

УДК 37.018.43:004.43

Умрик Марія Анатоліївна

кандидат педагогічних наук, доцент кафедри теоретичних основ інформатики
Інститут інформатики НПУ імені М. П. Драгоманова, м. Київ, Україна
ymallia@gmail.com

Біляй Юрій Петрович

старший викладач кафедри теоретичних основ інформатики
Інститут інформатики НПУ імені М. П. Драгоманова, м. Київ, Україна
azb@ukr.net

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ДИСТАНЦІЙНОГО НАВЧАННЯ В ПРОЦЕСІ НАВЧАННЯ СУЧАСНИХ МОВ ПРОГРАМУВАННЯ

Анотація. У даному дослідженні розглянуто використання дистанційних технологій навчання в процесі організації науково-дослідного завдання із сучасних мов програмування. У статті наведено приклад навчального проекту із сучасних мов програмування. Аргументовано необхідність використання сучасних інформаційно-комунікаційних технологій XXI століття, зокрема дистанційних, у процесі вивчення сучасних мов програмування. Це дозволить сформувати студентами навичок, які вкрай необхідні в процесі програмування (це навички самоорганізації і самодисципліни, комунікації, навички роботи в команді тощо). Виокремлено структурні одиниці навчального проекту відповідно до використання технологій дистанційного навчання. Описано загальні характеристики і методикою використання відповідних сучасних інформаційно-комунікаційних технологій.

Ключові слова: сучасні мови програмування; дистанційні технології навчання; науково-дослідне завдання; навички XXI століття.

1. ВСТУП

Постановка проблеми. Актуальність даного дослідження зумовлена важливістю впровадження сучасних інформаційно-комунікаційних технологій як пріоритетного напрямку розбудови освітньої системи України (Закон України «Про основні засади розвитку інформаційного суспільства в Україні на 2007–2015 роки», Національна стратегія розвитку освіти в Україні на 2012–2021 роки). Тема даного дослідження є актуальною через протиріччя, яке виникло між: широким використанням дистанційних технологій навчання і недостатньою розробленістю окремих компонентів методичної системи навчання програмування з використанням сучасних інформаційно-комунікаційних технологій. На часі залишається проблема формування студентами — майбутніми вчителями інформатики, навичок XXI століття, які вкрай необхідні в процесі навчання програмування. Це такі навички як самоорганізація і самодисципліна, навички комунікації, навички роботи в команді і співпраця, відповідальність і адаптація, а також формування нових навичок і методів. Часто навчання програмування базується на виконанні кожним студентом власної окремої індивідуальної задачі, самостійна її розробка і захист розробленого додатку перед викладачем. У реальній життєвій ситуації великі проекти, в основному, розробляє команда програмістів, з конкретно визначеними ролями. Отже, існує також протиріччя між реальним життєвим циклом розробки програмного забезпечення і методикою навчання програмування вищими навчальними закладами.

Аналіз останніх досліджень і публікацій. Питання щодо можливості використання технологій дистанційного навчання висвітлені в низці праць українських і зарубіжних учених. Серед них — А. А. Андрєєв, В. Ю. Биков, Р. С. Гуревич,

В. В. Ільїн, А. П. Кудін, О. Г. Кузьмінська, В. М. Кухаренко, Н. В. Морзе [9], Є. С. Полат, Є. М. Смирнова-Трибульська, J. Bartram, T. Bates, B. Lockee, S. Catherine, F. Willits, M. Cornelia, S. Feldman, G. Randy, G. SARAH, N. Hara, R. Jones та ін.

Українські і зарубіжні вчені, роботи яких присвячені сучасному програмуванню, методиці навчання програмування, сучасним методам технологій програмування, — Брукс Ф., Буч Г., Гришко Л. В. [5], Дейкстра Е. [2], Крамар Ю. М. [6], Кнут Д. [1], Нечай О. С., Пратт Т. [3], Сайдаметова З. С. [4], Страуструп Б., Йордан Є. та ін.

Мета статті. З огляду на це, **метою** статті є висвітлення підходів використання дистанційних технологій навчання в процесі навчання сучасних мов програмування для формування студентами окремих навичок XXI століття, а саме навичок самоорганізації і самодисципліни, комунікації, роботи в команді, співпраці, відповідальності й адаптивності, а також формування нових навичок і методів.

2. МЕТОДИ ДОСЛІДЖЕННЯ

Дослідження проводилось у рамках НДР „Розробка комп’ютерно-орієнтованих методичних систем навчання природничих дисциплін” Національного педагогічного університету імені М. П. Драгоманова, державний реєстраційний номер 0111U000526. Під час дослідження були використані такі методи: аналіз теоретичних джерел, присвячених проблемам розробки інноваційних методик навчання програмування, вивчення й узагальнення передового досвіду використання дистанційних технологій навчання для формування студентами навичок XXI століття, зокрема робота в команді і співпраця.

3. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

3.1. Постановка завдання

Актуальним стає формування відповідних навичок XXI століття в галузі програмування для бакалаврів (майбутніх учителів інформатики), адже існуючі методичні системи навчання програмування не відповідають сучасним потребам суспільства.

Зокрема, розроблено курс „Сучасні мови програмування”, призначений для підготовки фахівців у педагогічних вищих навчальних закладах, які навчаються за напрямом підготовки «Інформатика*». Цей курс належить до варіативної частини циклу професійної і практичної підготовки.

Метою вивчення дисципліни “Сучасні мови програмування” є поглиблення знань студентів стосовно основних концепцій сучасних мов програмування, формування у них систематичних компетентностей з теорії і практики програмування на мовах високого рівня, формування цілісного уявлення про загальні властивості мов програмування високого рівня.

Під час вивчення даного курсу студентам пропонується виконати науково-дослідне завдання. Виконання науково-дослідного завдання відбувається під час роботи кожного студента над спільним проектом у команді, так моделюється реальний життєвий цикл розробки програмного забезпечення. Студенти отримують одне спільне завдання-проект, поділений на частини-підзадачі, кожну з яких виконує два студенти. Останнє завдання отримує студент — лідируючий програміст, який має об’єднати усі підпрограми в один модуль проекту. За потреби, враховуючи навчальний рівень групи, викладач може допомагати лідируючому програмісту, або, якщо це необхідно, роль лідируючого програміста може відігравати викладач.

Приклад завдання проекту.

Створити словник-перекладач.

Вимоги до виконання науково-дослідного завдання подані нижче.

1. Словом у словнику називатимемо будь-яку кількість символів без пропусків.
2. Результатом виконання індивідуального завдання є модуль, принцип роботи якого описаний у вимогах до підпрограм.
3. Модуль має містити одну булеву функцію з розділу (public) з назвою завдання, яке буде повертати значення TRUE у разі успішного форматування і збереження вихідного файлу.
4. Параметри, які передаються у функцію, вказані у вимогах до підпрограм.

✓ **Delphi**

```
functionTaskName(varresult_string: String): boolean;
```

✓ **C++**

```
boolTaskName(char* result_string);
```

5. Функції для роботи зі словником описані в 9 однойменних модулях-підпрограмах у порядку їх використання:

Для ускладнення завдання лідируючому програмісту можна не повідомляти порядок використання підпрограм для правильної функціональності результуючого проекту.

- DelExtra(‘шлях до вхідного файлу TaskName.txt’);
- OneGap(‘шлях до вхідного файлу TaskName.txt’);
- PureWords(‘шлях до вхідного файлу TaskName.txt’);
- SingleWord(‘шлях до вхідного файлу TaskName.txt’);
- Sort(‘шлях до вхідного файлу TaskName.txt’);
- DelSame(‘шлях до вхідного файлу TaskName.txt’);
- DelKnown(‘шлях до вхідного файлу TaskName.txt’, ‘шлях до словника Dictionary.txt’);
- Translate(‘шлях до вхідного файлу TaskName.txt’);
- AddGlossary(‘шлях до вхідного файлу TaskName.txt’, ‘шлях до словника NewDictionary.txt’);

6. Вимоги до підпрограм:

a *DelExtra*

Відформатувати файл так, щоб він містив лише букви кирилиці і спеціальні знаки — апостроф, тире та знак пробілу.

Вхідні дані: текстовий файл TaskName.txt, у якому містяться різні символи таблиці ASCII.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

b *OneGap*

Відформатувати файл так, щоб він починався зі слова. Між словами у тексті повинен бути лише один пропуск (знак пробілу).

Вхідні дані: текстовий файл TaskName.txt, у якому містяться лише букви кирилиці і спеціальні знаки — апостроф, тире та знак пробілу.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

c *PureWords*

У тексті файлу видалити ті спеціальні знаки, які знаходяться перед першою і після останньої букв слова.

Вхідні дані: текстовий файл TaskName.txt починається зі слова. Між словами у тексті лише один пропуск (знак пробілу).

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

d *SingleWord*

Відформатувати файл так, щоб кожен рядок файлу містив лише одне слово.

Вхідні дані: текстовий файл TaskName.txt починається зі слова. Між словами у тексті лише один пропуск (знак пробілу).

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

e *Sort*

Відсортувати рядки файлу за алфавітом.

Вхідні дані: текстовий файл TaskName.txt, кожен рядок якого містить лише одне слово.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

f *DelSame*

Видалити з файлу слова, що повторюються.

Вхідні дані: текстовий файл TaskName.txt, кожен рядок якого містить лише одне слово, упорядковане за алфавітом.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

g *DelKnown*

Із заданого файлу видалити слова, що містяться у файлі-словнику.

Вхідні дані: текстовий файл TaskName.txt, кожен рядок якого містить лише одне слово, упорядковане за алфавітом без повторів, і файл Dictionary.txt, який у кожному рядку містить слово і слово-переклад, розділені пропуском.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

h *Translate*

До кожного слова у файлі додати через пропуск слово-переклад (через запит до користувача).

Вхідні дані: текстовий файл TaskName.txt, кожен рядок якого містить лише одне слово, упорядковане за алфавітом.

Вихідні дані: текстовий файл TaskName.txt містить дані відповідно до завдання.

i *AddGlossary*

Додати із заданого файлу усі слова з перекладом до файлу словника, не порушуючи упорядкованість.

Вхідні дані: текстові файли TaskName.txt, Dictionary.txt, кожен рядок яких містить слово і слово-переклад, розділені пропуском, упорядковані за алфавітом.

Вихідні дані: текстовий файл NewDictionary.txt містить дані відповідно до завдання.

7. Завдання для лідируючого програміста.

Описати програму для роботи зі словником.

Вказівка.

Форма програми має містити поле для введення тексту, який записується у файл TaskName.txt і поле для виведення вмісту файлу NewDictionary.txt.

3.2. Обов'язки і права членів команди під час роботи над спільним науково-дослідним завданням

Робота в команді, зокрема під час програмування для виконання науково-дослідного завдання, передбачає як самостійність мислення всіх учасників команди, так і включення кожного студента до спільної роботи для розв'язання поставленого завдання. Студенти мають розуміти, що об'єднання окремих спеціалістів у групу-команди робить можливим реалізацію проектів, які не під силу одній людині. З іншого боку, важливим є розуміння того, що команда, як єдине ціле, несе відповідальність за все, що вона робить і не робить. Важливими стають не тільки професійні якості кожного учасника групи, компетентності у певній галузі, а також психологічні, комунікаційні аспекти поведінки кожної особи у групі. Саме робота у команді для виконання спільного проекту із сучасних мов програмування стимулює формування студентами відповідних навичок ХХІ століття, а саме навичок самоорганізації і самодисципліни, комунікації, роботи в команді, співпраці, відповідальності і адаптивності, а також формування нових навичок і методів.

Було виокремлено особливості і ключові фактори поведінки членів команди під час роботи над спільним проектом.

Обов'язки членів команди [8] подані нижче.

1) Кожен учасник команди зобов'язаний направити максимум зусиль на досягнення поставлених перед командою цілей. Наразі виключається егоїстична поведінка: приховування ідей і рішень, необхідних команді, перешкоджання роботі інших задля власної вигоди, перешкоджання співробітництва з причин власних симпатій і антипатій тощо.

2) Ніхто з учасників команди не повинен вважати свою думку вищою думкою і рекомендацій більшості, навіть якщо це рішення суперечить його власним ідеям.

3) В команді встановлюється рівноправ'я всіх учасників, та ділові стосунки між ними. Кожен член команди повинен бути тактовним по відношенню до колег.

4) Забороняється інформувати про зміст дискусій і рішень команди сторонніх осіб або використовувати ці рішення в особистих інтересах.

5) Терпимість і взаємна повага є найважливішими умовами успішної роботи в команді.

6) Кожен член команди зобов'язаний інформувати інших про все, що їм необхідно знати для виконання поставлених завдань.

Права членів команди [8] подані далі.

1) Вільно висловлювати свою думку і перевіряти її.

2) Вимагати встановлення конфіденційних, довірливих відносин. Думки, висловлені під час роботи в команді, заборонено повідомляти третім особам. Висловлені ідеї не можуть стати причиною для небезпеки учасника команди.

3) Отримувати об'єктивні і повні відомості від інших членів команди.

4) Вимагати, щоб висловлені думки використовувалися іншими членами команди поза роботою в команді тільки в тому випадку, якщо це не завдає шкоди виконанню поставлених перед командою завдань або якщо на це було отримано спеціальний дозвіл.

5) Вимагати, щоб у процесі роботи в команді дотримувалися принципи коректної поведінки між усіма учасниками.

3.3. Структурні елементи навчального проекту із сучасних мов програмування відповідно до використання технологій дистанційного навчання

Науково-дослідне завдання виконувалось у рамках відведених годин на позааудиторну самостійну роботу дисципліни «Сучасні мови програмування». Через це вкрай актуальним стало використання саме дистанційних технологій навчання.

Нижче описані структурні елементи навчального проекту відповідно до використання технологій дистанційного навчання.

3.3.1. Дистанційний курс для підтримки навчання сучасних мов програмування

Дистанційний курс для підтримки навчального процесу сучасних мов програмування розроблено і розміщено на навчальному порталі дистанційних курсів Інституту інформатики НПУ імені М. П. Драгоманова за адресою <http://www.moodle.ii.npu.edu.ua/course/view.php?id=136>.

Дистанційний курс може бути використаний як засіб навчання для студентів денної, заочної, дистанційної форм навчання на всіх етапах навчальної діяльності студентів під час вивчення сучасних мов програмування.

У дослідженні дистанційний курс був використаний як додатковий засіб підтримки навчального процесу для студентів денної форми навчання.

У розробці дистанційного курсу за основу було взято структуру [9] для електронно-навчальних курсів.

Дистанційний курс складається з окремих модулів, які поділені на навчальні одиниці — теми. Кожен структурний модуль містить:

- методичні рекомендації щодо вивчення модуля;
- теоретичний навчальний матеріал;
- лабораторні роботи;
- завдання для самостійної роботи;
- завдання для самоперевірки;
- модульний контроль;
- додаткові матеріали.

Окремо визначено модуль для проведення навчально-дослідного завдання.

У рамках тематики статті визначимо особливості використання дистанційного курсу під час проведення науково-дослідного завдання із сучасних мов програмування.

Структурний модуль навчально-дослідного завдання містить:

- завдання проекту «Словник-перекладач»;
- етапи виконання проекту з розподіленням завдань для кожного учасника проекту;
- список завдань, зашифрованих у QR кодах (опис і методика використання кодів показана нижче);
- права й обов'язки членів команди;
- календар важливих дат моніторингу виконання проекту, а також кінцеві дати складання і захисту проекту;
- ресурс чат у форматі IRC (опис і методика використання чату показана нижче);
- літературні й Інтернет джерела, необхідні для виконання проекту;
- список посилань на всі необхідні для виконання проекту онлайн ресурси в Інтернеті.

3.3.2. Список завдань, зашифрованих у QR кодах





QR (QuickResponse) — перекладається як «швидка відповідь». Коди є мініатюрними носіями даних, де зберігаються текстові повідомлення обсягом приблизно на половину сторінки формату А4. Популярності цей варіант кодування даних набув у Японії, через необхідність в кодуванні текстових і числових даних за допомогою невеликих картинок. QR-коди призначені для зберігання великого обсягу даних за невеликої площі їх розміщення [14].

У дистанційному курсі зручним є використання таких форматів QR-кодів: інтернет-адреса, контактні дані, практичні завдання, адреса електронної пошти, географічні дані, QR-коди рекламного характеру. Детальніше про QR-коди в дистанційному навчанні можна подивитися в [9].

Під час виконання науково-дослідного завдання із сучасних мов програмування кожен студент отримує своє завдання в рамках виконання спільного проекту у команді. Номер завдання обирається випадково зі списку завдань, зашифрованих в QR-кодах. Перелік усіх завдань знаходиться в дистанційному курсі дисципліни.

Таблиця 1

Приклади варіантів індивідуальних завдань

№ варіанта	Зашифрований QR код	№ варіанта	Зашифрований QR код
1		2	
3		4	

Для декодування QR-кодів студенти використовують або сервіси зчитування QR-кодів, перелік яких наведено в дистанційному курсі, або власні мобільні пристрої (смартфони, планшети тощо) зі встановленим спеціальним вільно поширюваним програмним забезпеченням (вказівки до роботи з цими програмами наведені в дистанційному курсі).

3.3.3. Чат у форматі IRC

IRC (англ. Internet Relay Chat) — сервіс Інтернету, використання якого надає користувачам можливість спілкування шляхом надсилання текстових повідомлень багатьом людям з усього світу одночасно (у режимі реального часу). Аналогію можна знайти у радіо на коротких хвилях, навіть використовується аналогічне поняття англ. channel (канал, хвиля) [11].

Відмінність даного формату від класичного полягає в тому, що користувач може налагодити роботу свого власного «каналу», тобто налаштувати список тих людей, які можуть потрапити до цього «каналу» і спілкуватися в ньому.

Даний тип чату корисний буде для налагодження роботи і спілкування в проекті із сучасних мов програмування. Нижче подано особливості проведення спілкування в чаті формату IRC.

- 1) Перший, найнижчий рівень спілкування в проекті — це парне спілкування, тобто необхідне налагодження парного каналу. Парне спілкування використовується для спілкування двох студентів під час реалізації парного дистанційного програмування, адже, як було вже сказано, кожен підзадачу проекту виконує два студенти. Поняття і методику проведення парного програмування наведено нижче.
- 2) Другий рівень спілкування — груповий. Для нього потрібне налагодження ширшого, ніж парне спілкування, каналу. На цьому рівні створюється комунікація для групи студентів, де вони можуть обговорювати питання, які виникли щодо взаємодії і роботи окремих підзадач.
- 3) Третій, останній рівень спілкування — спільний для усіх учасників проекту. Цей рівень видимий для всіх учасників і використовується для обговорення роботи створеного в цілому проекту.

Зазначимо, що для викладача усі рівні спілкування в чаті є видимі і прозорі. За необхідності викладач може корегувати дії студентів як окремої пари, так і групи загалом.

3.3.4. Парне дистанційне програмування

Парне програмування — техніка програмування, за якої код створюється парами людей, які програмують одну задачу, сидячи за одним робочим місцем. Один програміст («ведучий») займається безпосередньо програмуванням. А другий програміст («штурман» [1]) зосереджений на задачі в цілому. Час від часу ролі програмістів можуть змінюватися, зазвичай через півгодини.

Дистанційне (віддалене) парне програмування, також відоме як віртуальне парне програмування, або розподілене парне програмування — це різновид парного програмування, коли два або більше програмістів, знаходяться на відстані один від одного [2]. Вони працюють через засоби спільного редагування коду, віддалений робочий стіл, або через плагіни для інтегрованої системи розробки.

У науково-дослідному завданні два студенти отримують одну підзадачу спільного для всіх проекту. Програмування даної підзадачі запропоновано виконувати, використовуючи методику дистанційного парного програмування.

Для цього використовують програмні засоби «Віддалений робочий стіл», використовуючи який кожен член пари, знаходячись на відстані один від одного, мають змогу бачити робочий стіл іншого, і навіть коригувати і вносити правки в дії свого напарника.

Зазвичай, засоби для дистанційного парного програмування використовують лише в тому разі, коли студенти під час виконання завдання знаходяться на відстані один від одного.

У цьому проекті студентам було запропоновано використання програмного засобу «Підключення до віддаленого робочого столу» стандартних програм операційної системи Windows і вільно поширюваний програмний засіб TeamViewer [Ошибка! Неизвестный аргумент ключа.].

3.3.5. Репозиторій загального доступу

Усі індивідуальні завдання (підзадачі) відвантажують до репозиторію загального доступу через сервіс github.com.

GitHub — найбільший веб-сервіс за кількістю завантажених файлів для хостингу ІТ-проектів і їх спільної розробки [13]. Заснований на системі контролю версій Git і розроблений на Ruby on Rails і Erlang компанією GitHub. На рис.1 показано співвідношення файлів на різних мовах у різних подібних системах.

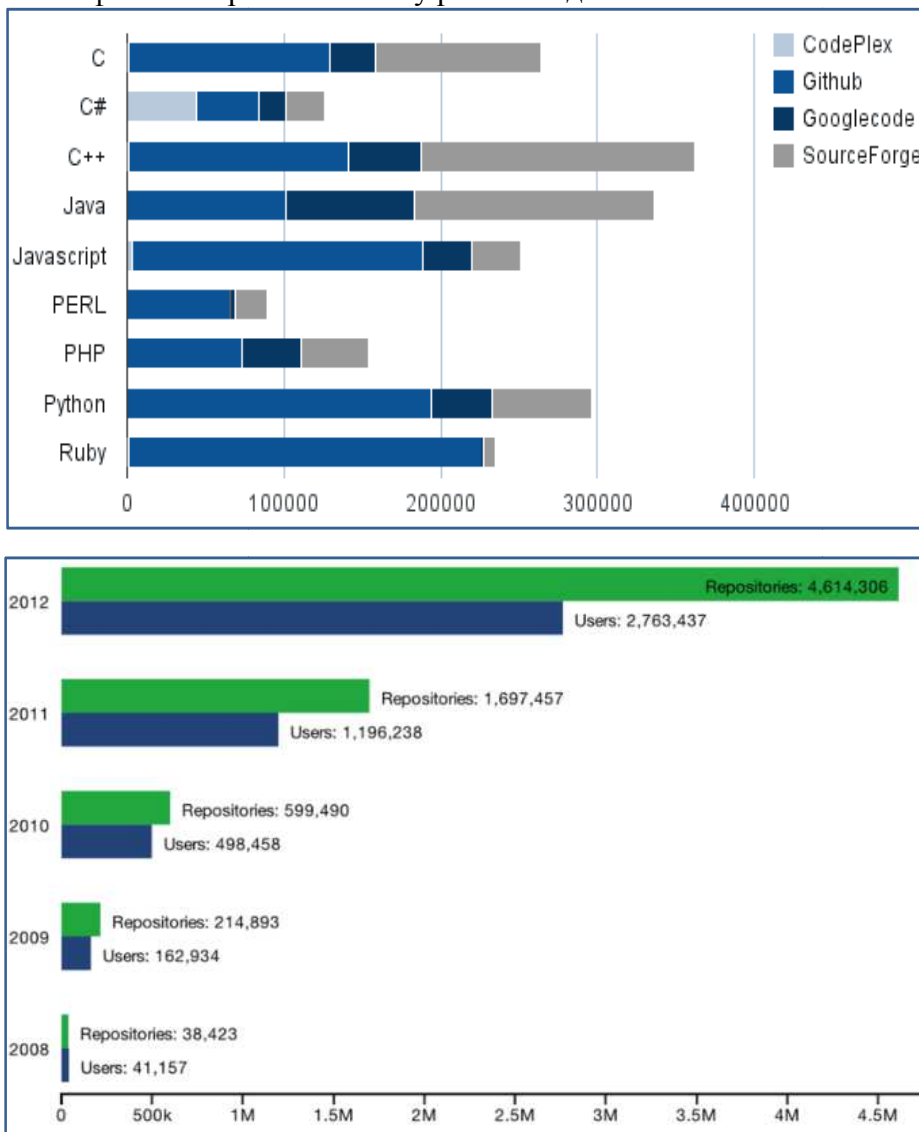


Рис. 1. Співвідношення файлів за мовами програмування на ресурсі Github (2012 рік, статистичні дані сайту <http://www.github.com>)

Сервіс абсолютно безкоштовний для проектів із відкритим вихідним кодом, а для комерційних проектів запропоновано різні платні тарифні плани.

Сайт має елементи соціальної мережі. Розміщений код розробники можуть коментувати, вносити правки, слідкувати за змінами у проектах інших розробників. Також у користувачів є можливість об'єднувати різні репозиторії, у тому числі і такі, що мають різних власників.

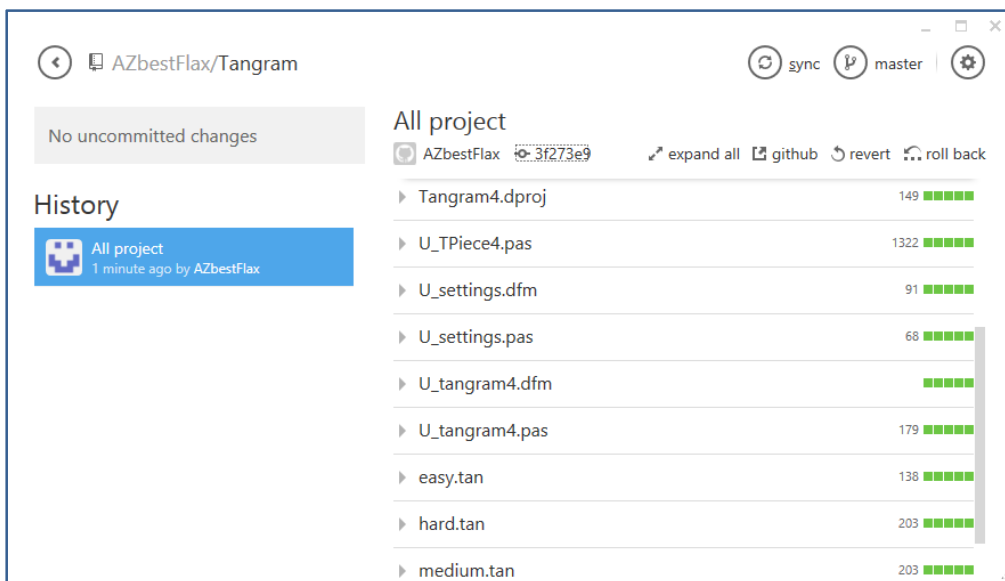


Рис. 2. Синхронізація файлів проекту (2014 рік, збережений екран сторінки проекту)

Для проектів створена персональна сторінка, Вікі і система відслідковування помилок. Безпосередньо на сайті можна переглядати файли проектів, реалізовано підсвічування синтаксису для більшості мов програмування. На платних тарифних планах можна працювати із сервісом через SSL і показувати репозиторій тільки визначеним користувачам. Сам код проектів можна отримати через Git (синхронізацію з локальним комп'ютером), а також завантажити звичайний архівний файл.

Починаючи з 5 грудня 2012 року, на сервісі додано можливість прямого додавання нових файлів у свій репозиторій через веб-інтерфейс сервісу.

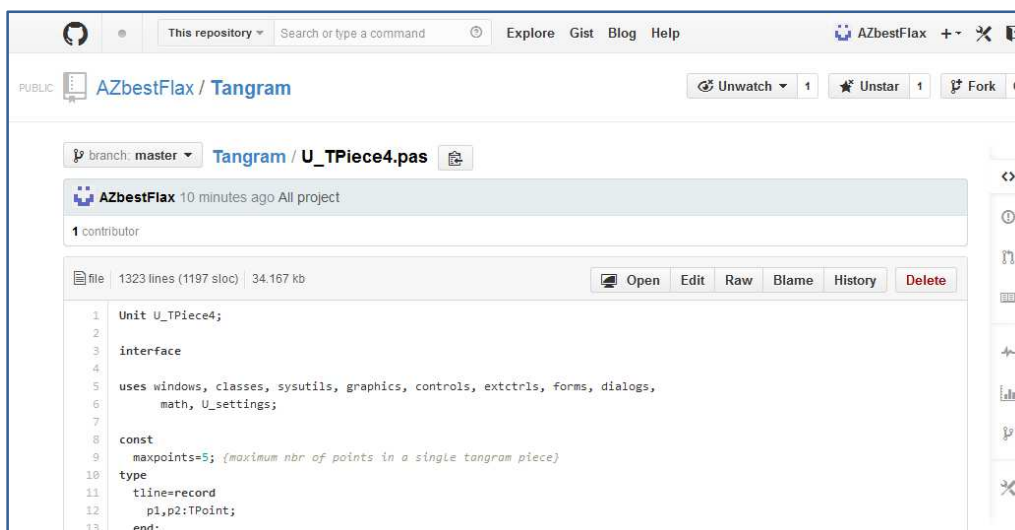


Рис. 3. Вихідний код проекту (2014 рік, збережений екран сторінки проекту)

На рис. 2 зображено додаток під операційну систему Windows для синхронізації файлів проекту.

На рис. 3 показано, який вигляд має файл і вихідним кодом через веб-інтерфейс.

З рис. 4 видно, як відображуються зміни, внесені розробниками, і доповнені коментарями.

Файли проекту можуть редагувати і коментувати як самі студенти, так і керівник проекту — викладач. Усі зміни фіксуються і можуть бути скасованими власником проекту. Отже, можна контролювати, коли, ким і які зміни були внесені до проекту. Використовуючи даний сервіс, можна задавати і контролювати терміни виконання проекту, так звані “дедлайни” (англ. dead-line — останній термін здачі).



Рис. 4. Відображення внесення змін у проект (2014 рік, збережений екран сторінки проекту)

3.3.6. Дистанційна тестуюча програма

Кожне індивідуальне завдання (власну підзадачу) студенти можуть перевірити на правильність виконання в системі онлайн тестування (online-judge).

Однією із систем онлайн перевірки є — ejudge, яка встановлюється на сервері. У цій системі є функція прийому й опрацювання відправлених розв’язків за протоколом http. Гнучка система призначення оцінок, вибудовування рейтингу учнів, вбудована можливість листування із судьями та інше, інше. Випробувати функціональність ejudge можна тут: <http://ejudge.ru/ege>.

Встановити дану систему можна на будь-яку Linux-подібну операційну систему, але для користувачів ALT Linux існують RPM-пакекти, для версій ALT 4.0 і 4.1.

Для використання системи онлайн перевірки завдань створені приклади вхідних даних, у яких передбачені типові помилки, що допускають студенти під час реалізації вказаних завдань.

Для прикладу розглянемо завдання, у якому пропонується видалити зайві пропуски між словами (для наочності позначимо пропуски знаками “*”). Типовими помилками є: перший пропуск перед першим словом, або після останнього; пропуски до або після розділових знаків; видалення потрібних пропусків та ін.

Приклади вхідних тестів:

Sample#1

```

Lorem*ipsum*dolor*sit*amet,*consectetur*adipiscing*elit,*sed*do*ei*usm
od*tempor*incididunt*ut*labore*et*dolore*magna*aliqua

```

Sample#2

*****Lorem*ipsum*dolor*sit*amet,*consectetur*adipiscing*elit,*sed*do*ei*usmod*tempor*incididunt*ut*labore*et*dolore*magna*aliqua***.****

Sample#3

Loremipsum**dolor**sit**amet***,***consectetur***adipiscing***elit**,***sed*do*ei*usmod*tempor*incididunt*ut*labore*et*dolore*magna*aliqua.*Ut*enim*ad*minim***veniam***,***quis*nostrud*exercitation*ullamco*laboris*ni*si*ut*aliquip*ex*ea*commodo*consequat**,*,*,*,***.**Duis*aute*irure*dolor*i*n*reprehenderit*in*voluptate*velit*esse*cillum*dolore*eu*fugiat*nulla*paria*tur.*Excepteur*sint*occaecat**cupidatat*non*proident,*sunt*in*culpa*qui*of*fi*cia*deserunt*mollit*anim*id*est*laborum***.**

Кожна відправлена спроба фіксується й перевіряється системою. Отже, можна слідкувати за процесом виконання завдання і коректністю його виконання.

4. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

1. У статті продемонстровані деякі підходи до використання дистанційних технологій у процесі навчання сучасних мов програмування для формування студентами окремих навичок ХХІ століття, які можуть бути використані й у вивченні інших предметів з програмування.

2. У результаті дослідження встановлено, що використання даних методик дозволяє покращити вміння студентів працювати в команді, зростає відповідальність студентів, розвиваються навички співпраці, оскільки постійний контроль над процесом створення проекту переходить від викладача на інших студентів, що сприяє кращій мотивації.

3. Використання спільних ресурсів і парного програмування для реалізації проекту сприяє соціалізації учасників проекту, розвитку самоорганізації і самодисципліни, а також більш швидкому обміну досвідом і знаннями між окремими студентами, порівняно з традиційним одноосібним програмуванням.

4. У подальших дослідженнях планується розробити більш масштабні проекти для розробки в межах не лише однієї групи (поток), а й проекти, розраховані на різні інститути й університети зі схожими спеціальностями.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Donald E Knuth, The Art of Computer Programming, Volumes 1-4, Addison-Wesley Professional 1997, 650 p.
2. Discipline of Programming [Edsger W. Dijkstra] Prentice-Hall Series in Automatic Computation, 1976, 217 p.
3. Pratt, Terrence W., Programming Languages: Design and Implementation, PrenticeHall, Inc., Englewood Cliffs, New Jersey, 1975.
4. Сейдаметова З. С. Подготовка инженеров-программистов по специальности «Информатика» / З. С. Сейдаметова — Симферополь : Крымское учебно-педагогическое государственное издательство, 2007 — 480 с.
5. Гришко Л. В. Методична система навчання основ програмування майбутніх інженерів-програмістів : автореф. дис. на здобуття вчен. ступеня канд. пед. наук : спец. 13.00.02 "Теорія і методика навчання (інформатика)" / Л. В. Гришко. — К., 2009. — 20 с.
6. Крамар Ю. М. Методи та інформаційно-програмні засоби синтезу та аналізу стилів програмування : автореф. дис. на здобуття вчен. ступеня канд. техн. наук : спец. 05.13.06 / Крамар Ю. М. — К., 2005. — 19 с.
7. Нечай О. С. Метод та засіб моніторингу дефектів проектування об'єктно-орієнтованого програмного забезпечення : автореф. дис. на здобуття вчен. ступеня канд. техн. наук: спец. 01.05.03 / Нечай О. С. — К., 2010. — 20 с.

8. Работа в команде — ключевые факторы успеха [Электронный ресурс] Центр дистанционного образования. — Режим доступа : http://www.elitarium.ru/2010/03/01/rabota_v_komande.html. — Заголовок з екрана.
9. Морзе Н. Оцінювання якості електронних навчальних ресурсів [Текст] / Н. Морзе, О. Глазунова // Наукові записки Тернопільського національного педагогічного університету імені Володимира Гнатюка. Сер. Педагогіка / гол. ред. Г. Терещук ; редкол. : В. Кравець, В. Мадзігон, М. Фіцула [та ін.]. — Тернопіль : ТНПУ, 2008. — № 7. - С. 3–19.
10. Умрик М. А. Організація дистанційної освіти в навчальному закладі / Умрик М. А., Нагороднюк В. В. // Науковий часопис Нац. пед. ун-ту імені М.П. Драгоманова. Серія №2. Комп'ютерно-орієнтовані системи навчання : зб. наук. праць / Редрада. — К. : НПУ імені М. П. Драгоманова, 2012. — № 13 (20). — С. 192–196.
11. IRC [Електронний ресурс] / Вікіпедія. — Режим доступу: <http://uk.wikipedia.org/wiki/IRC>. — Заголовок з екрана.
12. TeamViewer [Електронний ресурс] / Офіційна сторінка проекту. — Режим доступу : <http://www.teamviewer.com/ru/index.aspx>. — Заголовок з екрану.
13. GitHub [Електронний ресурс] / Вікіпедія. — Режим доступу : <http://uk.wikipedia.org/wiki/GitHub>. — Заголовок з екрана.
14. QR [Електронний ресурс] / Вікіпедія. — Режим доступу : <http://uk.wikipedia.org/wiki/QR-%D0%BA%D0%BE%D0%B4>. — Заголовок з екрана.

Матеріал надійшов до редакції 18.05.2014 р.

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ ДИСТАНЦИОННОГО ОБУЧЕНИЯ В ПРОЦЕССЕ ИЗУЧЕНИЯ СОВРЕМЕННЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Умрик Мария Анатольевна

кандидат педагогических наук, доцент кафедры теоретических основ информатики
Институт информатики НПУ имени М. П. Драгоманова, г. Киев, Украина
ymallia@gmail.com

Беляй Юрий Петрович

старший преподаватель кафедры теоретических основ информатики
Институт информатики НПУ имени М. П. Драгоманова, г. Киев, Украина
azb@ukr.net

Аннотация. В данном исследовании рассмотрено использование дистанционных технологий обучения в процессе организации научно-исследовательского задания во время изучения современных языков программирования. В статье приведен пример учебного проекта по предмету — «Современные языки программирования». Аргументирована необходимость использования современных информационно-коммуникационных технологий, в частности дистанционных в процессе обучения современных языков программирования, для формирования у студентов навыков XXI века, которые крайне необходимы в процессе программирования (самоорганизация и самодисциплина, навыки коммуникации, навыки работы в команде и т. д.). Указаны структурные единицы учебного проекта в соответствии с использованием технологий дистанционного обучения. Описаны общие характеристики и методика использования соответствующих современных информационно-коммуникационных технологий.

Ключевые слова: современные языки программирования; дистанционные технологии обучения; научно-исследовательское задание; навыки XXI века.

USING DISTANCE LEARNING TECHNOLOGIES IN THE LEARNING PROCESS OF MODERN PROGRAMMING LANGUAGES

Mariia A. Umryk

Phd (pedagogical sciences), associate professor of the Chair of Theoretical Foundations of Computer Science at the Institute of Informatics
National Pedagogical Dragomanov University, Kyiv, Ukraine
ymallia@gmail.com

Yurii P. Biliai

Lecturer of the Chair of Theoretical Foundations of Computer Science at the Institute of Informatics
National Pedagogical Dragomanov University, Kyiv, Ukraine
azb@ukr.net

Abstract. In the research it was reviewed the use of distance learning technologies in the organization of research tasks while studying modern programming languages. The article contains an example of a training project on the subject "Modern programming languages". The authors argue the necessity of the use of modern information and communication technologies, in particular in modern programming languages distance learning, for the formation of students' 21st century skills that are essential in the process of programming (it is skills such as self-organization and self-discipline, communication skills, teamwork skills etc.). It is pointed out the structural units of the training project in accordance with the use of distance learning technologies. It is described the general characteristics and the use of appropriate methods of modern information and communication technologies.

Keywords: modern programming languages; distance learning technologies; research assignment; 21st century skills.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Donald E Knuth, The Art of Computer Programming, Volumes 1–4, Addison-Wesley Professional 1997, 650 p. (in English).
2. Discipline of Programming [Edsger W. Dijkstra] Prentice-Hall Series in Automatic Computation, 1976, 217 p. (in English).
3. Pratt, Terrence W., Programming Languages: Design and Implementation, PrenticeHall, Inc., Englewood Cliffs, New Jersey, 1975. (in English).
4. Sejdametova Z. S. Preparation of software engineers in "Computer" / Z. S. Sejdametova. — Simferopol : Crimean educational and pedagogical state publishing house, 2007 — 480 p. (in Russian).
5. Hryshko L. V. Articles of programming training system for future software engineers : Abstract. Thesis. for obtaining sciences. degree candidate. ped. sciences special. 13.00.02 "Theory and Methods of Teaching (Science)" / L. V. Hryshko. — K., 2009. — 20 p. (in Ukrainian).
6. Kramar Yu. M. Methods and information and tools of analysis and synthesis programming styles: Abstract. Thesis. For obtaining sciences. Degree candidate. tech. Sciences special: 05.13.06 / Kramar Yu. M. — K., 2005. — 19 p. (in Ukrainian).
7. Nechaj O. S. Method and means of monitoring defects designing object-oriented software: Abstract. Thesis. For obtaining sciences. Degree candidate. tech. Sciences special: 01.05.03 / Nechaj O. S. — K., 2010. — 20 p. (in Ukrainian).
8. Teamwork-key success factors [online] / Center for Distance Education. — Available from : http://www.elitarium.ru/2010/03/01/rabota_v_komande.html. — Titlescreen. (in Russian).
9. Morze, N. Evaluation of the quality of e-learning resources [Text] / N. Morze, O. Hlazunova // Scientific Notes Ternopil National Pedagogical University named after Volodymyr Hnatiuk. Avg. Pedagogy / head. eds. G. Tereshchuk; redkol.: V. Kravec', V. Madzihon, M. Ficula [та ін.]. — Ternopil : TNPU, 2008. — № 7. — Pp. 3–19. (in Ukrainian).
10. Umryk M. A. Distance education in school. / Umryk M. A., Nahorodnjuk V. V. // Scientific Annals Nat. ped. Univ of M.P. Dragomanov. Series № 2. Computer-oriented learning systems: sciences. Labor / Redrada. — K. : NPU of M. P. Dragomanov, 2012. — № 13 (20). — Pp. 192–196. (in Ukrainian).
11. IRC [online] / Wikipedia. — Available from : <http://uk.wikipedia.org/wiki/IRC>. — Titlescreen. (in Ukrainian).
12. Team Viewer [online] / Official site of the project. — Available from : <http://www.teamviewer.com/ru/index.aspx>. — Titlescreen. (in Russian).
13. GitHub [online] / Wikipedia. — Available from : <http://uk.wikipedia.org/wiki/GitHub>. — Titlescreen. (in Ukrainian).
14. QR [online] / Wikipedia. — Available from : <http://uk.wikipedia.org/wiki/QR-%D0%BA%D0%BE%D0%B4>. — Titlescreen. (in Ukrainian).