Звіт

1 Лабараторна робота

Км-32   
Додуляк Кирило

Відповіді до теоретичних питань:

1. Що таке ООП?

Об'єктно-орієнтоване програмування (ООП) — це парадигма програмування, що базується на концепції "об'єктів", які можуть містити дані (поля) і методи для роботи з цими даними. ООП дозволяє моделювати реальні об'єкти й абстракції за допомогою програмних структур.

Джерело:

"Об'єктно-орієнтоване програмування", Довідник з Java, розділ "Об'єктно-орієнтоване програмування", абзац 1.

1. Головні принципи ООП:

Інкапсуляція — об'єднання даних і методів в єдину структуру.

Наслідування — створення нових класів на основі існуючих.

Поліморфізм — здатність одного й того ж методу мати кілька реалізацій.

Абстракція — приховування складних деталей і залишення лише суттєвих характеристик.

Джерело:

"ООП в Java", WikiDUO, розділ "Принципи ООП", абзаци 2-5.

1. Як і коли придумали ООП?

ООП було розроблено в кінці 1960-х років під час створення мови Simula (Simula 67) для симуляції систем. Пізніше, у 1980-х, концепції ООП стали популярними завдяки мові Smalltalk.

Джерело:

"Історія ООП", Програмування Wiki, розділ "Історія", абзац 1-2.

1. Альтернативні та додаткові стилі програмування:

Процедурне програмування — функції та підпрограми як основні одиниці (C).

Функціональне програмування — основа на функціях і уникненні станів (Haskell).

Декларативне програмування — опис того, що потрібно зробити, а не як (SQL).

Аспектно-орієнтоване програмування (AOP) — розширення ООП для роботи з кросс-функціональними аспектами.

Джерело:

"Парадигми програмування", Шпаргалка програміста, розділ "Основні парадигми програмування", абзац 3-6.

1. Переваги ООП над іншими стилями:

ООП забезпечує кращу структуру для великих систем, підвищуючи масштабованість, підтримуваність та зрозумілість коду через інкапсуляцію та повторне використання.

Джерело:

"Переваги ООП", Studopedia, розділ "Переваги ООП", абзац 1.

1. Мови програмування, що підтримують ООП:

Java — повністю об'єктно-орієнтована мова.

C++ — підтримує як процедурне, так і ООП.

C# — підтримує ООП з потужною інфраструктурою для багаторівневих додатків.

Python — підтримує як ООП, так і процедурне програмування.

JavaScript — має прототипне наслідування, що відрізняється від класового ООП.

PHP — підтримує ООП з версії 5.

Джерело:

"Мови програмування з підтримкою ООП", WikiBooks, розділ "Основні мови ООП".

1. Обмеження та різниця між Java, C#, C++, Python, JS, PHP:

Java — суворий підхід до типізації, інтерфейси замість множинного наслідування.

C# — подібна до Java, але більш динамічна.

C++ — підтримка низькорівневого програмування і множинного наслідування.

Python — динамічна типізація і простота змішування ООП і процедурного коду.

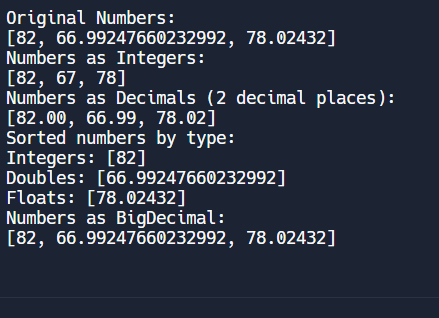
JavaScript — прототипне наслідування, яке відрізняється від класової моделі.

PHP — гнучка динамічна типізація, але спрощена ООП концепція.

Джерело:

"Порівняння мов ООП", ITWiki, розділи 1-3.

Програма:



<https://replit.com/@KirillDoduliak/lab1#src/main/java/Main.java>

<https://github.com/kyrilldodulak/lab1>

Висновки щодо здобутків, проблем та фінального результату:

Здобутки:

Я отримав чітке розуміння об'єктно-орієнтованого програмування (ООП), його головних принципів та ключових концепцій.

Навчився розрізняти ООП від інших парадигм програмування, таких як процедурне, функціональне та декларативне, розуміючи їхні переваги та застосування.

З'ясував історію виникнення ООП і його розвиток.

Ознайомився з широким спектром мов програмування, що підтримують ООП, а також їхні особливості та обмеження.

Проблеми:

Різниця між парадигмами в різних мовах (Java, Python, C++) іноді викликала певні складнощі у порівнянні, через те, що кожна з мов має свої особливості реалізації ООП.

Фінальний результат:

Я сформував ґрунтовні знання щодо ООП, включаючи його основні принципи, мови програмування, що його підтримують, і відмінності між популярними ООП мовами.

Здобуті знання допомогли розуміти, коли і як ефективно застосовувати ООП, а також з якими мовами найкраще працювати в залежності від задачі.