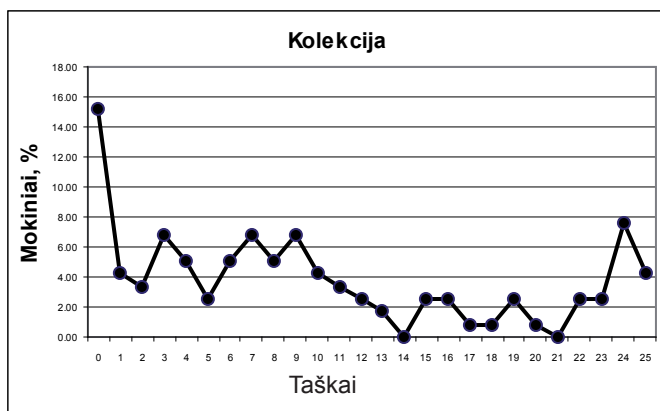
Jeigu  $A[k] = 0$ , tai reiškia, kad žaisliuko, kurio numeris k, Lina neturi. Analogiškai formuojamas Jurgio kolekcijos sąrašas.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	0	1	0	0	1	3	0	1	1	0	0	0	0	1	0	0	0	0	0	0
B	0	0	0	1	2	1	2	0	1	1	0	1	1	0	1	1	0	0	0	2

Rikiavimo ir reikšmės šalinimo iš masyvo algoritmai tapo nebereikalingi. Mainams skirti tie žaisliukai, kurių reikšmė masyve didesnė už 1. Linos mainams tinkamas žaisliukas yra tas, kurio  $A[nr] > 1$  ir  $B[nr] = 0$ . Analogiškai yra gaunamas Jurgio mainų sąrašas. Bendros kolekcijos sąrašas

sudaromas tokiu pat būdu, kaip ir asmeninių kolekcijų. Paskui reikia išspausdinti tuos numerius, kuriuos masyve atitinka nelygi nuliui reikšmė. Šis būdas paprastas, programa gerokai trumpesnė. Galimi panašūs kiti sprendimo būdai.

Tokia užduotis gerai skiria gabesnius mokinius nuo pradedančiųjų, nevaržo mokinių kūrybiškumo ir savarankiškumo renkantis sprendimo būdą. Užduotis gerai atskyrė mokinius, kurie mokydamiesi daugiau dėmesio skyrė kūrybai, o ne standartinių vadovyje pateiktų veiksmų atkartojimui. Vertinti tokias programas sudėtinga, nes panaudoti labai įvairūs sprendimo būdai, skiriasi programų apimtis. Vertinimo rezultatai vaizduojami paveiksle. Užduotį sprendė 118 mokinių, užduoties sunkumas buvo 27 %, skiriamoji geba – 69 %.



Pav. Užduoties vertinimo pasiskirstymas

Mokiniai, surinkę 0–5 taškus (37,29 %), geba atlikti tik pačius paprasčiausius veiksmus: bando aprašyti kintamuosius, užrašyti komentarus. Mokiniai, kurie surinko 6–10 taškų (27,97 %), mėgina kurti uždaviniui išspręsti reikalingus duomenų tipus, iš failo skaityti

pradinius duomenis. Mokiniais, surinkusiems 11–20 taškų (17,80 %), dažnai pritrūksta praktinių įgūdžių pasirinkti uždavinio sprendimo būdą ir jį realizuoti. Patys stipriausi mokiniai, įvertinti 21–25 taškais (16,95 %), taškus praranda dažniausiai dėl to, kad neatidžiai skaito uždavinio sąlygą.

## Išvados

Programų sudėtingumą ir apimtį iš esmės apibrėžia pasirinktų duomenų tipų sudėtingumas. Mokiniai, kurdami duomenų tipus, apie tai retai susimąsto.

Pasirinkdami praktinių užduočių sprendimo būdą, jie neapgalvoja, kuris iš galimų bus paprastesnis.

Dažnai kuriamos duomenų struktūros vien tik duomenims ir rezultatams atskirai laikyti.

Neretai pradiniai duomenys iš tekstinio failo kopijuojami į masyvus ir tik po to dalis mokinių bando juos perkelti į patogesnę veiksmams formą.

Mokiniui suteikta visiška kūrybos laisvė. Tuo puikiai pasinaudoja gabesni mokiniai, kurie geba taikyti racionalius uždavinio sprendimo būdus, pasirinkdami tinkamas duomenų struktūras ir algoritmus. Silpnesni mokiniai dažniausiai kuria tokias sudėtingas ir ilgas programas, kad

jiems nepakanka laiko jų užbaigti.

Kai kurių veiksmų ribojimas egzamino užduotyse (pvz., draudimas naudoti dvimačius masyvus) padeda sumažinti vertintojų įtaką galutiniam vertinimo rezultatui, nes vertinimo instrukcija tampa tikslesnė.

## LITERATŪRA

BLONSKIS, Jonas; DAGIENĖ, Valentina (2008). Analysis of students' developed programs at the maturity exams in information technologies. *Informatics Education – Supporting Computational Thinking* : Proceedings of the 3rd International Con-

ference on Informatics in Secondary Schools – Evolution and Perspectives, ISSEP 2008, Torun, Poland, July 1–4, 2008. Berlin; Heidelberg; New York: Springer-Verlag, 2008, p. 204–215.

BLONSKIS, Jonas; DAGIENĖ, Valentina (2006). Evolution of Informatics maturity Exams and Challenge for Learning Programming. *Lecture Notes in Computer Science*, vol. 4226: Informatics Education – the Bridge between Using and Understanding Computers. (International Conference in informatics in Secondary Schools – Evolution and Perspectives, ISSEP 2006, Vilnius, Lithuania, November 2006). Proceedings. Roland T. Mittermeir (Ed.). Heidelberg; New York: Springer, p. 220–229.

BLONSKIS, Jonas; DAGIENĖ, Valentina (2003). Programavimo pagrindų mokymo vidurinėje ir aukštojoje mokyklose lyginamoji analizė. *Informacijos mokslai*, t. 26, p. 23–28. ISSN 1392-0561.

*Informacinių technologijų brandos egzaminų programa* [žiūrėta 2009 m. rugpjūčio 18 d.]. Prieiga per internetą: <[http://www.nec.lt/failai/26\\_programa\\_IT.pdf](http://www.nec.lt/failai/26_programa_IT.pdf)>.

*Informacinių technologijų valstybinių brandos egzaminų užduotys* [žiūrėta 2009 m. rugpjūčio 18 d.]. Prieiga per internetą: <<http://www.nec.lt>>.

Projektas „Brandos egzaminų kokybės sistemos plėtra“, 2006–2008 [žiūrėta 2009 m. rugpjūčio 18 d.]. SFMIS Nr. BPD2004-ESF-2.4.0-03-05/0107. Projekto vykdytojas – Nacionalinis egzaminų centras. Konsultantas – Jungtinės Karalystės konsultavimo bendrovė „Anglia Assessment“ (direktorius G. Bethell). Prieiga per internetą: <<http://www.egzaminai.lt/49/>>.

2007 m. informacinių technologijų pilotinės (bandomosios) užduoties analizė [žiūrėta 2009 m. rugpjūčio 18 d.]. Prieiga per internetą: <[http://www.nec.lt/failai/655\\_leidiniai\\_2007\\_IT\\_pilotines\\_uzd\\_analize.pdf](http://www.nec.lt/failai/655_leidiniai_2007_IT_pilotines_uzd_analize.pdf)>.

*Dr. J. Kazicko mokinių kompiuterininkų konkursų užduotys* [žiūrėta 2009 m. rugpjūčio 18 d.]. Prieiga per internetą: <<http://forumas.ktu.lt>>.

## 1 PRIEDAS

**Dėžutės.** Atvežtas prekes dėžutėse krovėjai sudėjo į lentyną viena eile. Vadybininkas paprašė pardavėjo perdėlioti dėžutes taip, kad dėžutės, kuriose yra kompiuterio klaviatūros, būtų sudėtos iš eilės. Dėliojant galima sukeisti vienu metu bet kurias dvi dėžutes. Parašykite programą, kuri nustatytų, kiek mažiausiai keitimų reikia padaryti duotoje lentynoje, kad dėžutės su klaviatūromis būtų sudėtos iš eilės. Tekstiniame duomenų faile pirmoje eilutėje yra lentynoje esančių dėžučių skaičius  $d$  ( $1 \leq d \leq 1000$ ), o antroje yra simbolių eilutė  $d$  ilgio, kurioje raidė K žymi dėžutes su klaviatūra ir raidė O – kitas dėžutes. Rasti minimalų perdėliojimų skaičių. Pvz., jeigu turime dėžučių seką KOOKOKO, tai pakanka vienos dėžučių sukeitimo operacijos (jeigu KKOOKOK-KOOK, tai reikės 3 operacijų).

**Sprendimas.** Simbolių eilutę galima surikiuoti abėcėlės tvarka. Tačiau rikiuojant atliktas sukeitimų skaičius negarantuoja rezultato. Paprasčiausias variantas, kai perrenkami visi galimi variantai pasirenkant reikalaujantį mažiausiai sukeitimo operacijų, neracionalus, be to, programa nėra trumpa. Racionalesnis būdas gali būti toks: jeigu duotoje simbolių sekoje yra  $m$  raidžių K, tai dėliojant reikia pasiekti, kad visos K raidės eitų iš eilės. Tai reiškia, kad šioje sekoje reikia surasti tokią  $m$  simbolių seką, kurioje būtų mažiausiai raidžių O. Jų skaičius ir bus ieškomas rezultatas.

```
type Mas = array[1..1000] of char;
var L : Mas; d: integer; // Lentynos duomenys – simbolių masyvas
//-----
function Lentyna: integer;
var i, j, m, kk, s : integer;
begin
  m := 0; // Iš viso yra simbolių K
  for i := 1 to d do
    if L[i] = 'K' then m := m + 1;
    // Ieškomas minimalus O simbolių skaičius m ilgio atkarpoje.
  kk := d;
  for i := 1 to d - m + 1 do begin // Atkarpų skaičius d - m + 1.
    s := 0;
    for j := i to i + m - 1 do // Kiek atkarpoje, kurios ilgis m, yra simbolių O
      if L[j] = 'O' then s := s + 1;
      if s < kk then kk := s; // Minimalios reikšmės paieška
    end;
  Lentyna := kk; // Rezultatas
end;
//-----
```

## 2 PRIEDAS

**Kolekcija.** Šokoladinių kiaušinių žaisliukų visą kolekciją sudaro 100 žaisliukų. Kiekvienas žaisliukas turi savo numerį kolekcijoje. Lina ir Jurgis visą vasarą rinko žaisliukus. Rugsėjo pradžioje sutarė apsimainyti žaisliukais taip, kad kiekvieno turima kolekcija pasipildytų naujais žaisliukais. Mainams galima siūlyti tuos žaisliukus, kurių kolekcijoje yra daugiau kaip vienas. Tačiau mainams netinka tie žaisliukai, kuriuos jau turi draugas. Mainams negalima siūlyti kelių vienodų žaisliukų. Parašykite programą, kuri atrinktų Linos siūlomus mainams su Jurgiu žaisliukų numerius; atrinktų Jurgio siūlomus mainams su Lina žaisliukų numerius; sudarytų bendros kolekcijos žaisliukų numerių sąrašą. Numeriai negali kartotis. Siūlomų mainams žaisliukų numerius spausdinkite didėjimo tvarka tekstiniam faile. Pirmoje eilutėje spausdinkite Linos siūlomų mainams žaisliukų numerius, o antroje – Jurgio siūlomų mainams žaisliukų numerius. Jeigu kuris nors iš jų neturi nė vieno žaisliuko mainams, tuomet spausdinkite atitinkamoje eilutėje 0 (nuli). Trečioje eilutėje spausdinkite Linos ir Jurgio bendros kolekcijos žaisliukų numerius didėjimo tvarka. Numeriai negali kartotis (2 lentelė).

2 lentelė. Duomenų ir rezultatų pavyzdys

Duomenų pavyzdys	Rezultatų pavyzdys
8 14	0
5 6 6 9 14 6 8 2	7 20
5 20 6 7 13 7 9 10 12 20 15 16 4 5	2 4 5 6 7 8 9 10 12 13 14 15 16 20

## THE PECULIARITIES OF PRACTICAL TASKS OF INFORMATION TECHNOLOGY STATE EXAMINATION

Jonas Blonskis, Renata Burbaitė, Valentina Dagienė

### Summary

The composition of Information Technologies State examination (IT SE) is the following: the test of IT and programming composes 50 % and practical programming tasks – 50 %. In the examination program, it is determined that the examination material has to meet the following requirements: knowledge and understanding – 30 %, practical skills – 30 % and problem solving – 40 %. While the test examines the knowl-

edge and practical skills, the practical tasks are used to examine pupils' ability to think independently, make decisions and write a rational program. It involves selecting a way to solve the task, creating data structures and realizing algorithms to work with data in the mentioned structures, and ability to find nonstandard ways of algorithms application. Some teachers also ascribe task difficulty to the aforementioned.