

ВІРТУАЛЬНІ ЛАБОРАТОРНІ РОБОТИ З ПРОГРАМУВАННЯ З ВИКОРИСТАННЯМ КОМПІЛЯТОРІВ ТА ЛЕКСИЧНОГО АНАЛІЗУ КОДУ

Валерій Онищенко, студент групи ПІ-42, Національний Університет «Львівська Політехніка» (НУЛП), Україна

Науковий керівник – **Павло Сердюк**, канд. техн. наук, доцент кафедри програмного забезпечення, НУЛП, Україна

Сучасні можливості комп'ютерних технологій здатні забезпечити доступ до інформації та передачу знань часто більш ефективно ніж традиційні засоби навчання [1]. Створення нових методів навчання та впровадження все більшої кількості нових технологій потрібні для того, щоб підвищити здатність студентів до самонавчання [2].

За стрімким розвитком дистанційного навчання існуючі засоби опитування встигають не повністю, тому є потреба у впровадженні нового програмного забезпечення, оскільки застосування тільки традиційних методів навчання призводить до низького рівня умінь і практичних навичок [3].

Типове дистанційне опитування за допомогою тестів не дає повноцінної характеристики знань студента з дисциплін програмування, оскільки не перевіряє творчої складової, яка необхідна при складанні програм. З іншої сторони перевірка написаних студентом програм потребує значної кількості часу.

У даній роботі запропоновано використовувати вбудовані компілятори для перевірки коду, написаного студентами, у лабораторних віртуальних роботах з програмування. Використання компілятора дозволить зменшити об'єм коду для перевірки правильності виконання завдання та унеможливить наявність помилок у цій перевірці.

Починаючи з Windows Vista платформа .NET Framework входить до складу операційної системи, з цього випливає, що кожен комп'ютер з операційною системою Vista або новішою уже містить компілятор мови с#, що спрощує завдання розробнику програмної системи лабораторних робіт з програмування тому, що нема потреби поставляти компілятор разом з програмою.

Використовувати компілятор можна без середовища розробки програмного забезпечення, за допомогою консолі. Розглянемо на прикладі компіляцію та виконання за допомогою консолі програмного коду виду:

```
using System;

namespace Sample {
    class Program {
        static void Main(string[] args) {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

Лістинг 1. Приклад елементарного програмного коду, вивід на консоль "Hello World!".

Цей код буде збережено програмною системою у тимчасовий файл *test.cs*. Далі програма у консолі виконає наступні дії:

- Відкрити файл *vsvars32.bat* перейшовши перед тим у відповідний каталог. Цей файл задає змінні середовища для підтримки побудови з командного рядка;
- перейти у каталог з *.cs* файлом, після чого компілювати його.

Стрічка, що виконує описані вище дії може виглядати так:

```
cd /d C:\Program Files (x86)\Microsoft Visual Studio 11.0\Common7\Tools & vsvars32.bat & cd /d C:\ & csc /t:exe test.cs
```

Після успішної компіляції у каталозі, поруч з файлом *test.cs* має з'явитись файл з тою ж назвою та розширенням *.exe*. Для виконання файлу *test.exe* програмна система виконає у консолі: *test.exe*, результатом буде поява на екрані тексту: *Hello World!* Результат виводу на екран порівнюватиметься з очікуваним результатом, і якщо вони співпадають, то користувачеві буде запропоновано наступне завдання лабораторної роботи.

Лексичний аналізатор потрібний у ситуаціях, коли необхідно штучно обмежити можливості студента, що виконує лабораторну роботу та повідомити його про те, що він обрав не правильний шлях для виконання лабораторної роботи. Прикладом може слугувати заборона на використання службового оператора *Console.WriteLine(“”)*. Якщо користувач використовуватиме даний оператор у своєму кодї, компілятор це пропустить, бо код буде синтаксично правильним, проте студент зможе вивести будь яке повідомлення на екран і тим самим втратиться сенс проходження лабораторної роботи, тому необхідно заборонити використання цього оператора і в залежності від специфіки завдання заборонити деякі інші чи навпаки зробити їх наявність обов'язковою.

Розроблена програмна система дозволяє розширити можливості дистанційного навчання та підвищити рівень знань та умінь студентів. Використання компілятора та лексичного аналізатора у програмній системі дозволяє більш ефективно перевіряти творчу складову знань з програмування.

Література

1. Мигович С. М. Огляд сучасного стану дистанційного навчання в Україні / С. М. Мигович // Науковий вісник НУБіП серія "Педагогіка, психологія, філософія" випуск 160. — Київ, 2012.
2. Педагогіка: учеб. пособие для студ. пед. учеб. завед. / В. А. Слостенин, И. Ф. Исаев, А. И. Мищенко, Е. Н. Шиянов. – М. : Школьная Пресса, 2002. 512 с
3. М. О. Роздобудько, О. В. Бордюг. Електронна лабораторна робота з фізики як засіб формування інформаційної компетентності учнів / М. О. Роздобудько // Управління якістю підготовки майбутніх учителів фізики та трудового навчання випуск 15. — К-П, 2009. 96 с