

## Курсова робота з кросплатформного програмування (КП)

Навчальна дисципліна «Кросплатформне програмування» забезпечує набуття студентами знань та умінь для створення кросплатформних програмних продуктів на базі опанування відповідних технологій. Дисципліна відноситься до циклу дисциплін професійної та практичної підготовки і базується на вивченні таких дисциплін:

1. Алгоритмізація та програмування (АтаП);
2. Об'єктно-орієнтоване програмування (ООП);
3. Операційні системи (ОС).

**Кросплатформне програмування** – технологія створення і об'єднання в єдину систему **компонентів**, які розроблялись на різних програмних та апаратних платформах.

Поняття кросплатформності використовують на різних рівнях абстрагування :

Кросплатформними називають деякі сучасні мови програмування високого рівня.

Наприклад, С, С++, Delphi – кросплатформні мови рівня компілювання. Для цих мов є компілятори під різні платформи.

Java і С# – кросплатформні мови рівня виконання.

Їхні **exe**-файли можна запускати на різних платформах без попереднього перекомпілювання.

Сучасні кросплатформні системи створюють з використанням **компонентно-орієнтованого підходу**.

**Компонентне програмування** є подальшим розвитком об'єктно-орієнтованого програмування і концепції «повторного використання коду» (**reuse**). Компонентне програмування повинне забезпечити простішу і швидшу процедуру інсталювання прикладного програмного забезпечення (ПЗ), а також збільшити відсоток повторного використання коду, тобто підсилити основні переваги ООП.

Під **компонентою** розуміємо незалежний модуль програмного коду, призначений для повторного використання і розгортання. Компонента є структурною одиницею програмної системи, яка має чітко визначений інтерфейс. Компонента може бути незалежно поставлена, додана до складу деякої системи або видалена з неї, може включатися до складу систем інших постачальників. Таким чином, компонента – це виділена структурна одиниця розгортання з чітко визначеним інтерфейсом.

**Метою курсової роботи є**

- закріплення отриманих знань з програмування із використанням алгоритмічних мов Delphi, С++, С#, Java.
- застосування відомих алгоритмів розв'язання прикладних задач та розробка власних алгоритмів;
- застосування технології ООП для вирішення задачі, яку необхідно розв'язати у процесі виконання курсової роботи;
- набуття навичок розробки та формування відповідної супровідної документації для створеного ПЗ;

Процес виконання курсової роботи має підготувати студентів до подальших етапів навчальної (дипломна робота) та практичної діяльності. Студент повинен навчитися розв'язувати задачі, та створювати відповідне ПЗ, яке можна буде портувати в середовища інших ОС або на інші мікропроцесорні платформи (МП).

У ролі базових програмних середовищ доцільно використовувати такі середовища, як

- Microsoft Visual Studio 2012 або вищих версій;
  - Embarcadero RAD Developer Studio 10.1 та 10.2.
  - Qt Creator V 5.3 та ін.
  - Java Development Kit
- та інші популярні середовища.

Розробку ПЗ, при виконанні курсової роботи, можна проводити, використовуючи такі ОС, як Windows, Linux, MacOS, iOS і Android.

**Пояснювальна записка** до курсової роботи повинна налічувати такі частини:

- титульний лист;
- анотація українською та англійською мовами;
- зміст;
- вступ;
- 3 або 4 **Розділи**;
- висновки;
- список використаної літератури;
- додатки із текстом прокоментованих програмних модулів, які розробив автор роботи у процесі її виконання.

**Розділи** записки повинні містити:

Короткий порівняльний опис особливостей ОС та програмних середовищ, у яких розроблявся програмний проект.

Короткий порівняльний опис мов програмування, якими був реалізований програмний проект.

Опис та аналіз математичного забезпечення, яке було використано у роботі.

Опис архітектури та алгоритмів розробленого програмного проекту.

Опис тестових задач та результатів їхнього тестування.

**Увага!** Для отримання високої оцінки за курсову роботу необхідно реалізувати програмний проект або двома мовами програмування, або однією мовою, але у різних середовищах програмування або у різних операційних системах. У пояснювальній записці необхідно подати порівняння двох реалізацій мов програмування порівняння особливостей та можливостей, середовищ, які використовувались у роботі. Короткий перелік середовищ подано нижче.

- **Microsoft Visual Studio** – багатомовне інтегроване середовище розробки програмного забезпечення, яке дозволяє розробляти як консольні програми, так і програми з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також web-сайти, web-додатки, web-служби, як у native, так і в managed кодах із використанням платформи .NET.
- **Mono** – кросплатформний вільний відкритий варіант платформи .NET, який відповідає стандартам ECMA. Mono включає компілятор мови C# – mcs, середовище виконання .NET – mono (із підтримкою JIT) і mint (без підтримки JIT), зневаджувач, і ряд бібліотек, включаючи реалізацію ADO.NET і ASP.NET. У межах проекту також розробляються прив'язки для графічної системи GTK+ на платформі .NET. Середовище виконання mono може виконувати модулі, які написані мовами C#, F#, Visual Basic .NET, Java, Boo, Nemerle, Python, Forth, JavaScript, PHP і Object Pascal (за наявності компілятора у середовище .Net/Mono). Очікується також підтримка мов C, Ada 2005 і Eiffel. Ліцензія на компілятор та інші програми – GPL. Ліцензія на середовище виконання й інші бібліотеки – LGPL. Ліцензія на бібліотеки класів – X11 License. Патенти на C#/CLI належать Microsoft, і точаться суперечки щодо ризику і допустимості зростання залежності від Mono або C#.
- **Xamarin** – фреймворк для кросплатформної розробки мобільних додатків (iOS, Android, Windows Phone) із використанням мови C#. Фреймворк налічує декілька основних частин:
  - Xamarin.iOS – бібліотека класів для C#, яка надає розробнику доступ до iOS SDK;
  - Xamarin.Android – бібліотека класів для C#, яка надає розробнику доступ до Android SDK;
  - Компілятори для iOS і Android;
  - IDE Xamarin Studio;
  - Плагін для Visual Studio.
- **Embarcadero Delphi** – інтегроване середовище швидкої розробки програмного забезпечення, яке працює під Microsoft Windows і підтримує розробку програмного забезпечення для операційних систем Microsoft Windows x86 та x64, Linux, Mac OSx86, Apple iOS та Android мовою програмування Delphi (Object Pascal). Embarcadero Delphi входить до групи засобів розробки Embarcadero RAD Studio.
- Середовище **Embarcadero C++ Builder 2010**, яке функціонує у складі Embarcadero RAD Studio. Має графічний інтерфейс, який є сумісний із середовищем Delphi.
- **Qt** – кросплатформний інструментарій розробки ПЗ мовою програмування C++. Надає можливість виконувати, написане за його допомогою ПЗ, на більшості сучасних ОС, просто компілюючи текст програми для кожної ОС без зміни вихідного коду. Містить всі основні класи, які можуть бути потрібні для розробки прикладного ПЗ, починаючи з елементів графічного інтерфейсу і закінчуючи класами для роботи з мережею, базами даних, OpenGL, SVG і XML. Qt також може бути використаним в багатьох інших мовах програмування таких, як Ada (QtAda), C# (Qyoto/Kimono), Java (Qt Jambi), Qt Jambi, Node.js, Pascal, Perl, PHP (PHP-Qt), Ruby (QtRuby), та Python (PyQt, PySide).

Текст цього документу підготував

доц. Іван Хвищун 15.03.2018 року.

Доповнення Тексту зроблено 02.05 2018 року. (Вимоги до курсової роботи із кросплатформного програмування автор неодноразово повідомляв студентам 2-го курсу на лекціях із КП).