

УДК 377.04

**Базурін Віталій Миколайович**

кандидат педагогічних наук, доцент кафедри професійної освіти та комп'ютерних технологій  
Глухівський національний педагогічний університет імені Олександра Довженка, м. Глухів, Україна  
ORCID ID 0000-0002-6614-4889  
u-3700@ukr.net

## СЕРЕДОВИЩА ПРОГРАМУВАННЯ ЯК ЗАСІБ НАВЧАННЯ УЧНІВ ОСНОВ ПРОГРАМУВАННЯ

**Анотація.** У статті розкриваються умови вибору середовища програмування як засобу навчання учнів загальноосвітньої школи програмування сучасними мовами. Визначено основні умови, які впливають на вибір середовища програмування: технічні характеристики комп'ютерів і системні вимоги середовища програмування; наявність операційних систем і додаткового програмного забезпечення, необхідного для функціонування програмного середовища; функціонал програмного середовища; інтерфейс програмного середовища; наявність документації на програмне середовище; наявність навчально-методичного забезпечення; рівень компетентності вчителя інформатики. Проаналізовано характеристики найбільш поширених середовищ програмування мовами C/C++, C#, Java. Обґрунтовано вибір середовища програмування для вивчення кожної з указаних мов програмування для навчання програмістів-початківців і учнів, які вже мають навички розроблення програм.

**Ключові слова:** мова програмування; середовище програмування; інтерфейс; функціонал.

### 1. ВСТУП

**Постановка проблеми.** Стрімке зростання ринку праці у сфері ІТ зумовлює збільшення потреби у кваліфікованих фахівцях, які володіють сучасними мовами програмування. Однак чинним Державним стандартом з інформатики і навчальними програмами на вивчення основ алгоритмізації і програмування відведено близько 13% загального навчального часу [1]. Як зазначають науковці [2], цього недостатньо для формування стійких навичок створення програм конкретною мовою програмування. Саме тому доцільно інтенсифікувати процес вивчення мов програмування.

На думку В. Д. Руденка [1], вивчення сучасних мов програмування (C++, C#, Java тощо) доцільно здійснювати у старших класах, у процесі вивчення профільних спецкурсів.

Однією з вагомих проблем вивчення програмування у шкільному курсі інформатики є вибір середовища програмування. Перший етап навчання конкретної мови програмування полягає саме в ознайомленні з інтерфейсом середовища програмування. Від вибору середовища програмування значною мірою залежить ефективність вивчення сучасних мов програмування, більшість з яких передбачають створення екранних форм і використання візуальних компонентів. Рационально побудований інтерфейс середовища програмування значно спрощує процес засвоєння учнями початкових навичок роботи з даним середовищем, здійснення основних операцій з проектом і файлами, що містять текст програми та інші складові проекту.

У той же час застосування середовища програмування, яке має зручний і рационально скомпонований інтерфейс, зводить до мінімуму час пошуку необхідних компонентів, а також час виконання рутинних операцій (збереження файлів, компіляція програми тощо). Важливу роль відіграє також відносна швидкодія середовища програмування, тобто час створення, збереження та компілювання програми.

**Аналіз останніх досліджень і публікацій.** Проблемам застосування інформаційно-комунікаційних технологій у навчанні присвячені праці М. І. Жалдака,

Ю. С. Рамського, Н. В. Морзе, В. В. Лапінського, С. А. Ракова, Ю. В. Триуса, В. Д. Швець [3] та інші.

Методичні аспекти навчання програмування у середній школі та вищому навчальному закладі досліджують М. І. Жалдак, Н. В. Морзе [4], Ю. С. Рамський, О. І. Кривонос, В. Д. Руденко, С. О. Семеріков, О. М. Спірін [5], П. Г. Шевчук [6-8] та інші.

Зокрема, Н. В. Морзе виділяє ознайомлення із середовищем програмування в окремий етап навчання програмування.

Так, у дослідженнях О. М. Спіріна і П. Г. Шевчука розглядаються програмні технології операційної системи Windows 8 з погляду навчання програмування [5], відмінні риси мови програмування С# [6], програмно-технологічні умови застосування мов програмування у загальноосвітніх навчальних закладах [7]. У працях Ф. С. Ільєсової розкриваються особливості навчання мовою програмування Java [9].

Вибору мови програмування для навчання у школі присвячені праці Я. М. Глинського, Л. В. Палюшок [10], С. С. Жуковського, О. В. Коротун [11] та ін.

**Мета статті** – проаналізувати дидактичні можливості найбільш поширених середовищ програмування для навчання програмування учнів загальноосвітніх шкіл.

## 2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 2.1. Основні чинники, які впливають на вибір середовища програмування для застосування у процесі навчання програмування учнів середніх шкіл

Одним із завдань навчання інформатики в основній школі Державний стандарт визначає розвиток алгоритмічного мислення як засобу планування та організації діяльності [12, с. 27]. Важливим завданням навчання інформатики у старшій школі, за Державним стандартом, є формування в учнів здатності будувати і використовувати інформаційні моделі, а також засоби опису і моделювання явищ і процесів [12, с. 27].

Учні повинні знати етапи процесу описання алгоритмів, у тому числі за допомогою програмних засобів, способи описання алгоритмів, призначення базових алгоритмічних структур, програмного коду, розуміти поняття формалізації задачі, поняття комп'ютерного моделювання, процес перетворення даних за формальними правилами, призначення комп'ютерної програми і пояснювати етапи її виконання на комп'ютері, уміти визначати властивості об'єктів і значення цих властивостей, здійснювати класифікацію і розпізнавати об'єкти, здійснювати формалізований опис об'єкта, реалізувати виокремлення підзадач в опису задачі, уміти складати алгоритми для різних виконавців [12, с. 86-87].

Тому основи алгоритмізації та програмування становлять одну із змістових ліній шкільного курсу інформатики, яка є наскрізною для всього курсу. У процесі її вивчення учні засвоюють види алгоритмічних конструкцій, прийоми розробки алгоритмів, синтаксис конкретної мови програмування, інтерфейс програмного середовища, а також прийоми створення програми у відповідному програмному середовищі. Широкий спектр сучасних мов програмування надає вчителю можливість самостійно вибирати мову програмування для вивчення у школі.

Вибір мови програмування є складною психолого-педагогічною проблемою. Дослідники рекомендують використання різних мов програмування залежно від мети вивчення програмування. Одні аргументують вибір мови програмування її простотою, інші – наявністю навчально-методичного забезпечення, інші – актуальністю мови програмування і її поширенням. Дослідженню даного аспекту присвячена значна кількість праць науковців [2], [6].

У шкільному курсі інформатики доцільно розглядати процедурну та об'єктно-орієнтовну парадигми програмування. У рамках даних парадигм найбільш поширеними мовами є C/C++, Basic, Visual Basic, Pascal, Object Pascal, C#, Java, Python та інші.

Перелічені мови програмування можна умовно поділити на три групи відповідно до особливостей синтаксису: 1) мови на основі Basic; 2) мови на основі Pascal; 3) мови на основі C [7]. Мови, синтаксис яких базується на синтаксисі C, є найбільш поширеними, тому, як зазначає П. Г. Шевчук, вивчення мов цієї групи дозволяє уникнути значних проблем під час переучування в майбутньому [7, с. 31].

На думку П. Г. Шевчука, на вибір мови програмування впливає низка програмно-технологічних умов, до яких належать:

- технічні характеристики обчислювальної техніки, яка має відповідати системним вимогам програмного середовища;
- наявність операційних систем і додаткового програмного забезпечення, необхідних для повноцінного функціонування середовищ програмування (IDE (англ.) – Integrated Development Environment);
- наявність і доступність середовищ програмування, які доцільно використовувати у навчанні;
- рівень компетентності вчителів інформатики, який необхідний для навчання програмування на відповідній мові [8, с. 81–82].

Отже, наявність і доступність програмного середовища відіграють велике значення у виборі мови програмування для вивчення у школі. Причому середовище програмування спочатку є об'єктом вивчення, а в подальшому – засобом навчання.

На думку П. Г. Шевчука, важливими характеристиками середовища програмування, які впливають на його вибір з навчальною метою, є поширеність, доступність, особливості інтерфейсу, спосіб реалізації, системні вимоги, методична підтримка [7, с. 31]. Учений робить висновок про те, що вибір середовища програмування є важливим елементом системи навчання програмування.

Підсумовуючи сказане, відзначимо, що на вибір середовища програмування для використання у процесі вивчення мови програмування впливають такі умови:

- технічні характеристики комп'ютерів і системні вимоги середовища програмування;
- наявність операційних систем і додаткового програмного забезпечення, необхідного для функціонування програмного середовища;
- функціонал програмного середовища;
- інтерфейс програмного середовища;
- наявність документації на програмне середовище;
- наявність навчально-методичного забезпечення;
- рівень компетентності вчителя інформатики.

Розглянемо перелічені умови детальніше.

Технічні характеристики обчислювальної техніки мають виключно важливе значення для вибору програмного середовища. Якщо комп'ютерні класи у школі мають низькі характеристики, то на них не можна встановити сучасну операційну систему, яка потребує потужних ресурсів. Сучасні програмні середовища, наприклад, Microsoft Visual Studio версій 2013, 2015 і 2016, мають високі системні вимоги. Для комфортної роботи у даному програмному середовищі комп'ютер повинен мати досить високі характеристики апаратного забезпечення, і це негативно впливає на вибір даного програмного середовища – на третині комп'ютерів недоцільно використовувати дане середовище. Комп'ютер буде виконувати повільно операції компілювання, «зависати», втрачати програму. Зазвичай, це викликає зниження інтересу учнів до вивчення мови програмування, а отже, й ефективності процесу навчання програмування.

Доцільнішим є вибір програмного середовища, яке має нижчі системні вимоги. На нашу думку, системні вимоги середовища програмування мають бути орієнтовно такі: процесор 2–2,2 ГГц, обсяг оперативної пам'яті до 1 ГБ, обсяг місця на диску до 1 ГБ, операційна система Microsoft Windows XP/ Seven / 8.1 / 10. Вищі системні вимоги значно звужують коло комп'ютерів, на яких можна комфортно працювати у середовищі програмування. Нижчі системні вимоги, як правило, мають середовища програмування з обмеженим функціоналом.

Даний чинник звужує можливості використання у навчальному процесі потужних середовищ програмування, які мають високі системні вимоги.

Другий чинник – наявність операційних систем і додаткового програмного забезпечення – також відіграє важливе значення. Більшість програмних середовищ вимагають наявності на комп'ютері встановленого додаткового програмного забезпечення – середовища виконання програм Microsoft Framework (Microsoft Visual Studio, SharpDevelop, Delphi) або віртуальної машини Java Development Kit (Eclipse, NetBeans, JBuilder та інші). Додаткове програмне забезпечення, зазвичай, знаходиться у вільному доступі на сайті фірми – виробника програмного середовища або операційної системи. Проте можливі конфлікти між різними програмними засобами, які поширюються вільно.

Значну роль у виборі програмного середовища відіграє його функціонал. Для того, щоб виступати ефективним засобом навчання, функціонал програмного засобу повинен відповідати змісту навчання. Проте трапляються випадки, коли більший функціонал програмного середовища спонукатиме учнів по подальшого самостійного вивчення програмування.

Наприклад, повна версія Microsoft Visual Studio Standard Edition платна, причому її ціна досить висока. Безкоштовна версія Microsoft Visual Studio Express Edition має деякі обмеження стосовно функціоналу, однак дані обмеження не відіграють великого значення під час навчання, оскільки стосуються лише тих особливостей, які у школі, зазвичай, не вивчаються.

Обмежений функціонал програмного середовища SharpDevelop майже не впливає на його дидактичні можливості, адже стає значущим лише під час розроблення складних програм, які взаємодіють з базами даних тощо. Однак функціональних можливостей SharpDevelop версії, починаючи з 4.0 і вище, цілком достатньо для навчання програмування мовою C# і створення програм з досить складним алгоритмом.

Інший приклад – Eclipse і NetBeans. У середовищі Eclipse можна розробляти повнофункціональні програми мовою Java і перетворити програму в EXE-файл, але неможливо розмістити компоненти на екранній формі у візуальному режимі. У середовищі NetBeans є візуальний режим роботи з екранною формою, однак відсутня функція перетворення програми на EXE-файл.

Під час вибору програмного середовища слід враховувати також особливості інтерфейсу. До таких особливостей слід віднести:

- вид інтерфейсу (графічний або командний рядок);
- раціональне компонування інтерфейсу. Доступ до основних функцій і до основних компонентів повинен бути якомога простішим;
- мова інтерфейсу значно полегшує процес засвоєння прийомів роботи у програмному середовищі;
- наявність підказки значно спрощує процес виправлення помилок. Тому для учнів-початківців доцільно використовувати програмне середовище з україномовним інтерфейсом. Для учнів, що мають навички створення програм, можна використати програмне середовище з англійським інтерфейсом;

- наявність документації на програмне середовище. Цей чинник впливає на швидкість засвоєння програмного середовища вчителем;
- наявність відповідного методичного забезпечення залежить від поширення програмного середовища у навчальних закладах.

Однією з характеристик програмного середовища є можливість роботи з візуальними компонентами, яка дозволяє розробляти програми у візуальному режимі. Використання консольного введення-виведення ускладнює процес засвоєння учнями основ роботи у середовищі програмування і значно сповільнює процес розробки програм. Це негативно впливає на рівень інтересу учнів до програмування.

Навпаки, графічний інтерфейс сприяє активізації роботи учнів. Учні змінюють положення компонентів на формі і відразу бачать внесені зміни. Це сприяє підтриманню інтересу учнів до програмування на необхідному рівні, мотивує їх на подальшу роботу.

Раціональне компонування інтерфейсу впливає позитивно на вивчення програмного середовища і на процес розробки програм учнями. Графічний інтерфейс середовища Delphi можна вважати одним із найбільш раціонально скомпонованих. Основні компоненти екранних форм розміщені на вкладках, їх властивості доступні у вікні Інспектора об'єктів. У процес створення екранної форми учень може встановити форму на екрані так, як йому зручно. У середовищах Microsoft Visual Studio і Sharp Developer положення форми на екрані у процесі розробки програми змінити не можна, доступ до компонентів форми і їх властивостей ускладнений. Це знижує цінність даних середовищ програмування для навчання учнів програмування.

Україномовний інтерфейс значно спрощує процес навчання основ роботи у програмному середовищі і знижує поріг входження для мови програмування. У той же час мова інтерфейсу має для програмних середовищ не настільки критичне значення, як для прикладного програмного забезпечення, наприклад, текстових редакторів. У програмному середовищі учень, зазвичай, працює з кодом програми, а не з форматуванням тексту. Кількість елементів інтерфейсу досить обмежена, тому їх досить просто запам'ятати. Отже, мова інтерфейсу є швидше початковим бар'єром. Після його подолання вплив мови інтерфейсу на успішність вивчення мови програмування поступово нівелюється.

Наявність підказки значно підвищує цінність програмного середовища як засобу навчання. У процесі засвоєння прийомів написання програми учні роблять значну кількість помилок. Дуже важливо, щоб вони навчилися самостійно виправляти помилки. Україномовна або російськомовна підказка спрощує процес формування навичок пошуку і виправлення помилок у програмах. Це знижує поріг входження під час вивчення мови програмування.

Документація на програмне середовище може мати вигляд інструкцій користувача або навчального посібника з користування даним середовищем. У більшості випадків наявність документації більш важлива для учителя, ніж для учнів. Учні у більшості своїй не користуються документацією. Для вчителя наявність такої документації дозволяє швидше опанувати середовище програмування і, відповідно, швидше навчити програмуванню учнів. Для учнів, які оволодівають навичками роботи у потрібному середовищі програмування на поглибленому рівні, документація від розробника також важлива, адже у підручниках і навчальних посібниках, зазвичай, розкривається не весь функціонал програмних засобів.

Наявність відповідного методичного забезпечення пов'язана з поширенням програмного середовища. Чим частіше використовується програмне середовище, тим більша кількість методичного забезпечення наявна у бібліотеках і в мережі Інтернет. Наявність методичного забезпечення має важливе значення і для вчителя, і для учнів.

До методичного забезпечення відносять підручники, навчальні посібники і методичні рекомендації. В ідеалі, звичайно, мати підручник для навчання програмування конкретною мовою у конкретному програмному середовищі. Однак, як показує практика, написані підручники, зазвичай, застарівають досить швидко. Тому підручник може частково застаріти.

Методичні рекомендації може розробити сам учитель залежно від того, як він бачить вивчення даної мови програмування, на які питання слід звернути увагу, а які можна опустити.

Широта спектру методичного забезпечення, зазвичай, зростає з поширенням програмного середовища і впровадженням його у навчальний процес. Чим більша кількість навчальних закладів використовує певне програмне середовище, тим більшу кількість методичного забезпечення буде розроблено науковцями, методистами і вчителями.

Останній чинник – рівень компетентності вчителів інформатики, необхідний для вивчення мови програмування – безпосередньо пов'язаний з двома попередніми чинниками. Після впровадження у навчальний процес з інформатики конкретної мови програмування з використанням відповідного програмного середовища потрібен певний час для того, щоб учитель оволодів навичками програмування на тому рівні, який необхідний для ефективного здійснення навчального процесу.

Отже, на вибір програмного середовища впливає низка умов, проте важко сформулювати універсальні правила для вибору середовища програмування. В одному випадку слід враховувати один критерій, в іншому випадку – інший критерій тощо.

Середовища програмування, які буде розглянуто далі, можна умовно поділити на дві категорії: вузько спеціалізовані та універсальні. Вузько спеціалізованими є ті середовища програмування, які надають можливість розробляти програми лише однією конкретною мовою програмування. Універсальні середовища програмування надають можливість розробляти програми на кількох мовах. У подальшому дослідженні ми розглядатимемо характеристики універсальних середовищ у контексті вивчення конкретної мови програмування.

## **2.2. Аналіз дидактичних можливостей середовищ програмування мовою C/C++**

Мова C і її подальший розвиток (C++) – це потужна мова програмування, яка підтримує процедурну й об'єктно-орієнтовану парадигми програмування. Мови C/C++ широко використовуються для створення як прикладного, так і системного програмного забезпечення. Орієнтованість на розробку системного програмного забезпечення проявляється, зокрема, у можливості працювати безпосередньо з комірками пам'яті. На таких особливостях мови програмування доводиться окремо зупинятися під час навчання, наголошувати, наприклад, на тому, що в процесі зчитування даних слід додавати відповідний оператор. Попри це, існує специфіка зчитування даних з файлу і запису даних у файл. Зазвичай, робота з файлами викликає деякі утруднення для учнів (у даному випадку учні працювали в середовищі DEV C++).

Мова C++ відноситься до професійних мов програмування, тому людина, яка вивчила дану мову, у подальшому буде переучуватися менше, ніж та людина, що вивчила мову Pascal або Basic. Значна кількість сучасних мов програмування мають синтаксис, що базується на C++.

Специфіка навчання програмування мовою C полягає в тому, що введення-виведення даних здійснюється у консольному режимі. Введення-виведення даних у мовах C може бути форматуваним і неформатованим. У форматованому введенні-виведенні даних слід вказати специфікатор змінної, з якою здійснюється ця операція.

Тому в процесі навчання програмування мовою С слід спочатку сформувати в учнів навички написання команд введення-виведення даних, а лише після цього приступати до вивчення основних алгоритмічних конструкцій. У мові С++ є можливість використовувати більш спрощені конструкції консольного введення-виведення даних.

У процесі створення програми учні оперують з текстом програми, здійснюють налагодження програмного коду і його компіляцію у виконуваний файл. Форма як об'єкт роботи учнів наразі відсутня, і візуальні компоненти не використовуються. Тому інтерфейс середовища програмування повинен надавати доступ до основних операцій з файлами і фрагментами коду. Найбільш поширені середовища мови програмування, зазвичай, забезпечують виконання даної умови, отже, значно більшу роль відіграють функціонал і системні вимоги середовищ програмування.

Якщо проаналізувати інші умови, то можна відзначити, що методичне забезпечення до уроків програмування у конкретному середовищі майже відсутнє у відкритому доступі. Причиною тому є, на нашу думку, те, що роль інтерфейсу середовища програмування не надто важлива, а їх функціонал, зазвичай, подібний.

Найбільш поширені середовища програмування мовами С/С++ є Borland С++, DEV С++, Microsoft Visual Studio, Embarcadero RAD Studio, CodeBlocks та ін.

Середовище програмування Borland С++ 3.0 досі використовується у деяких навчальних закладах завдяки своїй простоті і невибагливості до обчислювальних ресурсів, проте морально застаріло. Воно розраховане на роботу в середовищі операційної системи MS DOS, яка нині майже ніким не використовується. Під час компіляції програм запускається консольний режим. Застарілий інтерфейс значно звужує перспективи використання середовища Borland С++ у навчальному процесі. Однак програми, створені у цьому середовищі, зазвичай працюють і в операційних системах родини Windows. Попри вказані недоліки, можна вказати ще один – підказка англomовна.

Розглянемо Microsoft Visual Studio. На сьогодні Microsoft пропонує версію середовища Visual Studio 2017 у трьох версіях: Community, Professional та Enterprise. Версія Visual Studio Community представляє для навчальних закладів найбільший інтерес, так як розповсюджується безкоштовно, потребує менше системних ресурсів і має достатній функціонал для навчання програмування. Більш ранні версії Visual Studio існували також у версіях Common Edition і Express. Версія Microsoft Visual Studio Common Edition була платна, а Express, подібно до Community, безкоштовна.

Системні вимоги Microsoft Visual Studio Community 2015:

- операційна система Windows 10/8.1/8/7 SP1/Server 2012 R2/Server 2012/Server 2008 R2 SP1;
- частота процесора 1,6 ГГц;
- обсяг оперативної пам'яті 1 ГБ (1,5 ГБ у випадку використання віртуальної машини);
- обсяг вільного місця на диску 4 ГБ;
- сумісний з DirectX 9 відеоадаптер, який підтримує роздільну здатність 1024x768 або більшу [13].
- Версії Microsoft Visual Studio Enterprise 2015 і Microsoft Visual Studio Professional 2015:
- операційна система Windows 10/8.1/8/7 SP1/Server 2012 R2/Server 2012/Server 2008 R2 SP1;
- частота процесора 1,6 ГГц;
- обсяг оперативної пам'яті 1 ГБ (1,5 ГБ у випадку використання віртуальної машини);
- обсяг вільного місця на диску 10 ГБ;

- сумісний з DirectX 9 відеоадаптер, який підтримує роздільну здатність 1024x768 або більшу [13].

Як альтернативу можна використати Microsoft Visual Studio Express. Ця версія повністю безкоштовна, має деякі обмеження стосовно програмних бібліотек і функціоналу, проте в процесі вивчення мов C/C++ ці обмеження не мають суттєвого значення.

Microsoft Visual Studio Express 2015 має такі системні вимоги:

- операційна система Windows 10/8.1/8/7 SP1/Server 2012 R2/Server 2012/Server 2008 R2 SP1;
- частота процесора 1,6 ГГц;
- обсяг вільного місця на диску 5 ГБ;
- сумісний з DirectX 9 відеоадаптер, який підтримує роздільну здатність 1024x768 або більшу [13].

На нашу думку, використовувати це середовище для навчання програмування мовою C++ недоцільно через його високі системні вимоги. У той же час вивчення даного середовища є бажаним, якщо мова C/ C++ є лише початковим етапом у вивченні програмування і планується поступовий перехід до потужніших і сучасніших мов (наприклад, C#).

Безкоштовні середовища програмування, які поширюються за ліцензією GNU/GPL, подібні між собою і за інтерфейсом, і за функціоналом. Єдиний їх недолік, але несуттєвий, – підказка англійською мовою. Проте учні починають вивчати англійську мову раніше за програмування, тому даний недолік, на нашу думку, не є критичним. Саме тому важко переоцінити роль методичного забезпечення, адже в підручнику, посібнику або методичних рекомендаціях повинні розкриватися особливості роботи у конкретному середовищі програмування.

Середовище програмування Bloodshed DEV C++ належить до класу вільно поширюваного програмного забезпечення. Його системні вимоги досить низькі, воно не вимагає для комфортної роботи потужного процесора або значного обсягу оперативної пам'яті. Bloodshed DEV C++ розраховане на роботу в середовищі 32-розрядних операційних систем родини Windows (XP, Vista, Seven). Рекомендований об'єм оперативної пам'яті 128 МБ, частота процесора – 1 ГГц.

Мова інтерфейсу – українська, проте підказка – англомовна. Це значно ускладнює процес налагодження програми програмістами-початківцями.

У даний час нові версії середовища програмування Bloodshed DEV C++ не розробляються. Проект підтримується SourceForge [14]. Останньою версією на момент написання статті є 5.11.

На момент написання статті для середовища програмування DEV C++ не знайдено документації, а існує методичне забезпечення, у більшості, англійською та російською мовою [15], [16].

Розвитком даного середовища є wxDev-C++ [17]. Дане середовище включає всі можливості DEV C++ і додатково містить дизайнер форм. У мережі Інтернет є документація, проте англійською мовою [18].

Середовище програмування Code::Blocks існує у кількох версіях, які призначені для операційних систем Windows / Mac / Linux [19; 20]. На момент написання статті актуальною версією є 16.01. Дане середовище програмування поширюється за ліцензією GNU/GPL. Мова інтерфейсу і мова підказки – англійська. Системні вимоги: операційна система Windows (NT/2000, XP, Seven, Vista, Seven, 8/8.1), Mac (Mac OS X), Unix. Для функціонування середовища програмування необхідно додатково встановити компілятор GCC, MSVC++ або інший.



На сайті фірми-виробника присутня документація щодо роботи з даним програмним засобом, проте англomовна. Методичного забезпечення у вільному доступі не знайдено.

Середовище програмування CodeLite є мультиплатформним і також поширюється за ліцензією GNU/GPL [21]. Його функціональні можливості значно ширші, ніж у середовищ DEV C++, wxDev-C++, Code::Blocks. Інтерфейс CodeLite інтуїтивно-зрозумілий, у стилі Windows. Мова інтерфейсу і мова підказки – англійська.

Висновок, що у випадку, коли вивчення програмування обмежено мовами C/C++, то доцільно використати DEV C++ або Code::Blocks. У тому випадку, коли C або C++ є лише базою для подальшого вивчення програмування, а в подальшому планується перехід до C# або інших мов, то варто використати Microsoft Visual Studio Express Edition. Учні отримають навички роботи у середовищі Microsoft Visual Studio Express Edition, і в процесі вивчення C# етап ознайомлення з програмним середовищем можна вважати пройденим. У цьому випадку процес вивчення мови C# помітно спрощується.

### **2.3. Аналіз дидактичних можливостей середовищ програмування мовою C#**

Мова C# набула значного поширення останнім часом. Проблемам навчання учнів програмування мовою C# присвячені дослідження О. М. Спіріна, П. Г. Шевчука [6] та ін.

Мова C# розроблена і вдосконалюється фірмою Microsoft. C# має синтаксис, подібний до синтаксису C++. Структура програми аналогічна до структури програми мовою C++.

На основі аналізу встановлено, що найбільш поширеними є такі середовища програмування мовою C#: Microsoft Visual Studio, SharpDevelop, MonoDevelop та інші.

Як ми вже зазначали у попередньому параграфі, середовище Microsoft Visual Studio є одним із найбільш поширених. Нові версії випускаються раз на рік або раз на два роки.

У даному середовищі створення екранної форми і додавання на форму візуальних компонентів здійснюється за допомогою Конструктора форм. Усі компоненти знаходяться на палітрі компонентів. Вікно властивостей об'єктів відкрито не постійно, а викликається у разі потреби. Інтерфейс англomовний, підказка теж англomовна.

Інтерфейс середовища програмування Microsoft Visual Studio інтуїтивно-зрозумілий. Системні вимоги ми вказали раніше. До переваг даного середовища можна віднести наявність підтримки, документації англійською і російською мовами, наявність значної кількості уроків у мережі Інтернет, постійні оновлення на сайті фірми-розробника.

Дане середовище містить більшість потрібних файлів бібліотек, які можна підключати в міру необхідності. Microsoft Visual Studio є професійним середовищем, у ньому можна створювати повноцінні, повнофункціональні програмні засоби.

Microsoft Visual Studio – одне з кількох середовищ розробки програм мовою C#. Стосовно написання програм на C#, то Microsoft Visual Studio має більші переваги перед іншими, оскільки повнофункціональним є лише воно.

Безкоштовна версія – Microsoft Visual Studio Express – має деякі функціональні обмеження, проте вони виявляються лише програмістами-професіоналами. Для навчання програмування учнів загальноосвітніх шкіл цілком достатньо функціональних можливостей Microsoft Visual Studio Express.

Як ми вже зазначали, Microsoft Visual Studio Express знаходиться у вільному доступі на сайті фірми Microsoft. Необхідне для його роботи середовище виконання програм Microsoft Framework .NET також знаходиться у вільному доступі на сайті фірми Microsoft.

Єдиний суттєвий недолік Microsoft Visual Studio Express Edition – високі системні вимоги.

SharpDevelop – безкоштовна альтернатива Microsoft Visual Studio. Дане програмне середовище простіше за Microsoft Visual Studio, проте функціональні можливості його аналогічні. Для роботи SharpDevelop потребує встановлення Microsoft Framework .NET відповідної версії [14].

У мережі Інтернет присутні уроки програмування мовою C# у середовищі SharpDevelop, проте англійською та російською мовами.

Системні вимоги SharpDevelop значно нижчі, ніж у Microsoft Visual Studio. Відповідно, дане програмне середовище досить швидко працює на більшості комп'ютерів. Інтерфейс SharpDevelop має відмінності від інтерфейсу Microsoft Visual Studio. Функціонал SharpDevelop дещо обмежений, проте ці обмеження суто специфічні: наприклад, підключення файлів бібліотек, необхідних для створення SMTP-з'єднань.

Середовище програмування MonoDevelop розраховане на роботу в операційних системах Linux, Unix та Windows [22]. Даний програмний засіб можливо інтегрувати з Microsoft Visual Studio. Окрім програм мовою C#, надає можливість створювати програми мовами C, C++, Java та іншими. Дане середовище має гнучкі налаштування для створення і компіляції проекту. Проте для програміста-початківця ці налаштування швидше ускладнюють роботу, ніж спрощують її.

Інтерфейс MonoDevelop подібний до Sharp Develop. Мова інтерфейсу англійська, підказка також англійська. Під час налагодження програми можна встановити точку зупинки, покроковий режим налагодження, відслідковувати значення змінних і комірок пам'яті.

На основі аналізу дидактичних можливостей програмних середовищ слід відзначити, що в навчальному процесі доцільним є використання середовища SharpDevelop. Маючи достатньо широкий функціонал, дане середовище забезпечує більшу швидкодію у роботі, ніж Microsoft Visual Studio.

#### **2.4. Аналіз дидактичних можливостей середовищ програмування мовою Java**

Мова Java на даний час є однією з найбільш поширених мов програмування. Її було розроблено фірмою Sun Microsystems на початку 90-х років минулого століття, і дана мова продовжує розвиватися і вдосконалюватися.

Найбільш поширеними середовищами програмування мовою Java є IntelliJ IDEA, Eclipse, NetBeans, JBuilder. Для роботи кожного з названих середовищ необхідне встановлення платформи JDK і пакета JRE, тому ми не будемо розглядати дану особливість середовища програмування (вона характерна для всіх засобів розроблення програм мовою Java).

Розглянемо детальніше середовища програмування мовою Java.

IntelliJ IDEA належить до комерційних середовищ розробки і підтримує мови програмування Java, Python, Scala та інші. Розробник IntelliJ IDEA – фірма JetBrains – випускає середовище у двох версіях – Community Edition (безкоштовна, що має обмежений функціонал) і Ultimate Edition (платна, має повний функціонал). Дане середовище має широкий функціонал, допускає розробку програм для різних платформ (Windows, Mac, Android, Linux) [23].

Версія Ultimate Edition надає можливість створювати екранну форму і редагувати її у візуальному режимі, підтримує мови Java, Python, PHP, Ruby та інші. Версія Community Edition допускає лише роботу з кодом програми і лише одну мову програмування – Java.

З огляду на вказані обмеження у навчальному процесі доцільно використовувати версію IntelliJ IDEA Ultimate Edition, проте її ціна значно зужує можливості застосування даного середовища програмування з навчальною метою.

Стосовно методичного забезпечення, то шкільний підручник «Вивчаємо Java у школі» [24] розкриває особливості розробки програм саме в середовищі IntelliJ IDEA.

Середовище програмування Eclipse орієнтоване на розробку програм мовою Java, проте існують додаткові модулі для розробки програм іншими мовами [25]. Наприклад, модуль CDT призначений для написання програм мовами C і C++. Середовище програмування Eclipse підтримується фірмою Eclipse Foundation. Eclipse є платформо-незалежним середовищем, яке здатне працювати на різних платформах і в різних операційних системах.

Це середовище є, на нашу думку, більше професійним програмним засобом, оскільки припускає одночасну роботу кількох програмістів над одним проектом. Корисною для навчання є можливість компіляції проекту у файл з розширенням EXE.

Інтерфейс Eclipse англomовний, підказка також англomовна. Існують навчальні посібники щодо розроблення програм мовою Java у середовищі Eclipse [26; 27].

До суттєвих недоліків Eclipse слід віднести неможливість створення форм у візуальному режимі. Це значно ускладнює процес створення програм програмістами-початківцями. Водночас мова Java має низку методів для створення форми і компонентів на формі засобами програмного коду, для чого необов'язкова наявність візуального режиму.

Отже, поріг входження для оволодіння програмуванням у середовищі Eclipse досить високий, тому його успішно подолає менша кількість учнів.

Стосовно документації на програмне забезпечення, то на сайті проекту [25] її знайти досить важко. У той же час методичне забезпечення (в основному, російською мовою) у мережі Інтернет знайти досить просто.

Середовище програмування JBuilder розроблено фірмою Borland, подальша підтримка здійснюється фірмою CodeGear. Інтерфейс даного програмного засобу англomовний, підказка також англomовна.

Системні вимоги JBuilder 2008:

- рекомендований обсяг оперативної пам'яті 1,5 ГБ;
- розмір місця на диску 1,2 ГБ;
- процесор Intel Pentium IV 4/ 2.4 МГц або вище;
- привід DVD-ROM;
- роздільна здатність екрану 1024x768, кількість кольорів – 256;
- операційна система Windows XP, Windows Vista, Mac OS X, Red Hat Enterprise Linux 4 [28, с. 22].

Під час запуску середовище завантажується досить тривалий час, проте швидкодія його після завантаження порівняно висока. Візуальний режим редагування форми відсутній. Це значно ускладнює процес розробки програм початківцями, а отже, і підвищує поріг входження.

Для даного програмного засобу можна знайти і документацію, і методичне забезпечення [28], [29].

Середовище програмування JBuilder доцільно використати для вивчення програмування мовою Java на поглибленому рівні.

Наступне середовище – NetBeans – належить до вільно-поширюваного програмного забезпечення. Інтерфейс його англomовний, підказка також англomовна. У середовищі NetBeans доступний візуальний режим створення форм і розміщення на них компонентів. Для навчання програміста-початківця це дуже важливо, оскільки він має можливість бачити зовнішній вигляд форми програми у процесі розроблення.

Системні вимоги NetBeans 8.0 допускають встановлення середовища на більшість комп'ютерів і комфортну роботу у ньому. Системні вимоги:

- процесор Intel Pentium III 800 МГц або еквівалент;
- оперативна пам'ять 512 МБ;
- обсяг вільного місця на диску 750 МБ [30].

Середовище програмування NetBeans 8.0 доцільно використати для навчання початківців, так і тих учнів, які вже мають навички розробки програм мовою Java.

Висновок: для навчання програмістів початківців доцільно використати NetBeans IDE, а для навчання програмування у класах з поглибленим вивченням інформатики – IntelliJ IDEA або Eclipse.

### 3. ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Середовища програмування – важливий засіб навчання програмування учнів загальноосвітніх шкіл. На початковому етапі вивчення конкретної мови програмування середовище програмування є об'єктом вивчення і лише потім – дидактичним засобом. На основі аналізу програмно-технологічних і психолого-педагогічних умов вибору середовища програмування визначено такі середовища, застосування яких більш ефективно сприятиме формуванню в учнів навичок створення програм. Причому, для програмістів-початківців доцільно використовувати одні середовища, а для учнів, що вже мають навички програмування – інші.

У тому випадку, коли планується вивчати кілька мов програмування, доцільно використовувати таке середовище програмування, яке надає можливості розроблення програм більшістю даних мов програмування.

Перспективи подальших досліджень пов'язані з теоретичним обґрунтуванням, розробкою методики навчання учнів програмування сучасними мовами програмування (C++, Java тощо) і впровадження цієї методики у навчальний процес загальноосвітньої школи.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] В. Д. Руденко, “Сучасна комп'ютерна грамотність і проблеми змісту шкільної інформатики”, *Український педагогічний журнал*, №3, с.158-169, 2013.
- [2] В. В. Лапінський, “Проблема вибору першої мови програмування – сьогоднішнє бачення”, *Комп'ютер у школі та сім'ї*, №1, с. 14–17, 2014.
- [3] В. Д. Швець, “Застосування пакету EXCEL для обробки даних лабораторних робіт з фізики”, *Фізика та астрономія в школі*, №6, с. 50 – 53, 2003.
- [4] Н. В. Морзе, *Методика навчання інформатики*, Київ, Україна: Навчальна книга, 2004.
- [5] О. М. Спирін, та П. Г. Шевчук, “Порівняльний аналіз програмних технологій операційної системи Windows 8 для навчання програмування”, *Інформаційні технології та засоби навчання*, №2, с.65-73, 2014. [Електронний ресурс]. Доступно: <http://journal.iitta.gov.ua/index.php/itlt/article/view/997>. Дата звернення: Бер.11, 2017.
- [6] П. Г. Шевчук, “Від Pascal до C#”, *Комп'ютер у школі та сім'ї*, №4, с.47-52, 2011.
- [7] П. Г. Шевчук, “Проблема вибору мови та середовища програмування в якості засобу навчання”, на звітній наук. конфер. Інституту інформаційних технологій і засобів навчання АПН України, Київ, 2010, с. 30-31.
- [8] П. Г. Шевчук, “Програмно-технологічні умови використання мови C# для навчання програмування в загальноосвітніх навчальних закладах”, *Науковий часопис Національного педагогічного університету імені М. П. Драгоманова*, серія 2, Комп'ютерно-орієнтовані системи навчання, Київ, Україна: Вид-во НПУ ім. М. П. Драгоманова, 2011. – Вип. 10 (17). – С. 80-83.
- [9] Ф. С. Ільєсова, “Педагогічні аспекти навчання мови програмування Java”, *Комп'ютерно-орієнтовані системи навчання: Науковий часопис НПУ ім. Драгоманова*. Серія 2, Київ, Україна: Видавництво НПУ, 2012. – Випуск 8 (15). – С. 42-45.
- [10] Я. М. Глинський, та Л. Я. Палюшок, “Яку мову програмування вивчати у школі”, *Комп'ютер у школі та сім'ї*, №8, с. 9-15, 2013.

- [11] С. С. Жуковський, та О. В. Коротун, “Про перспективу введення мови програмування С++ в навчальний процес загальноосвітніх навчальних закладів”, *Комп’ютер у школі та сім’ї*, №1, с. 23-25, 2014.
- [12] *Державний стандарт базової і повної загальної середньої освіти* / Затверджено постановою Кабінету Міністрів України від 23 листопада 2011 р. № 1392.
- [13] Visual Studio 2015 System Requirements [Електронний ресурс]. Доступно: <https://www.visualstudio.com/en-us/productinfo/vs2015-sysrequirements-vs#vs>. Дата звернення: Берез.4, 2017.
- [14] SourceForge – Download, Develop and Publish Open Source Software [Електронний ресурс]. Доступно: <https://sourceforge.net/>. Дата звернення: Берез. 23, 2017.
- [15] DEV С++ Blog [Електронний ресурс]. Доступно: <http://orwelldevcpp.blogspot.com/>. Дата звернення: Лютий 27, 2017].
- [16] Bloodshed Software – Providing Free Software to the internet community [Електронний ресурс]. Доступно: <http://www.bloodshed.net/>. Дата звернення: Січень 29, 2017.
- [17] wxDev-C++ [Електронний ресурс]. Доступно: <http://wxdsn.sourceforge.net/>. Дата звернення: Берез.3, 2017.
- [18] Programming with wxDevC++ [Електронний ресурс]. Доступно: <http://wxdevcpp-book.sourceforge.net/>. Дата звернення: Лютий 16, 2017.
- [19] Code::Blocks [Електронний ресурс]. Доступно: <http://www.codeblocks.org/>. Дата звернення: Січень 22, 2017.
- [20] Среда разработки Code::Blocks – *Информатика (Язык С) – Фоксфорд, Учебник* [Электронный ресурс]. Доступно: <http://foxford.ru/wiki/informatika/sreda-razrabotki-code-blocks>. Дата обращения: Янв.31, 2017.
- [21] CodeLite – A Free, open source, cross platform C,C++,PHP and Node.js IDE [Електронний ресурс]. Доступно: <https://codelite.org/>. Дата звернення: Лютий 25, 2017.
- [22] MonoDevelop [Електронний ресурс]. Доступно: <http://www.monodevelop.com/>. Дата звернення: Берез. 2, 2017.
- [23] IntelliJ IDEA The Java IDE [Електронний ресурс]. Доступно: <https://www.jetbrains.com/idea/>. Дата звернення: Берез. 3, 2017.
- [24] В.Д. Руденко, *Вивчаємо Java у школі, Ч.1, Синтаксис мови*, Харків, Україна: Ранок, 2016.
- [25] Eclipse – The Eclipse Foundation open source community website [Електронний ресурс]. Доступно: <http://www.eclipse.org/>. Дата звернення: Берез. 1, 2017.
- [26] Т. С. Машнин, *Eclipse: разработка RCP-, Web-, Ajax и Android-приложений на Java*, СПб., Россия: БХВ-Петербург, 2013.
- [27] В. В. Степанцов, *Практическая работа в Eclipse 3.2 : Учебно-методическое пособие для вузов*, Воронеж, Россия: Издательско-полиграфический центр Воронежского государственного университета, 2007.
- [28] М. Ленди, С. Сиддикви, и Дж.Свишер, Borland JBuilder. *Руководство разработчика*, пер. с англ. Москва, Россия: Издательский дом «Вильямс», 2004.
- [29] Ч. Калверт, и М. Калверт, JBuilder. *Разработка профессиональных приложений*, пер. с англ., Киев, Украина: ООО «ТИД «ДС», 2004.
- [30] Заметки о выпуске IDE NetBeans 8.0.1 [Электронный ресурс]. Доступно: [https://netbeans.org/community/releases/80/relnotes\\_ru.html?print=yes](https://netbeans.org/community/releases/80/relnotes_ru.html?print=yes). Дата обращения: Февр. 24, 2017.

*Матеріал надійшов до редакції 12.03.2017 р.*

## СРЕДЫ ПРОГРАММИРОВАНИЯ КАК СРЕДСТВО ОБУЧЕНИЯ УЧАЩИХСЯ ОСНОВАМ ПРОГРАММИРОВАНИЯ

### **Базурин Виталий Николаевич**

кандидат педагогических наук, доцент кафедры профессионального образования и компьютерных технологий

Глуховский национальный педагогический университет имени Александра Довженко, г. Глухов, Украина

ORCID ID 0000-0002-6614-4889

*u-3700@ukr.net*

**Аннотация.** В статье раскрываются условия выбора среды программирования в качестве средства обучения учеников общеобразовательной школы программированию на современных языках. Определены основные условия, которые влияют на выбор среды программирования: технические характеристики компьютеров и системные требования среды программирования; наличие операционных систем и дополнительного программного обеспечения, необходимого для функционирования среды программирования; функционал среды программирования; интерфейс среды программирования; наличие документации на программную среду; наличие учебно-методического обеспечения; уровень компетентности учителя информатики. Проанализированы характеристики наиболее распространенных сред программирования на языках C/C++, C#, Java. Обосновано выбор среды программирования для изучения каждого из указанных языков программирования для обучения начинающих программистов и учеников, которые имеют навыки разработки программ.

**Ключевые слова:** язык программирования; среда программирования; интерфейс; функционал.

## PROGRAMMING ENVIRONMENTS AS A MEANS OF TEACHING PUPILS TO PROGRAMMING BASICS

**Vitalii M. Bazurin**

PhD (Pedagogy), Associate Professor of the Department of Professional Education and Computer Technologies  
Glukhiv national pedagogical university named by Oleksandr Dovzhenko, Glukhiv, Ukraine

ORCID ID 0000-0002-6614-4889

*u-3700@ukr.net*

**Abstract.** The article reveals the conditions for choosing the programming environment as a means of teaching students of the general education school to programming in modern languages. The main conditions that influence the choice of the programming environment are determined: technical characteristics of computers and system requirements of the programming environment; availability of operating systems and additional software required for the functioning of the programming environment; functional of the programming environment; the interface of the programming environment; availability of documentation for the software environment; availability of educational and methodological support; level of competence of the teacher of computer science. The characteristics of the most common programming environments in C / C ++, C #, Java are analyzed. The selection of the programming environment for studying each of the specified programming languages is substantiated for the training of beginning programmers and students who have programming skills.

**Keywords:** programming language; programming environment; interface; functional.

## REFERENCES (TRANSLATED AND TRANSLITERATED)

- [1] V.D. Rudenko, "Modern computer literacy and content issues of school informatics", *Ukrainskyi pedahohichnyi zhurnal*, #3, s.158-169, 2015. (in Ukrainian)
- [2] V.V. Lapinskyi, "The choice of the first programming language - Today Vision", *Kompiuter u shkoli ta simi*, # 1, s. 14–17, 2014. (in Ukrainian)
- [3] V.D. Shvets, "Application package EXCEL data processing laboratory works on physics", *Fizyka ta astronomiia v shkoli*, # 6, s. 50 – 53, 2003. (In Ukrainian)
- [4] N.V. Morze, *Methods of teaching computer science, teach. manual in 3 h.*, Kyiv, Ukraine: Navchalna knyha, 2004. (in Ukrainian)
- [5] O.M. Spirin, and P.H. Shevchuk, "Comparative analysis of software technologies operating system Windows 8 for learning the programming", *Information Technologies and Learning Tools*, # 2, s. 65-73, 2014. [online]. Available: <http://journal.iitta.gov.ua/index.php/itlt/article/view/997>. Accessed on: Jan.24, 2017. (in Ukrainian)
- [6] P.H. Shevchuk, "From Pascal to C#", *Kompiuter u shkoli ta simi*, #4, s.47-52, 2011. (in Ukrainian)
- [7] P.H. Shevchuk, "The choice of language and programming environment as a means of learning", Reporting Scientific Conference of the Institute of Information Technologies and Learning Pedagogical Sciences of Ukraine: Materials Conference.– Kyiv, 2010, s.30-31. (in Ukrainian)

- [8] P.H. Shevchuk, “Software-technical conditions of use C # language for teaching programming in secondary schools”, *Naukovyi chasopys Natsionalnoho pedahohichnoho universytetu imeni M. P. Drahomanova*. Serii 2, vyp. 10 (17), Kompiuterno-orientovani systemy navchannia, Kyiv, Ukraine: Vyd-vo NPU im. M.P. Drahomanova, 2011, s. 80-83. (in Ukrainian)
- [9] F.S. Iliasoza, “Pedagogical aspects of learning the programming language Java”, Kompiuterno-orientovani systemy navchannia: *Naukovyi chasopys NPU im. Drahomanova*. Serii 2. Vypusk 8 (15), Kyiv, Ukraine: Vydavnytstvo NPU, 2012, s. 42-45. (in Ukrainian)
- [10] Ya.M. Hlynskyi, and L.Ya. Paliushok “What programming language to learn in school”, *Kompiuter u shkoli ta simi*, #8, s. 9-15, 2013. (in Ukrainian)
- [11] S.S. Zhukovskiy, and O.V. Korotun, “Prospects for the introduction of C ++ programming language in the educational process of secondary schools”, *Kompiuter u shkoli ta simi*, #1, s.23-25, 2014. (in Ukrainian)
- [12] *State standard of universal middle education base and complete* / It is ratified by on of Cabinet of Ministers of Ukraine from November, 23, 2011 № 1392. (in Ukrainian)
- [13] Visual Studio 2015 System Requirements [online]. Available: <https://www.visualstudio.com/en-us/productinfo/vs2015-sysrequirements-vs#vs> . Accessed on: March 04, 2017.(in English)
- [14] SourceForge – Download, Develop and Publish Open Source Software [online]. Available: <https://sourceforge.net/> Accessed on: Feb.23, 2017. (in English)
- [15] DEV C++ Blog [online]. Available: <http://orwelldevcpp.blogspot.com/> Accessed on: Feb.27, 2017.(in English)
- [16] Bloodshed Software – Providing Free Software to the internet community [online]. Available: <http://www.bloodshed.net/> Accessed on: Jan.29, 2017.(in English).
- [17] wxDev-C++ [online]. Available: <http://wxdsgn.sourceforge.net/>. Accessed on: March 3, 2017.(in English).
- [18] Programming with wxDevC++ [online]. Available: <http://wxdevcpp-book.sourceforge.net/>. Accessed on: Febr.16, 2017.(in English).
- [19] Code::Blocks [online]. Available: <http://www.codeblocks.org/>. Accessed on: Jan.22, 2017.(in English).
- [20] Environment Development Code :: Blocks - Informatics (C Language) - Foksford. Tutorial – Foksford. Uchebnyk [online]. Available: <http://foxford.ru/wiki/informatika/sreda-razrabotki-code-blocks>. Accessed on: Jan.31, 2017. (in Russian)
- [21] CodeLite – A Free, open source, cross platform C,C++,PHP and Node.js IDE [online]. Available: <https://codelite.org/>. Accessed on: Febr.25, 2017.(in English).
- [22] MonoDevelop [online]. Available: <http://www.monodevelop.com/>. Accessed on: March 2, 2017.(in English).
- [23] IntelliJ IDEA The Java IDE [online]. Available: <https://www.jetbrains.com/idea/>. Accessed on: March 6, 2017] .(in English).
- [24] V.D. Rudenko, and O.O. Zhugastrov, Learning Java in school, Kharkiv, Ukraine: Ranok, 2016, Ch.1. Syntaksys movy. – 96 s. (in Ukrainian)
- [25] Eclipse – The Eclipse Foundation open source community website [online]. Available: <http://www.eclipse.org/>. Accessed on: March 1, 2017.(in English).
- [26] T.S. Mashnyn Eclipse: Development RCP-, Web-, Ajax and applications for Android-Java, SPb., Russia: BHV-Peterburh, 2013. (in Russian)
- [27] V.V. Stepantsov, *Practical work in Eclipse 3.2: Teaching-methodological manual for universities*, Voronezh, Russia: Yzdatelsko-polyhrafycheskyi tsentr Voronezhskoho hosudarstvennoho unyversyteta, 2007. (in Russian)
- [28] M. Landy, S. Siddiqui, and J. Swisher, Borland JBuilder. *MANUAL developer* : translate from English, Moscow, Russia: Yzdatelskyi dom «Vyliams», 2004. (in Russian)
- [29] Ch. Calvert, M. Calvert, JBuilder. *Development of Professional Applications* : translate from English, Kyiv, Ukraine: OOO «TYD «DS», 2004. (in Russian)
- [30] Notes issue of IDE NetBeans 8.0.1 [online]. – Available: [https://netbeans.org/community/releases/80/relnotes\\_ru.html?print=yes](https://netbeans.org/community/releases/80/relnotes_ru.html?print=yes). Accessed on: Febr.24, 2017. (in Russian)



This work is licensed under Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.