УДК 378.14

СЕРГІЙ МАРТИНЮК, ЯРОСЛАВ ВАСИЛЕНКО, РОМАН ДІТЧУК

ПРОВЕДЕННЯ ON-LINE ОЛІМПІАД З ІНФОРМАТИКИ

Досліджено практику, історичні аспекти та перспективи проведення шкільних і вузівських on-line олімпіад з інформатики, конкретизовано вимоги щодо добору завдань, їх перевірки за допомогою системи тестів. Описано можливості використання розробленої системи віддаленого проведення олімпіад з web-інтерфейсом і тестування розв'язків завдань учасників турнірів. Обґрунтовано перспективи подальших досліджень з даного питання.

Ключові слова: *on-line олімпіада, програмування, задачі, тестування, система.*

СЕРГЕЙ МАРТЫНЮК, ЯРОСЛАВ ВАСИЛЕНКО, РОМАН ДИТЧУК

ПРОВЕДЕНИЯ ON-LINE ОЛИМПИАД ПО ИНФОРМАТИКЕ

Исследованы практика, исторические аспекты и перспективы проведения школьных и вузовских on-line олимпиад по информатике, конкретизированы требования к подбору заданий, их проверки при помощи системы тестов. Описаны возможности использования разработанной системы отдаленного проведения олимпиад с web-интерфейсом и тестирования решений заданий участников турниров. Обоснованы перспективы дальнейших исследований по данному вопросу.

Ключевые слова: *on-line олимпиада, программирование, задачи, тестирование, система.*

SERGIY MARTYNYUK, YAROSLAV VASYLENKO, ROMAN DITCHUK

ON-LINE COMPETITIONS IN COMPUTER SCIENCE

The practice, historical aspects and perspectives of school and university on-line competitions in computer science are investigated; requirements of tasks choice and checking them by means of tests system are specified. The opportunities of usage the remote system of organizing competitions with web-interface and testing system of participants are described. The perspectives for further research on the subject are grounded.

Key words: *on-line competition, programming, problems, testing, system.*

Олімпіади є одним з об'єктивних показників визначення продуктивності навчальної й позакласної роботи з інформатики. Викладачі зацікавлені в одержанні таких об'єктивних даних шляхом порівняння рівня знань «своїх» учасників з іншими. Тому вони з інтересом стежать за результатами таких заходів.

Метою статті є дослідження сучасної практики й історичних аспектів проведення олімпіад з інформатики, аналіз наявних систем проведення змагань з програмування та перевірки їх результатів, а також опис розробленої системи віддаленого проведення олімпіад з web-інтерфейсом і програми тестування розв'язків завдань.

Традиційно під олімпіадами з інформатики розуміють олімпіади з програмування. Можна вказати на дві основні причини такого вузького розуміння інформатики стосовно олімпіад. Насамперед у часи «до персональних комп'ютерів» роботу з ПК вважали програмуванням. Викладання ж програмування зводилося до вивчення алгоритмів, оскільки алгоритмізація тривалий час була найактуальнішим напрямом програмування. Такий же вузький зміст був закладений у перший шкільний підручник А. П. Єршова. Уже перші олімпіади ще в часи безмашинного викладання інформатики були присвячені розв'язанню задач на створення алгоритмів на шкільній алгоритмічній мові.

Інерція такого розуміння серед неспеціалістів не переборена й досі, не випадково фахівців з використання і застосування комп'ютерів на підприємствах й сьогодні найчастіше називають програмістами.

Іншою, більш важливою причиною, за якою з усіх розділів інформатики для олімпіад вибирали програмування, є те, що олімпіада з програмування універсальна в тому розумінні, що відмінності, пов'язані з використанням учасниками різного інструментарію — обчислювальної техніки, мов програмування і середовищ програмування різної потужності — за правильного добору задач є другорядними порівняно з рівнем загального алгоритмічного мислення, досвідом і навичками програміста, знанням можливостей мови програмування, навичками формалізації задачі. З цієї точки зору програмування близьке до математики та фізики. Не випадково саме з цих предметів найбільш усталені традиції проведення олімпіад на всіх рівнях — починаючи від шкільних і міських (районних) і до міжнародних. Саме універсальність олімпіад із програмування є причиною їх популярності.

Високо оцінюючи значення олімпіад з інформатики, потрібно мати на увазі, що програмування не є основним змістом шкільного курсу інформатики. Тому школяр практично не може показати успішні результати на такій олімпіаді, спираючись тільки на знання, отримані на уроках з інформатики. Досвід свідчить, що впоратися з олімпіадною задачею можуть тільки ті школярі, які одержали додаткову підготовку з програмування, що виходить за рамки стандартної шкільної програми з інформатики.

Учителеві інформатики в будь-якому випадку важливо стежити за підсумками олімпіади, за змістом олімпіадних задач. Від нього не вимагається обов'язково використовувати ці задачі на звичайних уроках. Однак він може зробити деякі загальні висновки і на їх основі скорегувати зміст курсу інформатики. Олімпіади взагалі займають особливе місце серед інших організаційних форм навчання. Метою проведення олімпіад з інформатики є пошук обдарованих, розвинутих та освічених школярів; оцінка рівня викладання інформатики у школі в цілому; пред'явлення граничного рівня вимог, орієнтира при вивченні інформатики; реалізацію зворотного зв'язку ланцюга «школа — ВНЗ».

Дещо з історії шкільних олімпіад з інформатики. Навесні 1985 року була прийнята Постанова «Про заходи по забезпеченню комп'ютерної грамотності учнів середніх навчальних закладів і широкого впровадження електронно-обчислювальної техніки в навчальний процес», а вже з осені 1985 р. в усіх школах країни розпочалося викладання курсу «Основи інформатики та обчислювальної техніки». До вирішення складних завдань викладання шкільної інформатики відразу долучилися видатні вчені академіки А. П. Єршов, Є. П. Веліхов, Б. М. Наумов та інші. Завдяки цьому за доволі короткий термін у країні сформувалися колективи, які могли, опираючись на освітню, наукову, промислову та культурну комп'ютерну інфраструктуру, вирішувати поставлені в освіті завдання за короткі терміни.

Початок проведення олімпіад з інформатики був наступним важливим кроком у створенні інфраструктури викладання інформатики в школі, оскільки для інтенсивного руху країни в напрямі інформатизації освіти було явно не достатньо. Виникла потреба у висококваліфікованих фахівцях, здатних розробляти інформаційні технології завтрашнього дня. Зараз важко сказати, у кого першого виникла ідея проведення Всесоюзних олімпіад школярів з інформатики, але цілком очевидно, що цей предмет швидко розвивався і не міг тривалий час залишатися без олімпіади. Восени 1987 року в Міністерстві освіти СРСР відбулася перша організаційна нарада, на якій були присутні академіки А. П. Єршов, М. М. Красовський, А. Л. Семенов, В. М. Кирюхін, а також представник міністерства освіти СРСР і член Центрального оргкомітету Всесоюзної олімпіади школярів Т. А. Саричева. На нараді було прийнято рішення провести першу в країні олімпіаду школярів з інформатики навесні 1988 року в м . Свердловськ (нині Єкатеринбург).

Свердловськ не випадково був обраний місцем проведення першої олімпіади: у той час у багатьох школах міста і Свердловської області вже були персональні комп'ютери, розроблена сучасна на той час програма та підручники для викладання шкільної інформатики. На першій організаційній нараді було погоджено також Положення про олімпіади з інформатики та призначені голови програмного комітету та журі. Головою програмного комітету став академік А. П. Єршов, головою журі — академік М. М. Красовський. Перша олімпіада з інформатики,

яка відбулася з 13 до 20 квітня 1988 року в Свердловську, носила назву Всесоюзної, у ній взяли участь 80 школярів з усіх союзних республік.

У той час досвіду в організації таких змагань не було ні в країні, ні у світі. Для того щоб визначитися з методикою і змістом олімпіад з інформатики, до складу журі було запрошено кращих на той час фахівців у галузі шкільної інформатики й олімпіадного руху, по одному представнику від кожної союзної республіки. У результаті тривалих суперечок і обговорень поступово формувалися ті правила, які лягли в основу правил проведення сучасних олімпіад. Кількісний склад учасників перших олімпіад визначався з урахуванням наявних можливостей у забезпеченні комп'ютерами і пропорційно до численності школярів у союзних республіках. Починаючи з III Всесоюзної олімпіади, яка пройшла в 1990 році в м. Харкові, було вирішено проводити два тури олімпіади з використанням комп'ютерів. До цього I тур був теоретичним, без використання комп'ютерів, II тур — практичним.

Олімпіада, яка пройшла в 1992 році в м. Могілів, носила назву Міждержавної, у ній взяли участь школярі практично з усіх держав, які утворилися після розпаду СРСР.

Паралельно із Всесоюзними, проходили і Всеукраїнські олімпіади. Перша з них відбулася 1989 року у м. Чернівці і відтоді проходить щороку в різних містах України. Завдання учасників полягає в написанні розв'язків певних алгоритмічних задач на одній з дозволених мов програмування. Олімпіаду проводять у 2 тури, після першого відбувається тестування розв'язків учасників на встановленому наборі тестів і оголошуються проміжні результати. Кількість задач у кожному турі може бути різною, але зазвичай учасникам пропонують 3 задачі різної складності, рідше — 2 або 4. Кількість балів за задачу залежить від того, скільки тестів «пройшов» розв'язок. Вважається, що розв'язок пройшов даний тест, якщо відповідь збігалася з очікуваною, і при цьому в процесі виконання програма не перевищила встановлені ліміти часу та пам'яті.

За час проведення міжнародних і всеукраїнських олімпіад з інформатики та програмування для школярів і студентів накопичений величезний організаційний досвід, налагоджена взаємодія різних ланок у системі підготовки обдарованих молодих фахівців у галузі інформатики й інформаційних технологій, що роблять вагомий внесок у розвиток інформатизації країни. З учнями працювали висококласні фахівці та педагоги, орієнтовані не тільки на безпосередній результат, тобто на призові місця для своїх підопічних, але і на тривалу перспективу — на виховання майбутньої зміни фахівців у галузі інформаційних технологій і програмування.

Вимоги щодо добору завдань. Олімпіади з інформатики мають особливі вимоги щодо добору чи розробки задач. Вкажемо вимоги до таких задач:

- задача повинна бути невідомою для учасників олімпіади. З цієї ж причини слід уникати вибору задач із загальнодоступної та популярної літератури з програмування. В ідеалі задачі мають бути авторськими, але для цього потрібна наявність особливих навичок та досвіду в розробників таких задач. Тим паче, що розробка власних задач для олімпіади не є непосильною. Річ у тім, що нові задачі можна отримати шляхом переробки відомих задач;
- розв'язання задачі не має базуватися тільки на використані особливих знань. Якщо такі задачі й пропонуються, то тільки серед деяких задач за наявності альтернативних — із прийнятними для всіх умовами;
- олімпіадна задача повинна містити оригінальну ідею, яка вимагає від учасників використання нетрадиційних підходів до її розв'язання. Олімпадні задачі мають бути не тільки формально новими стосовно тексту, але й новими для школяра за суттю, змістовим наповненням, підходом до розв'язування, методами та засобами, які використовуються. У ході розв'язування такої задачі складністю повинні бути не технічні проблеми відомого методу для цієї задачі, а творчий процес винайдення самого методу;
- якщо ж правильність програм визначають шляхом тестування, то від задачі вимагається максимально чітке формулювання, яке виключає будь-яку двозначність.

Існує варіант, коли на олімпіаді пропонується не одна, а кілька задач, і кожний учасник має право самостійного вибору. На перший погляд, це дозволяє уникнути провалу для тих учасників, які не ознайомлені з тематикою, і тому не змогли осилити й однієї задачі. Однак у цьому випадку виникають одразу дві проблеми. Перша проблема пов'язана з тим, як оцінювати

результати. Практично неможливо передбачити, яка задача виявиться для учасників складнішою або легшою. Тим самим з самого початку ми ставимо учасників у неоднакові умови. Крім цього, перед учасниками постає питання вибору задачі. Замість того, щоб зосередитися на розв'язуванні конкретного завдання, вони змушені робити вибір, втрачаючи дорогоцінний час. Зменшення числа задач до однієї за умови одночасного її «розгалуження» є чіткою стійкою тенденцією. Бажано добирати завдання, при розв'язуванні яких програма використовує тільки текстовий режим, оскільки графічні можливості різних систем програмування дуже відрізняються. Задачі повинні бути ще й такими, щоб при розв'язувані можна було б обійтися засобами, наявними в усіх мовах програмування. В аспекті структур даних до таких універсальних засобів належать, крім простих типів даних, масиви та стрічкові величини.

Аналізуючи тематику задач олімпіад різних років, можна зазначити, що відбувається відхід від задач обчислювального характеру. Помітно, що такі задачі значною мірою потребують наявності математичного типу мислення і тому деякою мірою не відповідають вимогам олімпіад з інформатики. З іншого боку, відомо, що на олімпіадах з фізики останніх років почали з'являтися задачі, які потребують використання обчислювальної техніки.

Таким чином, виникає природній процес розподілу різноманітних комп'ютерних умінь на ті, які є предметом змагань на олімпіадах з інформатики, та на ті, які слугують інструментом для інших наук.

Системи перевірки завдань. Важливою особливістю олімпіади з інформатики є те, що тексти створених програм чи алгоритмів-розв'язань задач перевіряти значно важче, ніж розв'язання задач на олімпіадах з математики чи фізики. Досвід свідчить, що для цього необхідно задіяти більшу кількість фахівців високої кваліфікації з програмування, які взаємно контролюють один одного.

Виконати перевірку олімпіадних задач шляхом аналізу тексту програми доволі важко, навіть якщо вимагати від учасника подання повних коментарів. В умовах нестачі часу учасники квапляться писати програму, а не коментарі, залишаючи їх створення на потім. Коментарі, достатні з точки зору автора, журі може визнати недостатніми. Іноді перевірка тексту програми взагалі неможлива. Тому зазвичай вихід з цієї ситуації — тестування програм за допомогою набору тестів без аналізу її змісту. Тестування в таких випадках є єдиним виходом, хоча слід мати на увазі й негативні аспекти такого підходу перевірки правильності розв'язання.

Використання тестів не врятує ситуацію повністю, оскільки, з одного боку, важко стверджувати про їх достатність, а з іншого — виникає запитання: як оцінити програму, якщо тільки частина тестів «проходить» або «не проходить» ні один тест, коли програма містить синтаксичні помилки чи розв'язує тільки частину задачі.

Набір тестів повинен охоплювати критичні випадки вхідних даних. Залежно від того, скільки тестів успішно «пройшла» програма, автор отримує різну кількість балів. Як додаткові параметри, можна враховувати рівень відпрацьованості алгоритму, ефективність розв'язування, врахування граничних та особливих умов, зрозумілість описання програми.

На Всеукраїнських шкільних олімпіадах, тестуючи програму на наборах вхідних тестів, рахують бали окремо за кожен пройдений тест. Зазвичай за задачу дають 100 балів і її перевіряють на 20-ти тестах, тобто нараховують по 5 балів за один пройдений тест.

Але тут виникає ситуація, коли деякі учасники, знаючи про такий підхід до оцінювання, та не маючи можливості розв'язати якесь із завдань, пишуть програму, яка видає результат лише для граничних випадків вхідних даних (обчисливши їх вручну), адже не важко здогадатись, що серед тестів траплятимуться такі випадки. У деяких випадках такому учасникові вдавалось одержати 5–30 «нечесних» балів на одній задачі.

На сьогодні для проведення турнірів з інформатики використовують дві системи: Ejudge і Ccontester.

Система Ejudge. До її переваг відносять:

- обмежені й необмежені за часом турніри;
- одночасне проведення кількох турнірів;
- автоматична реєстрація учасників турніру;
- можливість участі в кількох турнірах під одним реєстраційним ім'ям;

- розподіл прав доступу до турнірів. Деякий користувач може бути адміністратором одного турніру і не мати жодних привілеїв у іншому;
- багатомовний інтерфейс. Поточна версія підтримує російську й англійську мови;
- захищене виконання програм;
- підтримка варіантних задач, коли під одним ім'ям кожен учасник отримує свій варіант завдання;
- web-інтерфейс адміністратора й учасника турніру;
- web-інтерфейс адміністратора турніру для створення нових турнірів і редагування налаштувань турнірів, які вже існують.

Система Contester може функціонувати під управлінням ОС Windows або Linux. До можливостей учасника у системі відносять:

- самостійна реєстрація;
- перегляд списку своїх спроб вирішення, імена відправлених файлів, вихідні коди кожної спроби;
- перегляд журналу компіляції у випадку помилки компіляції;
- перегляд турнірної таблиці в ACM-стилі;
- у позатурнірний час вирішувати завдання зі «збірників»;
- обговорення завдань, збірників, турнірів і розділів на вбудованому форумі.
- Інтерфейс адміністратора турнірної системи дозволяє:
- створювати, блокувати та видаляти облікові записи учасників системи;
- вмикати режим самостійної реєстрації учасниками;
- створювати і видаляти завдання, турніри, збірники та розділи;
- встановлювати і переносити час турнірів;
- вносити HTML-тексти завдань і малюнки до них;
- вносити тестові пари (input.txt і pattern.txt) до завдань;
- вносити і компілювати на сервері тестувальну програму;
- переглядати список рішень учасників, імена відправлених файлів, вихідні коди кожного рішення;
- переглядати журнали компіляції та перевірки кожної спроби;
- заново відправляти спроби на повторну компіляцію;
- завантажувати запаковані zip-файли із завданнями (їх умовою, тестами та налаштуваннями) і вивантажувати їх;
- додавати в систему мови програмування, вказувати свої командні рядки компіляції;
- розподіляти перевірку рішень на кілька серверних комп'ютерів.

Ми розробили систему для перевірки розв'язання задач if(it), назвавши її на честь міфологічного персонажа Іфіта, завдяки якому, за однією з легенд, були започатковані (за іншою версією — відроджені) олімпійські ігри; а також на честь умовного оператора, який, либонь, використовують усі програмісти найчастіше.

Користувачем системи може стати будь-який користувач. Для цього йому необхідно пройти реєстрацію на сервері, для чого спершу слід вказати адресу своєї поштової скриньки і код підтвердження (CAPTCHA) (рис. 1).

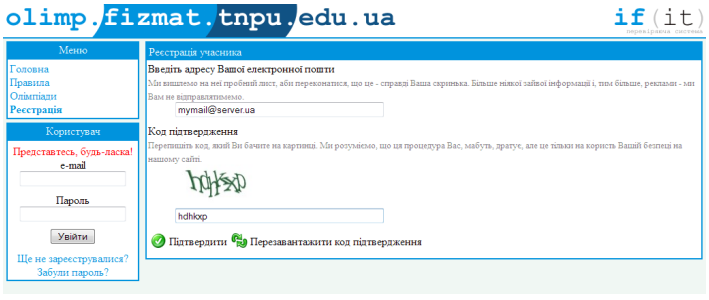

Рис. 1. Реєстрація учасників

Після цього на вказану адресу буде відправлено лист із посиланням, яке містить ключ підтвердження. Таким чином відбувається підтвердження і перевірка наявності скриньки, яку вказав користувач. Система не відправлятиме більше одного листа на одну скриньку протягом доби, таким чином не даючи можливість використовувати її для «закидання» листами поштових ящиків. Перейшовши за посиланням з поштової скриньки, користувач має змогу заповнити анкету даних (рис. 2).

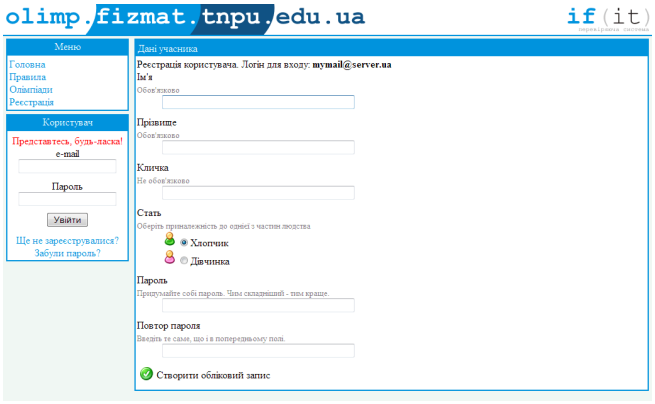

Рис. 2. Заповнення реєстраційної форми

Після підтвердження форми з особистими даними система створює обліковий запис користувача. Користувачі авторизуються за допомогою введення даних у поля, які відображаються на панелі користувача.

Користувачі мають можливість вносити виправлення у введенні дані та змінювати свій пароль. За втрати паролю користувач може відновити його за допомогою спеціальної форми, у якій він вказує електронну пошту і код підтвердження, після чого засвідчує бажання змінити пароль зі своєї поштової скриньки. Уся історія зміни особистих даних користувача зберігається у журналі, до якого мають доступ адміністратори. Для зберігання усі паролі хешуються алгоритмом md5.

Різні користувачі системи володіють різними привілеями та правами. У даній системі поділ прав базується за критеріями у групи. Кожен користувач належить до однієї з груп, що визначає доступні йому права у системі. Існують такі групи користувачів: «Непідтверджені», «Підтверджені», «Члени журі», «Адміністратори». На даний час систему сконфігуровано таким чином, що після реєстрації користувач стає членом групи «Непідтверджені». Члени цієї групи мають право на перегляд інформації, проте для того, аби брати участь у турнірах, адміністратор повинен провести їх ідентифікацію (перевести в групу «Підтверджених» користувачів). Ця дія має тимчасовий характер і буде ліквідована за покращення безпеки системи перевірки. Правом на зміну групи користувача володіє лише група адміністраторів.

Турніри можуть створювати і редагувати адміністратори та члени журі. Олімпіада має час початку та час закінчення. До початку інформація про неї доступна користувачам, але одержання списку її завдань не є доступним. Завдання відкриваються з настанням визначеного в олімпіаді часу початку. Час прийому розв'язків до задач необмежений, але при оцінювані результатів приймаються до уваги лише ті розв'язки, які поступили на сервер не пізніше ніж завершилась олімпіада.

Протягом олімпіади та після її завершення будь-хто може ознайомитися із рейтингом учасників та кількістю набраних ними балів.

Турнір може бути відкритим або закритим. У відкритому турнірі може зареєструватися будь-який підтверджений користувач. Для цього йому достатньо відкрити сторінку турніру і натиснути на кнопку «Брати участь» (рис. 3).

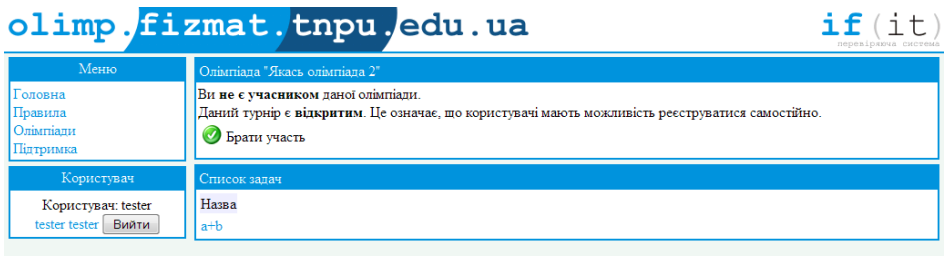

Рис. 3. Реєстрація участі у турнірі

Якщо ж турнір закритий, то учасників до нього можуть долучати лише адміністратори та члени журі (рис. 4).

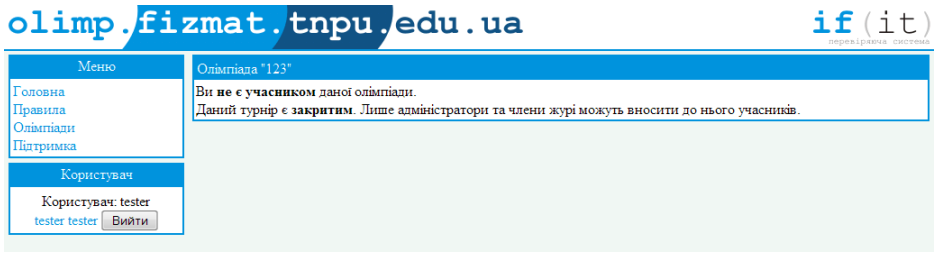

Рис. 4. Додавання учасника турніру

Кожна задача має свою унікальну назву та умову. Задача може перебувати в одному з трьох станів: «На редагуванні», «Закрита», «Відкрита». Задача на редагуванні може бути доступна лише адміністраторам чи членам журі. Закрита задача може бути доступна користувачеві лише у складі олімпіади, учасником якої він є. Відкрита задача доступна користувачам для вільного (тренувального) розв'язування.

В олімпіади можуть входити закриті та відкриті задачі. Залежність між задачами та олімпіадами визначається схемою «багато до багатьох», тобто одна задача може входити до кількох олімпіад.

Для кожної задачі адміністратори та члени журі можуть вносити (редагувати) тести, за допомогою яких здійснюватиметься перевірка задач. Кожен тест складається із вхідних даних, шаблону відповіді та кількості балів, яка буде нарахована користувачеві за правильну відповідь.

Практичний проект складається із двох складових: web-сервера і тест-сервера. Web-сервер реалізовує інтерфейс із користувачами (учасниками, членами журі, адміністраторами), роботу з базою даних, підрахунок балів, визначення рейтингу учасників. Тест-сервер займається компіляцією коду та запуском отриманого машинного коду на тестування. Такий поділ був зумовлений ризиками, пов'язаними із запуском програми користувача, яка є потенційно небезпечною для системи. Тому тест-сервер може бути встановлено на іншому спеціально відведеному комп'ютері. У випадку дестабілізації роботи тестувальної системи її можна перезавантажити без потреби призупиняти діалог з користувачами.

- Web-сервер виконує такі функції:
- спілкування з користувачем;
- ведення бази даних;
- обмін інформацією з тест-сервером.

Інтерфейс із користувачем побудований на власноруч розробленому php-двигуні, спроектованому на засадах AJAX (Asynchronius Javascript and XML). Php-сценарії з боку сервера та javascript-сценарії з боку клієнта забезпечують обмін інформацією між вікном браузера і web-сервером без перезавантаження всієї сторінки при запитах. Запит подається лише на ту частину інформації, яку потрібно оновити у вікні клієнта. Інформацію, яку передають клієнту, подають у вигляді XML-коду, парсингом якого займається javascript-сценарій на стороні клієнта. Такий спосіб взаємодії з клієнтами робить зручнішим інтерфейс, прискорює роботу сайту та, що найважливіше, зменшує навантаження на web-сервер.

Будуючи проект, на деякому етапі було вирішено відмовитись від підтримки браузера Internet Explorer версії 6 (і нижче) у зв'язку з невдалою підтримкою ним багатьох web-стан-

дартів. Ми вважаємо, що такий крок є виправданим, враховуючи постійне зменшення кількості користувачів даного оглядача і відмову від його підтримки таких сервісів як Google і Youtube.

Основними елементами інтерфейсу користувача є головне меню (розміщене ліворуч), панель користувача (ліворуч під головним меню) та область контенту, яка змінюється залежно від дій користувача (рис. 5).

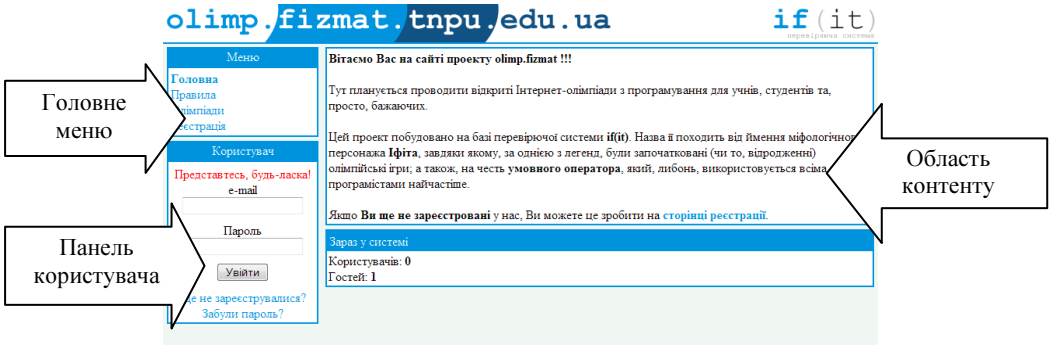

Рис. 5. Інтерфайс програми

Те, що проект складений за принципами AJAX, означає, що зазвичай при навігації по сайту завантажуватиметься із сервера лише вміст вікна контенту. Уся сторінка завантажується лише при першому вході на сайт та у випадку зміни даних користувача.

Уміст головного меню залежить від того, хто на даний час працює із системою, тобто для адміністратора, члена журі, учасника та неавторизованого користувача пункти головного меню будуть різними, залежно від їхніх прав у системі.

На панелі користувача завжди відображається ім'я користувача, який на даний момент перебуває в системі. Якщо користувач ще не авторизувався, то в даній області відображаються поля, в яких він може ввести свій логін та пароль. За логін ми вирішили використовувати адресу електронної пошти користувача, що вирішує проблему унікальності назви облікового запису та запам'ятання її учасником. Неавторизованому користувачеві доступна головна сторінка (зі статистикою кількості користувачів у системі на даний момент), список правил користування системою, список олімпіад, які відбуваються на даний момент, а також список змагань, які вже відбулися.

Web-сервер і тест-сервер обмінюються даними через мережу. Для повідомлення тест-серверу про те, що для нього є завдання, використовуються мережеві сокети. Тест-сервер при роботі прослуховує наданий йому мережевий порт, а web-сервер за необхідності відсилає на нього дані. Для зворотного зв'язку (від тест-сервера до web-сервера) використовуються http post-запити. Такий спосіб одержання даних є природним для web-сервера, а тому звільняє від необхідності реалізовувати прослуховування ним окремого порту. Програма-тестер написана на C++ для роботи під керуванням операційної системи Windows сімейства NT. Вона працює в багатопоточному режимі, що зумовлено необхідністю працювати із web-сервером і водночас запускати коди на компіляцію та перевіряти програми.

При реалізації багатозадачності виникла потреба працювати водночас із групами потоків, кожен з яких може виконувати однакове завдання. Для цього було створено бібліотеку Multithreader, яка включає кілька класів-шаблонів.

- CMT_Server — шаблон, що управляє групою однакових за призначенням потоків. До нього слід звертатися, щоб передати на виконання певне завдання. Завдання потрапляє у чергу. Коли у групі потоків з'являється вільний від завдань потік, то йому делегується перше завдання з черги;
- CMT_Thread — шаблон, який описує потік. На основі цього шаблону описано класи, які виконують функціональні завдання (в нас це, наприклад, web-клієнт, компілятор, тестер тощо). Основною в цьому шаблоні є віртуальна функція dotask(TTask* pTask), якій CMT_Server передасть опис завдання, яке потрібно виконати;
- CMT_Task — клас завдання, яке ставиться в чергу і буде передане одному з групи потоків.

Перерахуємо види потоків, які виконує тест-сервер:

- Listener. Цей потік увесь час перебуває в режимі «прослуховування» наданого йому порту мережі. На цей порт web-сервер відправляє команди керування та повідомлення про необхідність взяти на перевірку нові задачі;
- Web-client. Він виконує зворотний зв'язок із web-сервером, знімає із сервера нові завдання, тести для задач та повідомляє web-серверу про результати компіляцій та перевірок;
- Compiler. Цей потік запускає отримані коди на компіляцію, тобто запускається компілятор і йому передається код надісланої учасником програми, а також очікується відповідь компілятора, яка аналізується;
- Examiner. Цей потік виконує запуск скомпільованих програм. Імітуючи стандартний потік уводу, програмі передаються вхідні дані; перехвачується стандартний потік виводу; фіксується час виконання програми. По завершені роботи програми аналізується її вивід у стандартний потік шляхом перевірки на ідентичність шаблону. Алгоритм перевірки не критичний до типів розриву рядка («\r», «\r\n», «\n»), а також не зважає на останні (залишкові) розриви рядків.

Висновки. Отже, у статті розглянуто історію проведення шкільних олімпіад з інформатики та принципи, які використовують для добору чи складання олімпіадних завдань. Завдання, які пропонують на олімпіадах з програмування, потребують від учасників відповідної бази знань, яка виходить далеко за межі навчальних програм.

Для організації олімпіад розроблено систему if(it), яка завдяки web-інтерфейсу може слугувати для проведення як внутрішніх (локальних) олімпіад, так і зовнішніх (Інтернет-) олімпіад. Її можливості дозволяють проводити турніри одночасно, розв'язувати учасникам тренувальні задачі в позаолімпіадний час. Кожна категорія користувачів (адміністратори, члени журі та учасники) мають доступ до своїх можливостей через web-інтерфейс, що дозволяє їм працювати віддалено. На даний час програма встановлена на сервер фізико-математичного факультету Тернопільського національного педагогічного університету імені Володимира Гнатюка (http://olimp.fizmat.tnpu.edu.ua), де проходить процес тестування й апробації.

Перспективи подальших досліджень. Перспективними напрямами роботи щодо вдосконалення даного програмного продукту можуть бути: покращення захисту (захищеності системи від виконуваних програм учасників), уведення засобів інтерактивного спілкування учасників між собою, в тому числі засобів обговорення тренувальних завдань, покращення гнучкості налаштувань тестувальної програми, подальша апробація для реальних турнірів (міських та обласних олімпіад з програмування).

ЛІТЕРАТУРА

1. Безпека дитини у Всесвітньому павутинні [Електронний ресурс]. — Режим доступу: http://www.seotm.com/news/Internet/Bezpeka_ditini_u_Vsesv%D1%96tn__omu_pavutinn%D1%96.html
2. http://wikipedia.org
3. Кривонос О. М. Учнівські олімпіади з інформатики (сучасний етап) / О. М. Кривонос // Вісник Житомирського державного університету імені Івана Франка. — 2008. — №41. — С. 85–88.
4. Маланюк П. М. Олімпіадні задачі з програмування / Маланюк П. М. — Тернопіль, 2000. — 144 с.
5. http://olympiads.ru
6. http://uoi.in.ua
7. http://www.contester.ru
8. http://www.ejudge.ru
9. http://www.topcoder.com