

магістра спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. Дніпро : ПДАБА, 2020. 11 с.

4. Шатов С. В. Визначення параметрів уламків зруйнованих споруд та елементів будівель, які реконструюються. *Вісник Придніпровської державної академії будівництва та архітектури*. 2011. № 3. С. 8–14.

УДК 004.4

ВИВЧЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНИХ МОВ ПРОГРАМУВАННЯ

Автор – Гордій Чорний¹, студ. гр. КН-21

Науковий керівник – ст. викл. каф. комп'ютерних наук, інформаційних технологій та прикладної математики Євген Плахтій²

¹gordeyblack@gmail.com, ²plakhtii.ev@gmail.com

Придніпровська державна академія будівництва та архітектури

Об'єктно-орієнтоване програмування (ООП) зробило революцію в способі розробки програмних додатків, надавши повний набір концепцій і методології програмування, які дозволяють ефективно і масштабовано розробляти код, що зробило його фундаментальним компонентом сучасних мов програмування.

ООП – це парадигма програмування, яка наголошує на використанні об'єктів і класів для представлення та маніпулювання даними й функціональністю. ООП стало фундаментальним компонентом сучасних мов програмування, оскільки дозволяє розробляти ефективний і масштабований код, який можна легко підтримувати і розширювати. У цій роботі ми дослідимо історію та еволюцію ООП у мовах програмування, включаючи Java, C++ та Python. Проведемо оцінку їх сильних та слабких сторін з точки зору продуктивності, супроводжуваності та гнучкості, його ключових концепцій та методології, а також вплив, який вони мали на розробку програмних додатків.

Java – це високорівнева, об'єктно-орієнтована мова, що базується на класах і широко використовується для розробки корпоративних, мобільних і веб-додатків. Однією з головних переваг Java є її незалежність від платформи, що дозволяє їй працювати на будь-якій операційній системі, яка має віртуальну машину Java (JVM). Крім того, Java має потужний набір бібліотек та інструментів, які полегшують розробку складних додатків. Однак Java може бути повільною в компіляції та виконанні, що може вплинути на її продуктивність у певних сценаріях. Також ООП-функції Java включають інкапсуляцію,

успадкування та поліморфізм. Інкапсуляція гарантує, що дані приховані від зовнішнього доступу та можуть бути доступні лише через визначені методи. Спадкування дозволяє класам успадковувати атрибути та методи від батьківського класу, а поліморфізм дозволяє використовувати єдиний інтерфейс для представлення різних типів об'єктів [1].

C++ – це низькорівнева мова програмування, яка широко використовується для розробки операційних систем, системного програмного забезпечення та ігрових рушіїв. C++ підтримує всі можливості ООП, включаючи інкапсуляцію, успадкування, поліморфізм та абстрагування. C++ також надає розширені можливості, такі як шаблони, перевантаження операторів та віртуальні функції, що робить її потужним інструментом для розробки складних програмних систем. Відома своєю швидкістю та ефективністю, що робить її популярним вибором для додатків, які вимагають продуктивності в реальному часі. Однак ця мова може бути складною у вивченні та використанні, а її синтаксис може бути складним і схильним до помилок [2].

Python – це інтерпретована, динамічно типізована та об'єктно-орієнтована мова, яка широко використовується для аналізу даних, машинного навчання та веб-розробки. Python має простий та інтуїтивно зрозумілий синтаксис, що робить її легкою у вивченні та використанні, а також має велику та активну спільноту розробників, які роблять свій внесок у велику колекцію бібліотек та інструментів. Підтримує декілька функцій ООП, включаючи інкапсуляцію, успадкування та поліморфізм. Використовує динамічну типізацію, що дозволяє змінним змінювати тип під час виконання програми, на відміну від Java та C++, які використовують статичну типізацію. Python може бути повільною порівняно з такими скомпільованими мовами, як C++ та Java, що може обмежувати її використання у певних критичних до продуктивності додатках [3].

C# – це сучасна об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Microsoft. Вона була розроблена, щоб бути простою, ефективною та легкою у вивченні. C# широко використовується для розробки десктопних і веб-додатків, ігор та мобільних додатків. Деякі з переваг використання C# для ООП включають безумовно легкість у вивченні C#, бо він має простий синтаксис, який легко вивчити особливо для розробників з досвідом роботи з Java або C++. Надійність C# – це сильно типізована мова, яка забезпечує безпеку типів, запобігаючи поширеним помилкам, таким як виключення за нульовим посиланням. Ефективність C# компілюється в машинний код, що робить її швидшою та ефективнішою, ніж інтерпретовані мови, такі як Python. Інтеграція з фреймворком .NET C#

легко інтегрується з фреймворком .NET, надаючи розробникам широкий спектр бібліотек та інструментів для створення програмних додатків [4].

Підводячи підсумки, зауважимо, що з точки зору продуктивності C++ зазвичай вважається найшвидшою з трьох мов, за нею йдуть C#, Java і Python. Однак різниця у продуктивності між цими чотирма мовами може варіюватися залежно від конкретного додатку та апаратного забезпечення, на якому він працює. З точки зору зручності супроводу, Python часто вважається найлегшою в підтримці завдяки своєму простому і зрозумілому синтаксису, за нею йдуть Java і C++. Однак зручність супроводу може також залежати від складності програми та рівня кваліфікації команди розробників. З точки зору гнучкості, Python часто вважається найбільш гнучким завдяки динамічній типізації та здатності підтримувати як функціональні, так і об'єктно-орієнтовані стилі програмування. C# – це потужна мова програмування, яка пропонує низку переваг для ООП. Вона ефективна, проста у вивченні та добре інтегрується з платформою .NET.

Отже, вибір мови ООП залежить від кількох факторів, зокрема від конкретних вимог програми, рівня кваліфікації команди розробників та апаратного забезпечення на якому буде працювати програма. ООП стало невід'ємним компонентом сучасних мов програмування, надаючи програмістам потужний набір інструментів для розробки ефективних і масштабованих програмних додатків. Розуміючи історію, ключові концепції та вплив ООП на розробку програмного забезпечення, ми можемо оцінити його важливість і продовжувати розвивати його методологію та практику, щоб задовольнити потреби сучасного технологічного ландшафту, який постійно змінюється. C#, Java, C++ і Python – популярні мови ООП зі своїми сильними і слабкими сторонами, і кожна з них може бути хорошим вибором в залежності від конкретних потреб проекту. Зрештою, ключ до успіху в ООП полягає у виборі правильної мови для роботи та використанні найкращих практик в інженерії та дизайні програмного забезпечення.

Список використаних джерел

1. Bates B., Sierra K. Head First Java Second Edition, 2021. 720 p.
2. Lippman S. B., Lajoie J., Moo B. E. C++ Primer. Pearson Education, 2005. 969 p.
3. Barry P. Head First Python. O'Reilly Media, 2011. 494 p.
4. Chan J. Learn Python in one day and learn it well : Python for beginners with hands-on project : the only book you need to start coding in Python immediately. CreateSpace Independent Publishing, 2015.