

ОДИН З ПІДХОДІВ ДО ЗАБЕЗПЕЧЕННЯ ВЗАЄМОДІЇ СЕРЕДОВИЩ VISUAL STUDIO і ECLIPSE

Розглядається проблема взаємодії різнорідних середовищ з розробки в них програм для забезпечення подальшої сумісної інтеграції цих програми з урахуванням умов їх побудови в різних середовищах. Розглянуті два середовища розробки: Visual Studio .Net та Eclipse. Показано способи перенесення проєктів програм, створених в одному з названих середовищ в інше, забезпечення їхньої взаємодії при об'єднанні та виконанні для отримання кінцевого результату.

Вступ

У зв'язку з появою різних розподілених середовищ (Sun IBM, Microsoft, CORBA, COM, JAVA і ін.) для побудови програм засобами діючих в них мовах програмування (МП), виникла проблема забезпечення взаємодії різномовних і різноплатформених програм між собою, коли вони переносяться в невідповідне для їх розроблення середовище.

Під *взаємодією* розуміється сумісність двох і більше об'єктів. Даний термін має спеціальний спектр використання в діяльності програмістів (наприклад, взаємодія програм і середовищ між собою тощо) [1, 2]. Здатність до взаємодії двох і більше програм або систем пов'язана з обміном інформацією і використанням її для організації обчислень. Для забезпечення локальної взаємодії програм у МП існує апарат зв'язку підпрограм і функцій через оператори звернення типу CALL, а для розподілених програм у різних середовищах є, наприклад, такі засоби – RPC, RMI, ORB (stub, skeleton), IContract тощо. Відповідно цих засобів зв'язки різномовних та різноплатформених програм виконує інтерфейс, який специфікується загально прийнятою мовою IDL (Interface Definition Language). На загальному рівні опис інтерфейсу з типами даних, що передаються, слугує для різних середовищ механізмом забезпечення взаємодії різномовних програм.

Кожне сучасне операційне середовище реалізує цей механізм по-різному. І тому, при переносі програм з одного середовища в інше вирішуються питання забезпечення їхньої інтероперабельності в новому середовищі.

На даний час використовуються деякі засоби забезпечення взаємодії різнорідних програм у Європейському проєкті Grid – інфраструктурі глобальних обчислень за різнорідними програмами із класу e-science. Одним з загальних способів підтримки взаємодії у цьому середовищі є конфігураційний файл, в якому міститься інформація про різні механізми взаємодії програм у середовищах, де вони будувалися [3, 4].

У даній роботі розглянуті два середовища розробки програм – Visual Studio та Eclipse, на які орієнтується в проєкті Grid і застосовуються в фундаментальному проєкті ПІ-1-07 ІПС НАН України [5]. На прикладі цих середовищ наведено новий спосіб забезпечення взаємодії середовищ і програм, що створені в них.

Особливості розроблення програм у заданих середовищах

Засобами цих середовищ можна розробляти програми та складати їх шляхом збирання із компонентів, специфікованих різними мовами програмування. Це має велике значення тому, що для написання більш-менш складного повноцінного програмного продукту в більшості випадках і використовуються специфічні засоби різних МП. Наряду з мовними засобами для опису програм у науковому проєкті [5], що базується на інтегрованому середовищі Eclipse, виникла ідея поширення спектру МП для використання різних їх можливостей при об'єднанні складних різномовних програмних систем, що кожне середовище надає окремо в межах своїх МП і підходів до їх зборки.

Для реалізації цієї ідеї було запропоновано декілька пар мобільних середовищ, зокрема, CORBA ↔ Eclipse [6] та MS.Net ↔ Eclipse. Тобто через додавання до системи Eclipse відкритого доступу і механізмів взаємодії цих середовищ підвищується могутність цієї системи.

У системі CORBA реалізовані такий набір МП: Ada, C++, C, Lisp, Smalltalk, Java, COBOL, Object Pascal, PL/1, Python. Крім того, допускається застосування й мов з середовища .NET. Зв'язок програм у цих МП здійснюється через проміжний прошарок за допомогою брокера ORB, який оброблює інтерфейс в IDL, а саме, Stub для клієнтських програм і Skeleton для серверних програм. Наявність опису брокера в МП високого рівня дозволяє перенести його реалізацію й в інші середовища для забезпечення взаємодії різно-рідних програм у цьому середовищі. Наприклад, Borland Enterprise Broker підтримує МП Java і C++. Існують також інструменти, які розширюють можливості брокера для компілювання стабов IDL-описів для МП в інших середовищах (наприклад, платформа MS.NET забезпечена пакетом IIOPNet).

Автором був реалізований зв'язок між Visual Studio (VS) та Eclipse для мов C#. і Java іншим підходом, ніж той, що був розглянутий для системи CORBA. Visual Studio надає широкі можливості для програмування чотирьох мов: C#, F#, C++ , Visual Basic. Це покривається потужними засобами програмування, які VS надає для роботи з базами даних, створення графічного інтерфейсу користувача, режимів тестування та налагодження програм, фреймворки для розробки web-програм та багато іншого [7–10]. З іншої сторони є середовище розробки Eclipse, це потужне безкоштовне середовище розробки програм, яке за допомогою механізму плагінів надає можливість для написання програм на багатьох МП (Java, C\C++, Python, PHP, C#, COBOL та інші) з використанням різних фреймворків (Spring, Zend, Symphony, Django, Twisted і багато інших). Ще однією перевагою Eclipse є те, що воно не прив'язана до якоїсь платформи чи ін-

струментальних засобів. Деяку програму можна збирати різними компіляторами, запускати на різних платформах (наприклад, програму на C# можна збирати та розгортати як на платформі Mono, так і на платформі .NET). Усі розроблені у цьому середовищі програми зберігаються у репозитарії програм і інтерфейсів.

Для побудови різномовних програм через Visual Studio можна скористатись засобами платформи .NET. У ній є CTS (Common Type System, загальна система типів), за допомогою якої програма, яка використовує метод чи клас із модуля, написаний на іншій мові, правильно інтерпретує типи цього методу чи класу. CLR (Common Language Runtime) дозволяє виконувати програми, написані на мовах, які підтримує платформа .NET.

Платформа Eclipse дає модель загального інструментального засобу, призначеного для об'єднання інструментів, які готові або розроблені у іншому середовищі через точки розширення і механізми plug-in-ів. Для додавання їх до базової платформи платформа Eclipse має структуру підсистем, які складаються із множини plug-in-ів за механізмами *Runtime* (рис. 1).

Середовище Eclipse не має ніяких обмежень для підключення деяких програм, різних інструментів та середовищ. Вони забезпечує різні засоби для побудови різномовних програм, що розробляються у середовищах CORBA, COM та інші.

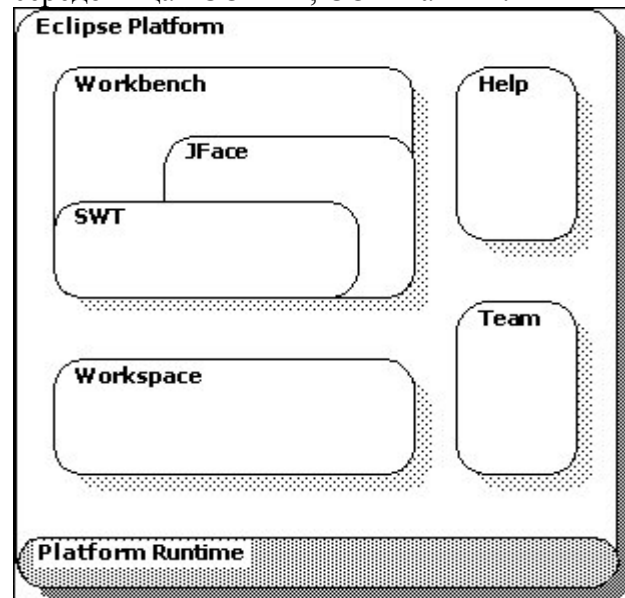


Рис. 1. Структура платформи Eclipse

Потрібно лише підключити відповідні плагіни у системі Eclipse. Все це значно розширює її можливості засобами забезпечення взаємодії різнорідних програм, що створені в інших середовищах [10, 11].

Зміст нового підходу для забезпечення взаємодії програм

Середовище Visual Studio має потужні засоби для створення програм на мові C#. Доцільно було б створювати проект саме в ньому, але є репозиторій, сховище програм, в якому програми розміщені в каталогах відповідно до їх тематики (фізика, хімія, математика і т. д.). Цей репозиторій є сховищем програм для середовища Eclipse. Було б дуже зручно вміти об'єднувати ці два середовища, щоб при створенні та розробці проекту використовувати властивості Visual Studio, а після реалізації програми помістити її в репозиторій Eclipse. Якщо потрібно буде внести зміни в дану програму, то можна просто внести зміни в програму із середовища Eclipse.

Для реалізації підходу потрібно мати два середовища: Visual Studio та Eclipse. Для підтримки в Eclipse мови C# потрібен плагін Emonic та програма NAnt для взаємодії Eclipse із визначеною платформою (Mono чи .NET). Встановивши ці програми можна створений проект програми із Visual Studio імпортувати в Eclipse та внести зміни в конфігураційний файл при її збірці.

Створюємо проект у Visual Studio. Для прикладу побудуємо графічний Windows-проект, який буде генерувати задану кількість білетів із заданої кількості питань та розподіляти для кожного білета за трьома рівнями складності питань (рис. 1).

Для переходу в Eclipse нас цікавлять вихідні файли програми, dll-бібліотки VS, від яких залежить ця програма та файли ресурсів (.resx), так як створено Windows-програму. Але для того, щоб можливо знову перейти в Visual Studio будемо імпортувати весь проект.

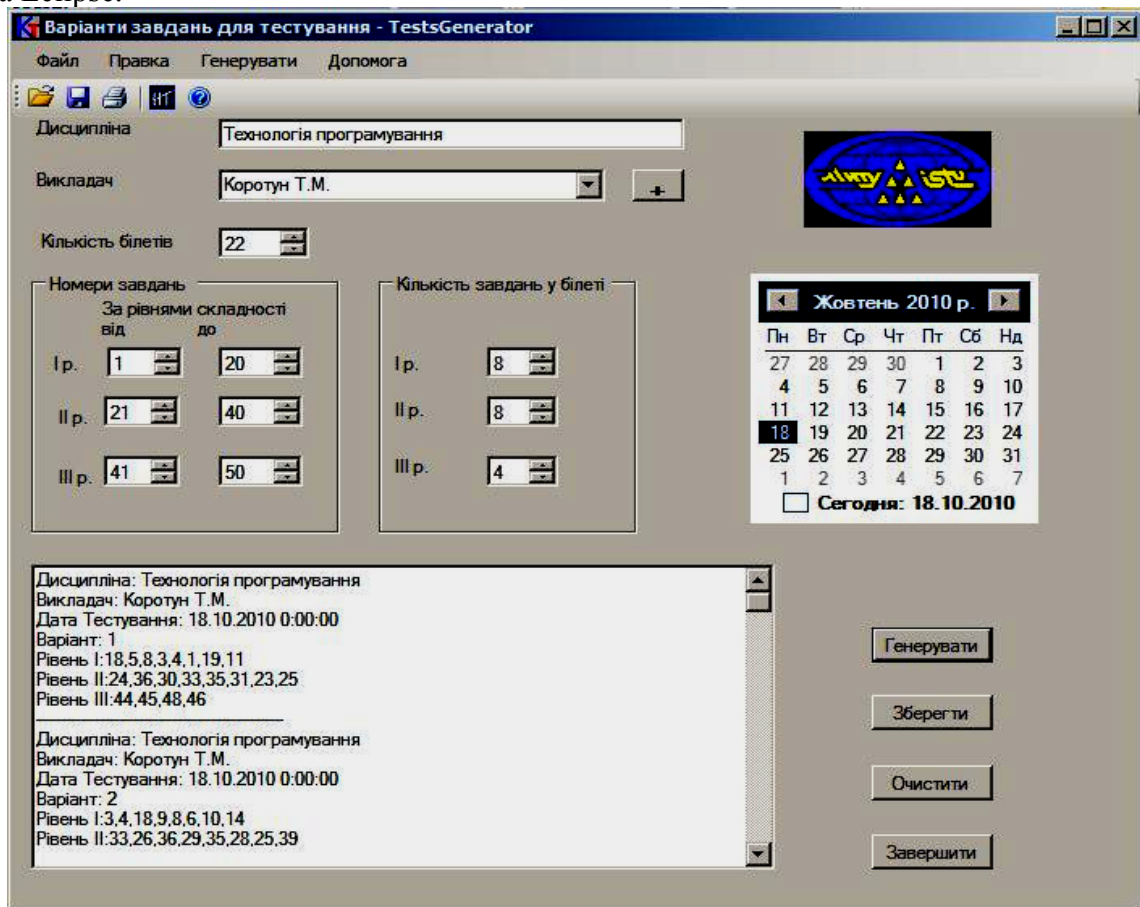


Рис. 2. Приклад тестової програми

Треба в Eclipse встановити плагін для мови C#. Для цього завантажується плагін emonic та NAnt. Розархівуємо архів з emonic та переносимо вміст папок архіву в аналогічні папки середовища Eclipse. Також необхідно розархівувати архів з NAnt. Створюємо в Eclipse пустий проект. Натиснувши правою кнопкою миші назву проекту, із контекстного меню вибираємо пункт Import -> File System та з файлової системи повністю імпортуємо вибраний проект. Відкривши текстовим редактором конфігураційний файл проекту (.build), змінюємо його вміст, вказуючи вихідні файли, бібліотеки та файли ресурсів.

Опис прикладу зміни конфігураційного файлу проекту:

```
<target name='SomeTargetName'>
    <csc debug='true' optimize='true'
output='${build}/SomeTargetName.exe'
target='exe' warninglevel='4'>
<sources>
    <include name='**/*.cs'/>
</sources>
```

```
<resources dynamicprefix="true">
    <include name='**/*.resx'/>
</resources>
<references>
    <include
name='System.dll'/><include
name='System.Data.dll'/>
    <include name='System.XML.dll'/>
</references></csc></target>
```

Для того, щоб перебудувати та запускати проект потрібно створити відповідні конфігурації. Для цього треба натиснути правою кнопкою миші кнопку на панелі інструментів Run as ... та з контекстного меню вибрати пункт External Configuration Tools Потім створити конфігурацію для збірки програми. Для цього в Location вказується шлях до програми NAnt, а в Working Directory шлях до папки проекту в репозиторії (рис. 3).

Для запуску програми потрібно створити відповідну конфігурацію. Для цього в розділ Location вказується цільовий exe-файл, а в Working Directory – шлях до папки проекту (рис. 4).

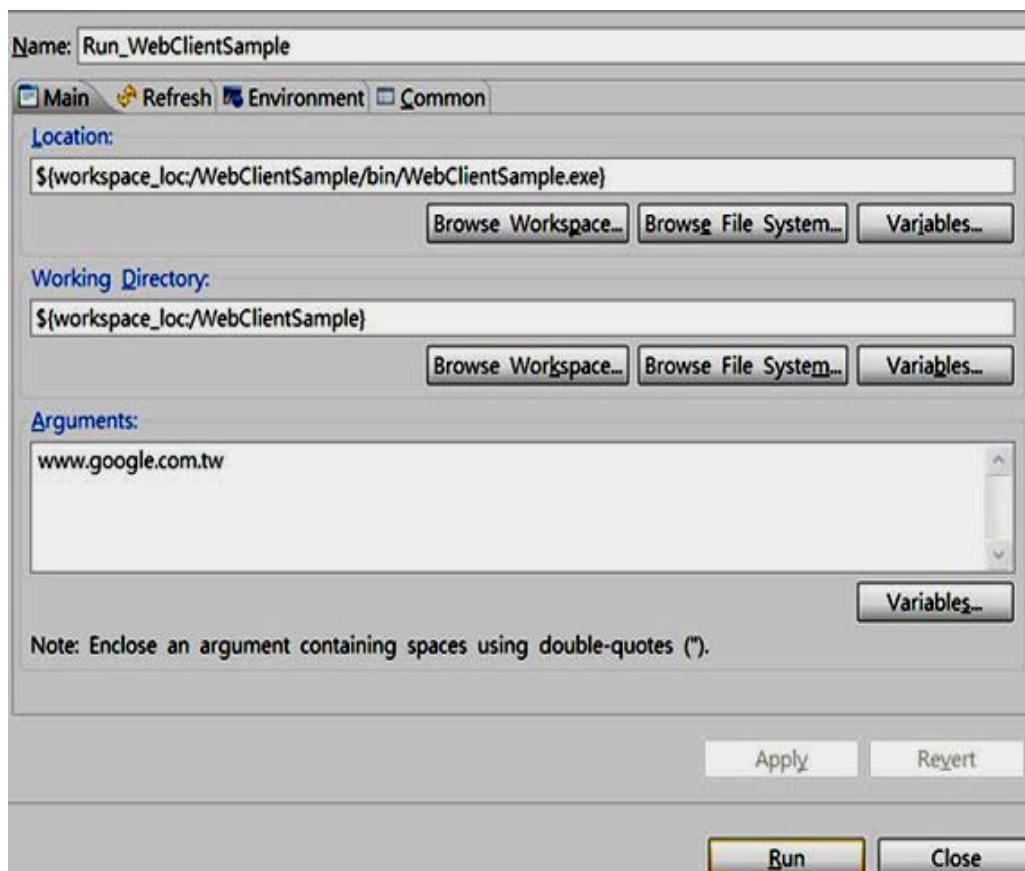


Рис. 3. Меню конфігурації для збірки

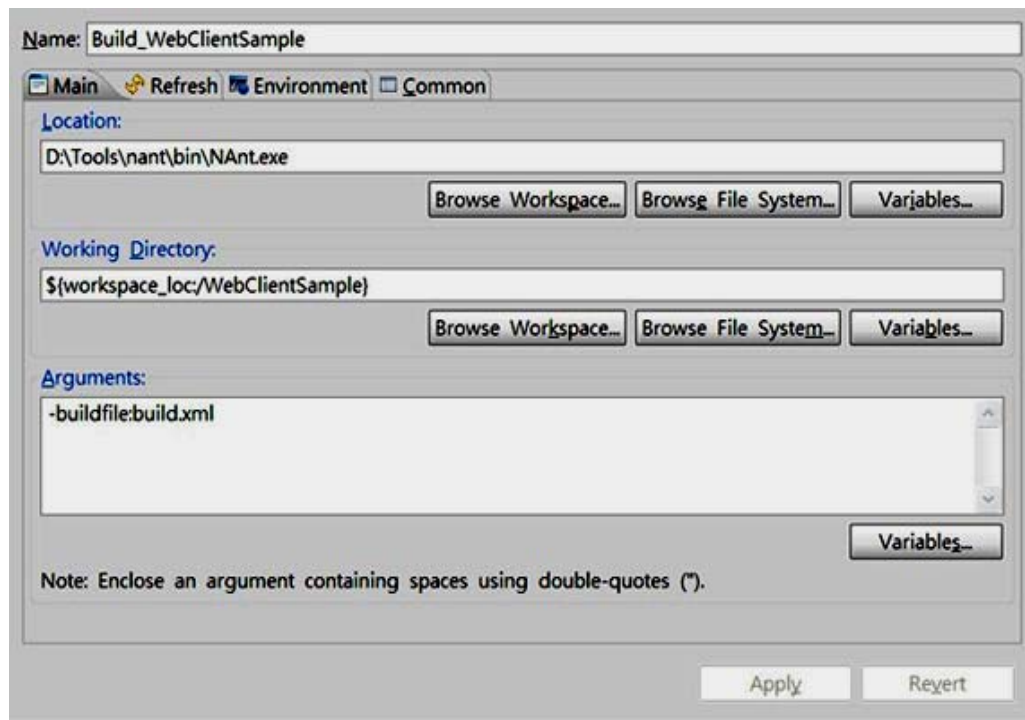


Рис. 4. Меню конфігурації для запуску проекту

Збірка різномовних програм у середовищі Visual Studio

Середовище Visual Studio має потужні засоби для програмування на мовах C#, F#, C++, Visual Basic. Завдяки платформі .NET та її CLR (Common Language Runtime) можна писати програми, які будуть складатися із компонентів, написаних на мовах, які підтримує платформа .NET. Вона є своєрідною віртуальною машиною, яка виконує свій специфічний байт-код. Вихідний код мови програмування компілюється в так званий керований код (managed code). Керований код представляє собою код на проміжній мові IL (Intermediate Language), який знаходиться посередині між високорівневою мовою (C#, F#, C++, Visual Basic) та мовою більш низького рівня (асемблером, машинним кодом). Під час виконання CLR на ходу компілює код IL в машинний код, застосовуючи для цього оперативну компіляцію (Just In Time – JIT). Компілятор JIT не компілює функцію чи метод при кожному його виклику: він робить це лише в перший раз, продукуючи при цьому машинний код, рідний за відношенням до платформи, на якій він виконується. Тобто під час виконання компілюється лише той

код, який необхідний. Такий спосіб компіляції відрізняється від інтерпретатора. Таким чином можна реалізувати збірку різномовних програм в Visual Studio за допомогою платформи .NET, звичайно, враховуючи те, що типи в методах та функціях мають відповідати загальній системі типів CTS.

Збірка різномовних програм у середовищі Eclipse

У Eclipse на відміну від Visual Studio немає таких обмежень у мовах та технологіях програмування. За допомогою механізму плагінів можна підключати засоби для створення програм в різних мовах програмування та фреймворках. Також є можливість створення власних плагінів. Це середовище є гнучким адаптивним інструментом, має високу якість. Для реалізації задачі об'єднання різномовних компонентів можна використовувати й інші технології, наприклад, COM, CORBA.

Для прикладу, можна показати як в Eclipse можна використовувати технологію CORBA. Завдяки механізму плагінів можна побудувати плагін для роботи з CORBA-об'єктами. Наприклад, можна взя-

Методи та засоби програмної інженерії

ти плагін Eclipse CORBA plugin (еср). Для його встановлення можна викачати архів із jar-файлами, або оновити Eclipse за допомогою команд Help -> Install New Software... Нажавши кнопку Add..., вказати <http://eclipsecorba.sf.net/update> та встановити плагін. Створювати новий IDL-інтерфейс можна за допомогою команд File -> New -> Other... та вибравши CORBA -> IDL file. Описавши новий інтерфейс можна скомпілювати стаб од-

ним із двох шляхів: використовуючи інтегрований компілятор OpenORB або вказати сторонній. У першому випадку потрібно зайти в Project -> Properties -> IDL Compilation та у вкладці Compilation Options зазначити Compile Stub (рис. 5), а у вкладці Compilers зазначити OpenORB (рис. 6). В іншому випадку потрібно зайти в External Tools Configuration та створити нову конфігурацію, вказавши в Location шлях до компілятора.

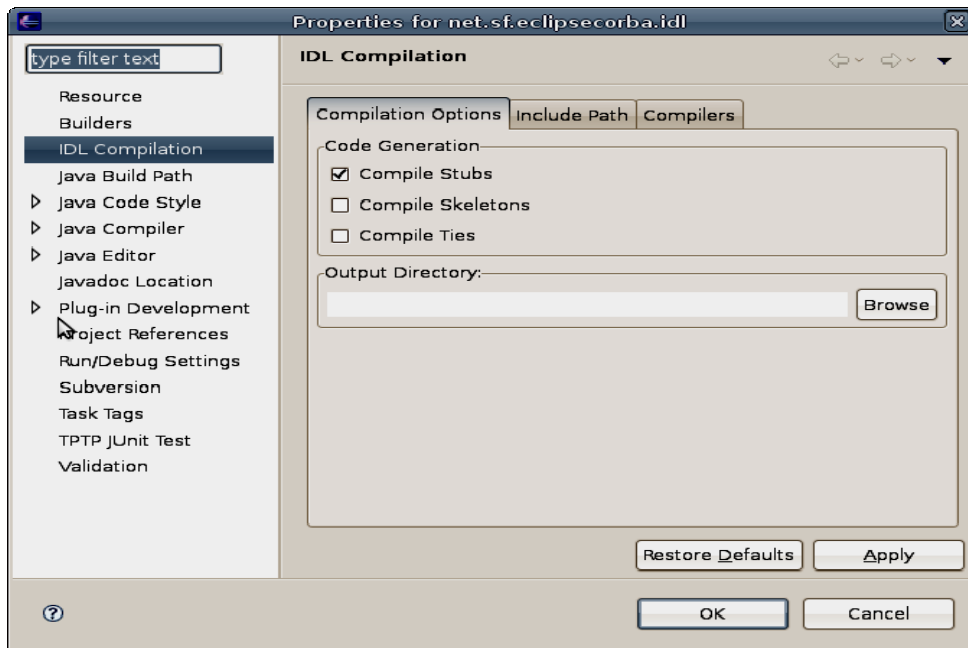


Рис. 5. Вибір опції для компіляції

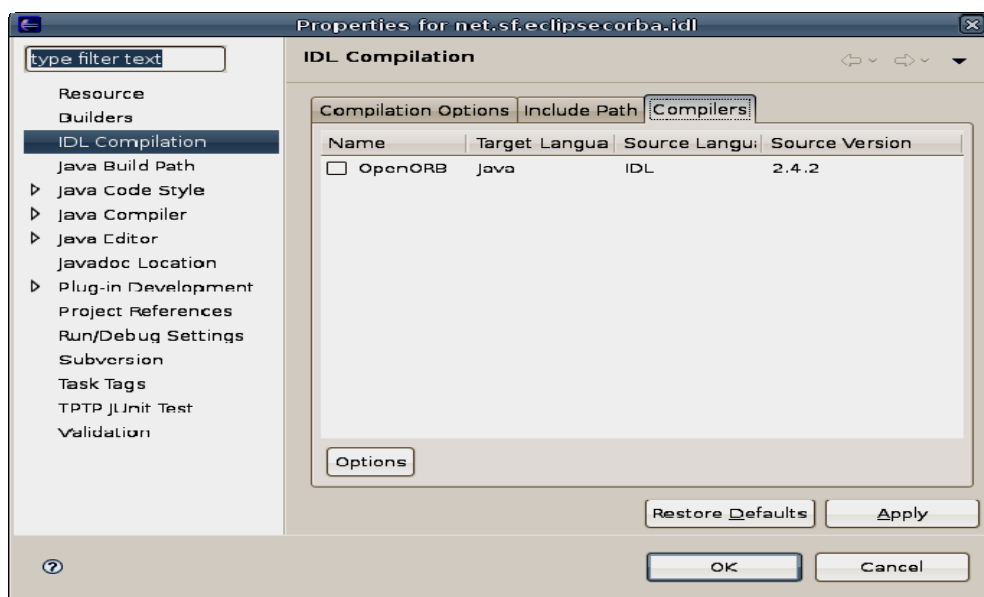


Рис. 6. Вибір компілятора

Висновки

У роботі дано аналіз підходів до забезпечення взаємодії різнорідних програм і сучасних середовищ. Наведені принципи взаємозв'язку різнорідних і різномовних програм через інтерфейс, в якому дається опис параметрів і типів даних, що необхідно передавати іншим програмам. Розглянуті варіанти операцій звернення програм між собою. Дано зміст і призначення нового способу реалізації зв'язків між середовищами розробки Visual Studio та Eclipse. Були розглянуті можливості цих двох середовищ, а також способи створення за допомогою них програм із компонентів, написаних на різних МП. Перелічені переваги та недоліки кожного із наведених підходів. Надані приклади, що підтверджують можливості та деякі додаткові переваги нового способу об'єднання двох середовищ MS.NET і Eclipse. Автор дякує за постановку задачі дослідження і допомогу в підготовці цієї статті професора К.М. Лавріщеву.

1. *Лаврищева Е.М.* Інтерфейс в програмуванні // Проблеми програмування. – 2007. – № 2. – С. 126–139.
2. *Лаврищева Е.М.* Проблема інтероперабельності разнородных объектов, компонентов и систем. Подходы к ее решению // Мат. 7 Міжнар. конф. 3 програмування “Укрпрог–2008”. – 2008. – № 2-3. – С. 28–41.
3. *Таковицкий О.* Технология Grid computing. – 2003. – С. 1–9.
4. *Андон П.І., Лаврищева К.М.* Розвиток фабрик програм в інформаційному світі // Вісник НАН України. – 2010. – № 10. – С. 15–41.
5. *Лаврищева Е.М.* Генерувальне програмування програмних систем і сімейств // Проблеми програмування. – 2009. – № 1. – С. 3–16.
6. *Островський А.И.* Подход к обеспечению взаимодействия программных сред JAVA и Ms.Net. // Проблеми програмування. – 2011. – № 2. – С. 40...
7. *Нейгел Кристиан, Ивѐн Билл, Глинн Джей, Уотсон Карл и Скиннер Морган.* С# 2008 и платформа .NET 3.5 для профессионалов. : Пер. с англ. – М. : ООО "И.Д. Вильямс", 2009. – 1392 с.
8. *Trey Nesh, Accelerated C# 2010.*: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2010. – 592 с.
9. <http://www.ibm.com/developerworks/library/os-ecl-vscd/> – міграція проектів на С# із Visual Studio в Eclipse.
10. <http://emonic.sourceforge.net/index.html> – головна сторінка плагіна Emonic.
11. <http://nant.sourceforge.net/> – головна сторінка засобу для збірки .NET програм NAnt.

Отримано 11.03.2011

Про автора:

Радецький Ігор Олегович,
студент 4 курсу КНУ ім. Тараса Шевченка,
факультет кібернетики.

Місце роботи автора:

Інститут програмних систем
НАН України,
інженер-конструктор I кат.
Тел.: 050 641 0472
e-mail: iradetskiy@mail.ru