View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Informacijos mokslai

INFORMACIJOS IR KOMUNIKACIJOS TECHNOLOGIJOS

Pasikeitimų informacinių technologijų valstybiniame brandos egzamine analizė

Bronius Skūpas
Vilniaus universiteto
Matematikos ir informatikos
instituto doktorantas
Vilnius University, Institute of
Mathematics and Informatics,
Doctoral student
Akademijos g. 4, LT-08663 Vilnius
Tel. +370 684 77349, faks. (8 5) 272 9209
El. paštas: *bronius.skupas@mii.vu.lt*

2011 metais informacinių technologijų valstybinio brandos egzamino vykdymo tvarka buvo esmingai pakeista – įvesta galimybė praktinę užduotį atlikti ne tik Paskalio, bet ir C++ programavimo kalba. Šis pakeitimas ne tik suteikia naujų galimybių mokiniams, bet ir sukelia tam tikrų vertinimo proceso problemų. Straipsnyje analizuojami būsimi vertinimo proceso pokyčiai, projektuojami reikiami pusiau automatinės vertinimo sistemos adaptavimo naujai situacijai poreikiai, prognozuojamos galimos vertinimo problemos ir teikiami objektyvesnio vertinimo pasiūlymai. Pirmasis egzaminas pagal naujas vykdymo ir vertinimo instrukcijas įvyko 2011 m. birželio 3 d. Analizuojama, kokią įtaką prognozuotos problemos turėjo egzamino vertinimui.

Pagrindiniai žodžiai: *informacinės technologijos, valstybinis brandos egzaminas, pusiau automatinis vertinimas.*

Įvadas

Informacinių technologijų valstybinis brandos egzaminas (ITVBE) atitinka aukštųjų mokyklų poreikius, kad stojantieji į informatikos specialybes turėtų bent minimalius programavimo pradmenis (Blonskis, Dagienė, 2003). Egzamino praktinių užduočių sprendimo rezultatai nusako mokinių programavimo gebėjimus (Blonskis, Burbaitė, Dagienė, 2009). Tačiau iki 2011 metų egzamino programa reikalavo praktines užduotis atlikti Paskalio programavimo kalba (*Informacinių technologijų valstybinio brandos egzamino programa*, 2009). Tai ribojo mokinių galimybes rinktis kitą programavimo kalbą informacinių technologijų programavimo modulyje, nors jau 2002 metais parengti Lietuvos bendrojo lavinimo mokyklos bendrosios programos ir bendrojo išsilavinimo standartai XI–XII klasei tam neprieštaravo (*Lietuvos bendrojo lavinimo mokyklos*..., 2002).

2010 metų gruodžio 13 dienos švietimo ir mokslo ministro įsakymu patvirtintas

ITVBE programos pakeitimas: „Praktinei Egzamino užduoties daliai atlikti naudotinos programavimo aplinkos Free Pascal, CodeBlocks ir Dev C++“ (*Švietimo ir mokslo ministro įsakymas dėl..., 2010*). Šis pakeitimas leidžia mokiniams praktines užduotis atlikti ne vien tik Paskalio, bet ir C bei C++ programavimo kalbomis. Pabrėžtina, kad dabartinė ITVBE programa numato dalį teorinio testo skirti programavimo Paskalio kalba užduotims ir tai neleidžia visiškai atsisakyti šios programavimo kalbos mokykloje. Tačiau ITVBE programos projekte, parengtame 2011 m. kovo 22 d., nebenurodoma, kad į testą įeis programavimo testiniai klausimai (ITVBE programos projektas 2011-03-22). Tai leidžia manyti, kad ateityje mokiniai ITVBE turės visišką laisvę pasirinkti programavimo kalbą iš trijų programavimo kalbų.

Ši galimybė skatins mokinius rinktis dabar populiaresnes C/C++ programavimo kalbas, nes žinios bus lengviau pritaikomos praktikoje, o šių kalbų sintaksė artima nemažai daliai kitų populiarių programavimo kalbų (Java, Javascript, C#, PHP). Taip pat pažymėtina, kad pastarųjų metų Lietuvos mokinių informatikos olimpiadų laimėtojai beveik visi programavo C++ kalba. Apklausa rodo, kad jiems daugiausiai imponuoja C++ kalboje esanti turtinga STL biblioteka, kokybiškos, naujoviškos programavimo aplinkos bei efektyvus kompiliuoto mašininio kodo optimizavimas.

Tyrimo tikslas

Pasikeitusi ITVBE programa turėjo paveikti pakitusias vykdymo ir (numanomai) vertinimo instrukcijas. Tačiau šie pasikeitimai neapsiriboja vien nurodymų ar instrukcijų pakeitimu. ITVBE buvo ir toliau bus vertinamas gana sudėtingu būdu – naudojant pusiau automatinę mokinių parašytų programų vertinimo sistemą. Programavimo kalbų pasirinkimo galimybė turėjo įtakos šios sistemos pritaikymo prie pakitusios situacijos darbams. Taip pat ji kelia žmogiškojo veiksnio – vertintojų parengimo vertinti ir vertinimo objektyvumo klausimus. Šiame straipsnyje analizuojamos galimos problemos ir siūlomi jų sprendimai. Egzaminui įvykus, bus patikrinta, kiek prognozės atitiko egzamino tikrovę.

Problemos ištirtumas

Mokinių parašytų programų automatinių vertinimo sistemų yra gana daug. Dažnai tos sistemos yra daugiakalbės – jas galima naudoti įvairiomis programavimo kalbomis parašytoms programoms testuoti. Tačiau tai daugiausia būdinga būtent dinaminę analizę ir konkrečiai juodosios dėžės metodą naudojančioms sistemoms. Kitokių sistemų adaptavimas kelioms programavimo kalboms reikalauja išsamios kiekvienos pridedamos programavimo kalbos analizės ir gana sudėtingų papildomų programavimo darbų, todėl toks adaptavimas vykdomas retai. Mokinių parašytų programų pusiau automatinių vertinimo sistemų pritaikomumas kelioms programavimo kalboms nėra ištirtas. „Daugiakalbių“ programavimo varžybų pasitaiko nemažai, tačiau „daugiakalbiškumas“ programavimo egzaminuose yra retas.

Apskritai, pusiau automatinių sistemų taikymo sritis, jų galimybės tebėra tyrimų objektas. Jos nėra populiarios universitetuose dėl rankinio darbo gausos. Tačiau visuotinai pripažįstama, kad automatinio juodosios dėžės vertinimo neužtenka objektyviam įvertinimui (Malmi ir kt.,

2005). Šiuo atžvilgiu Lietuvos ITVBE vertinimo aplinka yra palyginti naujoviška.

Darbo tyrimo objektas – pusiau automatinė ITVBE mokinių darbų vertinimo sistema, ITVBE vertinimo procesas, jo tobulinimo galimybės pasikeitus ITVBE vykdymo reikalavimams.

Tyrimo metodai

Atliktas empirinis tyrimas, kuriame nagrinėtos teorinės galimybės pritaikyti dabartinę ITVBE pusiau automatinę mokinių programų vertinimo sistemą pridedant papildomų kalbų. Atlikta ITVBE sistemos analizė ir modifikavimas. Ištirti Lietuvos mokinių olimpiados nugalėtojų programavimo kalbų pasirinkimai, atlikta kelių pasirinktų mokinių apklausa. Įvertintos teorinės „dvikalbystės“ įvedimo sukeltos grėsmės egzamino vertinimo objektyvumui. Parengta šių grėsmių diagnozavimo ir galimo mažinimo metodika.

Rezultatai

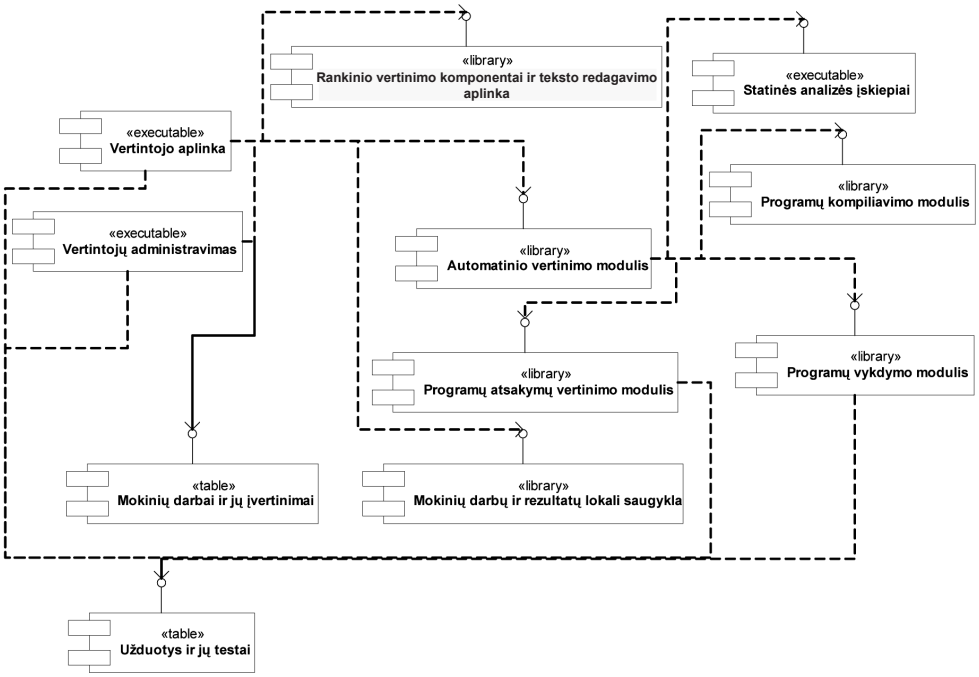
Kelių pastarųjų metų informatikos olimpiadų patirtis rodo, kad dauguma gerai pasirodančių mokinių nuo Paskalio kalbos pereina prie C++. Atlikus apklausą paaiškėjo, kad mokiniai palankiai žiūri į C++ dėl įvairių priežasčių: populiari profesionali „Microsoft“ programavimo aplinka Visual C++; greitai tobulėjančios nemokamos programavimo aplinkos bei kompiliatoriai; galingos bibliotekos; didelis panašios sintaksės kalbų populiarumas (Java, C#, Javascript, PHP); populiarėjanti su C kalba sietina „Linux“ operacinė sistema. Kai kurie mokiniai nurodė, kad C kalbą išmoko net anksčiau nei Paskalio ir pastaroji jiems nepatraukli. Apklausoje pasitaikė mokinių, kurie tvirtino, kad per artėjančius metus ketina mokytis C++, nes „ji olimpiadoje leidžia pasinaudoti standartiniais algoritmais“. Atrodo, kad ITVBE atitinka šių mokinių poreikius.

ITVBE mokinių programų vertinimo sistemos analizės rezultatai

Atlikus ITVBE mokinių programų vertinimo sistemos analizę padaryta išvada, kad sistema gana glaudžiai susijusi su Paskalio programavimo kalba: ji neskirsto darbų pagal programavimo kalbą, teksto peržiūros ir modifikavimo sistemoje naudoja Paskalio sintaksės paryškinimus, yra glaudžiai susieta su „FreePascal“ kompiliatoriumi, teksto išdėstymas įvertinamas specialiu statinės analizės įskiepiu, kuris netinkamas kitoms programavimo kalboms.

Po preliminarios sistemos analizės (žr. pav.) buvo atlikta visos sistemos išeitinio kodo bei jos modulių analizė. Buvo identifikuoti stipriai su Paskalio programavimo kalba susieti moduliai:

- **mokinių programų saugyklos modulis** turi saugoti mokinio programoje panaudotą programavimo kalbą, taip pat turi būti galimybė saugoti daugiau nei du išvesties failus, nes C++ rašytos programos pranešimus apie sistemines klaidas nukreipia į *stderr* standartinės išvesties įrenginį;
- **vartotojo sąsajoje esanti originalaus mokinio programos teksto peržiūros ir modifikavimo sistema.** Ji sudaryta iš dviejų tekstinių langų, tekstas yra spalvinamas, pagal programavimo kalbos sintaksę paryškinami baziniai žodžiai. Naudojamas teksto redagavimo kom-



Pav. ***ITVBE mokinių programų vertinimo sistemos komponentinė diagrama***

ponentas iš bibliotekos, kurį galima pritaikyti kelioms programavimo kalboms;

- **mokinio programos kompiliavimo modulis.** Svarbu naudoti programavimo kalbą atitinkantį kompiliatorių, todėl modifikavus šį modulį reikia modifikuoti ir nuostatų modulį, kuriame būtų numatyta saugoti informaciją apie sistemai pasiekiamus kompiliatorius. Kompiliatorių diagnostiniai pranešimai labai skirtingi, todėl reikia keisti diagnostinių pranešimų analizės mechanizmą;
- **mokinio programos vykdymo modulis.** Paskalis ir C++ skirtingai praneša sistemai apie įvykusią vykdymo klaidą. Taip pat C++ rašytos programos efektyviai naudoja du išvesties srautus;
- **mokinio programos statinės analizės įskiepiai** turi būti vykdomi skirtingi, priklausomai nuo programavimo kalbos.
-

ITVBE mokinių programų vertinimo sistemos modifikavimas

Modifikuojant vertinimo sistemą stengtasi atlikti tik keitimus, kurie iš principo nepakeistų vertintojo sąsajos. Taip pat stengtasi, kad sistemos funkcionalumas išliktų ne blogesnis nei buvo prieš tai. Dauguma pakeitimų vykdyta ne vertintojo dažniausiai matomoje sąsajos dalyje.

Atliekant sistemos modifikavimo darbus didžiausių sunkumų kėlė statinės analizės įskiepiai, nes daugeliu požiūriu jie labiausiai priklauso nuo programavimo

kalbos ir turi mokinio programai susikurti sintaksinį medį, jį analizuoti, rasti galimas sintaksės klaidas etc.

Dalies įskiepių atsisakyta dėl ribotų galimybių juos naudoti įvedus kelias kalbas. Tokio įskiepio pavyzdys – teksto išdėstymo tvarkingumo vertinimo įskiepis. Viena šio pasirinkimo priežasčių – C++ kalboje yra daug paplitusių teksto dėstymo būdų ir sunku apibendrinti, kas joms visoms bendra, išskyrus sistemingumą. Antra vertus, nemažai programavimo aplinkų jau realizuoja automatinį teksto dėstymo sutvarkymą. Todėl atsisakius automatinio įrankio teksto dėstymo kokybei vertinti, jei šis kriterijus bus vertinimo schemoje, jį teks vertinti rankiniu būdu.

Teorinės „dvikalbystės“ įvedimo ITVBE grėsmės vertinimo objektyvumui

Mokinių programų vertinimo objektyvumas priklauso nuo vertintojų kvalifikacijos

Vertintojų kvalifikacija negali būti objektyviai nustatyta. Keleto programavimo kalbų mokėjimas gali būti vertinamas kaip objektyvus pranašumas. Jei vertinimas pusiau automatinis, išryškėja vertintojo gebėjimai greitai ir efektyviai taisyti smulkias programavimo klaidas. Skirtingų programavimo kalbų mokėjimas negali būti visai vienodas dėl objektyvių priežasčių. Spėtina, kad kelias programavimo kalbas gerai įvaldę vertintojai gali efektyviau ir greičiau aptikti klaidas. Todėl tos klaidos gali būti vertinamos ne tiek objektyviai – kaip ne tokios rimtos.

Manytina, kad C++ kalbą kaip naują ITVBE gali būti įsisavinę mažiau vertintojų. Todėl šią kalbą pasirinkę mokiniai gali susidurti su dviem skirtingomis neobjektyvumo priežastimis:

- vertintojas programą su klaida nuvertina, nes jam dėl patirties stokos nesiseka greitai identifikuoti klaidos;
- vertintojas programą su klaida pervertina, nes jis moka ne vieną programavimo kalbą ir yra aukštesnės kvalifikacijos, todėl jam smulki klaida atrodo nereikšminga.

Pažymėtina, kad abi priežastys daugiau veikia tuos mokinius, kurių programos turi smulkių klaidų.

Galimi programavimo stiliaus vertinimo netolygumai

Skirtingose programavimo kalbose yra nusistovėję gana skirtingi požiūriai į teksto dėstymą, kintamųjų vardų formavimą, net komentavimą. Jei vertinimo instrukcijose bus numatyta nemažai taškų šiems programavimo stiliaus elementams, gali iškilti diskusijų dėl bendros nuomonės, kas laikytina tinkamu programavimo stiliumi. Šios diskusijos kyla ir Lietuvos mokinių informatikos olimpiadose. Todėl dalį patirties galbūt verta perimti iš olimpiados vertintojų.

Mokinių programų vertinimo objektyvumas priklauso nuo užduoties parengimo kokybės (Forišek, 2006). Kuriant užduotį reikia atsižvelgti į tai, kad skirtingose programavimo kalbose yra skirtingos standartinės bibliotekos. Todėl tam tikrų užduočių sudėtingumas gali smarkiai skirtis. Geras pavyzdys galėtų būti žodžių, atskirtų tarpais, skaitymas iš tekstinio failo. Paskalio kalboje tam nėra numatyta standartinių priemonių, todėl reikia skaityti po simbolį arba skaidyti į žodžius nuskaitytą visą eilutę. C kalbos bibliotekos *fscanf()* funkcija šį

darbą atlieka be papildomų pastangų. Galima rasti ir daugiau pavyzdžių: standartiniai rikiavimo algoritmai su C funkcija *sort()* ar rikiuotomis duomenų struktūromis iš STL bibliotekos.

Į tai neatsižvelgus gali susidaryti situacija, kuri iš dalies pasitaiko ir Lietuvos informatikos olimpiadose, kai dauguma olimpiados nugalėtojų yra pasirinkę C/C++ kalbas. Gal šios kalbos pasirinkimas ir nėra bėda olimpiadose, tačiau tai gali kelti grėsmę egzaminui dėl nelygiavertės egzamino užduoties skirtingų mokyklų mokiniams, nes ne visose mokyklose yra mokytojų, pasirengusių dėstyti šią kalbą.

Galimi techniniai vertinimo nesklandumai

Net esant gerai parengtai sistemai ir kokybiškoms užduotims, dirbant tik su Paskalio programomis kildavo tam tikrų techninių nesklandumų. Nemažai jų buvo susiję su „FreePascal“ leidžiamais skirtingais dialektais. Įdiegus „FreePascal“ standartiškai nustatomas FPC dialektas, kuris šiek tiek skiriasi nuo mokyklose labiausiai paplitusio istorinio „Turbo Pascal 7.0“ dialekto (TP7). O dar yra OBJFPC, DELPHI dialektai... Bėda ta, kad dalis mokinių naudoja įvairias konstrukcijas, būdingas tik tam dialektui, ir kompiliuojant kito dialekto režimu dažnai nepasiseka programų sukompiliuoti sėkmingai (Skūpas, Dagienė, 2010). Gana populiarios konstrukcijos, kai funkcijos vardas jos viduje naudojamas kaip kintamasis. TP7 dialekte tai neleistina, o jei funkcijos vardas pasitaiko dešinėje priskyrimo pusėje – tai interpretuojama kaip rekursinis kreipinys! Tokios problemos dažnai sukelia vertintojų diskusijų, ar tai mokinio klaida.

Įvedus C++ kalbą iš tiesų techninių nesklandumų galima sulaukti daugiau. Priežastis ta, kad pasirinkta ne viena C++ programavimo terpė, o dvi. Jose integruoti skirtingų kartų C++ kompiliatoriai, kurie turi skirtingas galimybes, turi ir skirtingų klaidų. Todėl gali tekti aiškintis, ar programa veikia vienodai, sukompiliuota skirtingais kompiliatoriais. Taip pat neužmirštinas ir dar vienas aspektas – šios aplinkos tinka sėkmingai programuoti ir grynąja C kalba. Dažniausiai tai galima atpažinti iš kitokio bylos prievardžio (.c) ar įtraukiamų bylų (.h). Tačiau mokinių patirtis nėra didelė, dauguma jų bus savamoksliai C/C++ programuotojai, todėl kils nesklandumų dėl supainiotų prievardžių ar įtraukų bylų.

Siūlomi sprendimai

Vertintojų kvalifikacijos problemą galima spręsti keliais skirtingais būdais. Šiais metais galimas sprendimas yra trumpi mokymai, kuriuose būtų pristatyti C/C++ kalbų pagrindai. Kadangi spėtina, jog dauguma mokinių naudosis mokyklose įprasta procedūrinio programavimo paradigma, nesudėtingų programų vertinimas gali būti ne toks sunkus, jei programos bus trumpos, paprastos, naudos mažai konstrukcijų. Išlieka pavojus, kad dalis mokinių naudosis objektiniu programavimu, kurį dauguma mokytojų nėra gerai įvaldę. Tačiau spėtina, kad dauguma šią programavimo paradigmą įvaldžiusių mokinių turi gerus programavimo įgūdžius ir jiems nereikės alternatyviojo vertinimo.

Kitas galimas kelias – skaidyti vertintojų grupę į mokančius C/C++ ir nemokančius. Tuomet pateikti C/C++ sprendimai pakliūtų tik tiems, kurie gerai moka šias

kalbas. Tačiau kyla jau minėtų grėsmių dėl skirtingos vertintojų kvalifikacijos.

*Techniniams nesklandumams mažinti v*ertinimo sistema parengta dirbti su skirtingais kompiliatoriais, tačiau net ir nedideli nesklandumai gali lėtinti darbą. Todėl vertinimo greičio požiūriu C/C++ sprendimus būtų geriau išskirti atskirai analizei. Išskirtas būdingas klaidas būtų gerai pateikti kitiems vertintojams. Kita vertus, prieš imantis vertinimo, vertintojams turėtų būti parodyti esminiai bruožai, būdingi C kalbai ir konstrukcijos C++ kalbai. Tokių bruožų pavyzdys – būdingi įterpti .h failai, tipiškų funkcijų naudojimas.

Rekomendacijos užduočių rengėjams greičiausiai buvo parengtos pagal Nacionalinio egzaminų centro metodikas, tačiau keletą minčių visgi galima pasiūlyti. Gal jos bus pritaikytos rengiant 2012 metų egzaminų užduotį. Taigi, norint parengti užduotį, kuri būtų panašaus sunkumo naudojantiems skirtingas programavimo kalbas, reikėtų:

- sąlygas formuluoti labai tiksliai, nurodant, kurias konstrukcijas, funkcijas galima naudoti, kurių ne;
- sukurti programavimo kalboms bendrą aprašą ir jį paskelbti prieš egzaminą, kuriame būtų fiksuotas egzaminui galimas naudoti programavimo kalbos poaibis. Šis aprašas palengvintų ir mokinių, ir vertintojų darbą;
- kurti sąlygas labai atidžiai, analizuojant visus galimus sprendimus, ir naudoti tik tas užduotis, kurios galbūt neturi esminių realizacijos skirtumų.

Situacijos įvertinimas po egzamino

2011 m. birželio 3 d. įvykęs pirmasis naujo tipo egzaminas atskleidžia keletą dalykų, kurių neįvertinto tyrimas. Vienas būtų toks: nors egzamino programoje nebuvo aiškiai detalizuota, tačiau teoriniai klausimai buvo pateikti naudojant pavyzdžius, parašytus abiem kalbomis. Tai leidžia mokiniams visai nemokėti Paskalio. Vertintina kaip teigiamas žingsnis, nors kai kuriems mokiniams sukėlė abejonių, ar užtenka atsakyti į klausimus tik pasirinktai kalbai (tai buvo egzamino nurodymuose ant užduoties sąsiuvinio viršelio, bet neatidžių mokinių pasitaikė).

C++ kalba pateikta 59 praktinių užduočių sprendimai: 32 pirmosios praktinės užduoties ir 27 antrosios. Šiuos sprendimus pateikė 33 kandidatai iš 1270 pateikusiųjų bent vieną praktinės užduoties sprendimą. Taigi, C++ kalbą pasirinko dar labai mažai abiturientų. Todėl vertinimo problemų tai beveik nekėlė – buvo pasirinkta strategija, kad šiuos darbus vertins tik C++ kalbą gerai mokantys ir savimi pasitikintys vertintojai. Tokių vertintojų buvo 5 iš 28. Pabrėžtina, kad trečiam vertinimui C++ buvo iškelta tik 6,8 proc. (palyginti su 15,0 proc. Paskalio darbams). Manytina, kad tai patvirtino prielaidą, jog C++ darbus vertino aukštesnės kvalifikacijos vertintojai. Tačiau neatmestina ir mintis, kad šiais metais C++ rinkosi stipresni abiturientai – jie už praktines užduotis surinko vidutiniškai po 21,1 taško (palyginimui: rinkęsi Paskalį – 11,2 taško). Manytina, kad vertinimo galimo netolygumo prielaidos nei patvirtintos, nei paneigtos, nes C++ šiais metais rinkosi per mažai abiturientų, kad darytume galutinę išvadą.

Išvados

Informacinių technologijų valstybinis brandos egzaminas atsidūrė tam tikroje kryžkelėje. Švietimo ministro įsakymu įvesta galimybė užduotis atlikti kitoje programavimo aplinkoje reikšmingai pakeičia egzamino esmę. Šis pakeitimas suteikia naujų galimybių mokiniams, skatina juos mokytis įvairių programavimo kalbų, įskaitant ir savarankišką mokymąsi, didina jų motyvaciją.

Egzamino pakeitimas taip pat sukuria naujų iššūkių. Didžiausi uždaviniai iškelti užduočių kūrėjams, vertinimo sistemos kūrėjams ir vertintojams.

Daugiausia įtakos vertinimo objektyvumui turės vertintojų kvalifikacija ir jų pasirengimas dirbti keliomis programavimo kalbomis. Labai svarbus bus darbo organizavimas, supažindinimas su kitų vertintojų priimtais vertinimo sprendimais.

Pusiau automatinės sistemos pritaikymas prie pasikeitusios situacijos sukelia problemų dėl to, kad nėra galimybių naudoti adekvačius statinės analizės įskiepius. Automatinio vertinimo dalies modifikavimas privertė peržiūrėti nemažą sistemos dalį, tačiau tai neturėtų sukelti vertinimo problemų. Techninių keblumų gali kelti kompiliatorių realizacijų bei dialektų skirtumai.

Įvertinti užduočių kūrėjų darbo pasunkėjimą kebliausia. Jis priklauso nuo konkrečių darbų vykdytojų kompetencijos ir išsilavinimo. Bendram egzamino rezultatui užduočių kūrėjų darbo kokybės įtaka negali būti nuvertinta ir, be abejo, sukels IT mokytojų diskusijų.

LITERATŪRA

ALA-MUTKA, Kirsti M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education* vol. 15, p. 83–102.

BLONSKIS, Jonas; BURBAITĖ, Renata; DAGIENĖ, Valentina (2009). Informacinių technologijų valstybinio brandos egzamino praktinių užduočių ypatumai. *Informacijos mokslai*, t. 50, p. 136–141. ISSN 1392-0561.

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

FORIŠEK, Michal (2006). *On* the suitability of tasks for automated evaluation. *Informatics in Education,* vol. 5(1), p. 63–76.

Informacinių technologijų valstybinio brandos egzamino programa (2009) [žiūrėta 2011 m. balandžio 22 d.]. Prieiga per internetą: <*http://www.nec.lt/failai/1257_programa_IT_2009_09_04.pdf*>.

Informacinių technologijų valstybinio brandos egzamino programos projektas 2011-03-22 [žiūrėta 2011 m. balandžio 22 d.]. Prieiga per internetą: <*http://www.pedagogika.lt/index.php?-73619855*>.

KOLCZYK, Ewa (2009). Examiner's Remarks on Informatics Matura Examination in Poland. *Informatics in Education*, vol. 8, no. 2, p. 251–260.

Lietuvos bendrojo lavinimo mokyklos bendrosios programos ir bendrojo išsilavinimo standartai XI–XII klasei (2002) [žiūrėta 2011 m. balandžio 22 d.]. Prieiga per internetą: <*http://www.pedagogika.lt/index.php?-1469555137*>.

MALMI, Lauri; KARAVIRTA, Ville; KORHONEN, Ari; NIKANDER, Jussi (2005). Experiences on automatically assessed algorithm simulation exercises with different resubmission policies. *Journal on Educational Resources in Computing (JERIC)*, September, vol. 5 no. 3, p. 7

SKŪPAS, Bronius; DAGIENĖ, Valentina (2010). Observations from semi-automatic testing of program codes in the high school student maturity exam. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research (Koli Calling '10)*. ACM, New York, NY, USA, p. 31–36.

Švietimo ir mokslo ministro įsakymas dėl švietimo ir mokslo ministro 2005 m. liepos 21 d. įsaky-

mo Nr. isak–1524 „Dėl informacinių technologijų brandos egzaminų programos patvirtinimo“ pakeitimo, 2010 m. gruodžio 13 d., Nr. V-2309 [žiūrėta 2011 m. balandžio 22 d.]. Prieiga per internetą: <http://www.nec.lt/failai/1833_2010-12-13-Pakeitimai_IT_VBE_programoje.pdf>.

ANALYSIS OF CHANGES IN THE NATIONAL MATURITY IT EXAM

Bronius Skūpas

S u m m a r y

The national maturity IT exam was changed in 2011 by adding the possibility to use C and C++ programming languages in practical assignments. Before 2011, it was possible to use only the Pascal programming language. This change leads to wide possibilities for students and involves great changes in the evaluation process as well. The semi-automatic evaluation system must be changed accordingly, and more attention to the selection of tasks, tests and even evaluators must be given. The author analyses the necessary changes in the evaluation process and system. He predicts the possible problems, analyses them and provides their possible solutions. The first exam of this style took place on 3 June 2011. Its results are compared with the prognosis.