

Лабораторна робота № 4 з дисципліни «Програмування на мові Java»

Тема 4. «Розробка програм в середовищі INTELLIJ IDEA для дослідження концепції математичного моделювання та об'єктно-орієнтованого програмування»

Мета роботи: набуття практичних навичок під час створення програмних проєктів на мові Java.

Завдання на лабораторну роботу

1. Написати програму мовою Java. Варіант обрати за списком групи. Представити виконання в IntelliJ IDEA. Продемонструвати детальні скріни виконання коду програми з поясненням. Обов'язково – наявність висновків.
2. Написати програму мовою Java. Варіант обрати за списком групи. Представити виконання в IntelliJ IDEA. Продемонструвати детальні скріни виконання коду програми з поясненням. Обов'язково – наявність висновків.
3. Написати програму мовою Java. Варіант обрати за списком групи. Представити виконання в IntelliJ IDEA. Продемонструвати детальні скріни виконання коду програми з поясненням. Обов'язково – наявність висновків.
4. Зробити звіт з лабораторної роботи та вчасно надіслати викладачу на перевірку (дедлайн для надсилання звітів по Лаб_4 – до 27.11.2023 року до 23:59).

Частина 1. Основи концепції числових методів та математичного моделювання в Java

Завдання 1

Варіант 1	Варіант 2
(для всіх студентів, хто в списку йде за непарним номером (1,3,5,7 і т д))	(для всіх студентів, хто в списку йде за парним номером (2,4,6,8 і т д))
$A = 1 + x + \frac{x^2}{3} + \frac{x^3}{6} + \frac{x^5}{9};$ $B = x(\sin x^2 + \cos^2 y);$ $x = 0.2, y = 0.01$	$C = x^3 t g^3 (x^2 + y)^3 + \frac{z}{x + y};$ $x = 0.2, y = 2.5, z = 11$
Написати програму на Java, яка обчислює значення А та В за заданими формулами. Вимоги: -використати метод класу Math для обчислень ступенів, тригонометричних функцій тощо; -перевірити результати, шляхом виконання тестування з різними значеннями х та у для перевірки коректності розрахунків та виведення результатів.	Написати програму на Java, яка обчислює значення С за заданою формулою. Вимоги: -використати метод класу Math для обчислень ступенів, тригонометричних функцій тощо; -перевірити результати, шляхом виконання тестування з різними значеннями х, у та z для перевірки коректності розрахунків та виведення результатів.

Завдання 2

Варіант 1	Варіант 2
(для всіх студентів, хто в списку йде за непарним номером (1,3,5,7 і т д))	(для всіх студентів, хто в списку йде за парним номером (2,4,6,8 і т д))
$D = \frac{3 \cos\left(x - \frac{\pi}{2}\right)}{1 + \sin^2 y}; F = 1 + \frac{z^3}{x + 7};$ $x = 1.1; y = -1.2; z = 4.4$	$L = \frac{x^2(x + 2)}{z - \sin^2(x + y)}; N = \sqrt{\frac{x}{y}} + \cos^2(x + z);$ $x = 0.1, y = 0.01, z = 0.7$
<p>Написати програму на Java, яка обчислює значення D та F за заданими формулами.</p> <p>Вимоги:</p> <ul style="list-style-type: none"> -використати метод класу Math для обчислень ступенів, тригонометричних функцій тощо; -перевірити результати, шляхом виконання тестування з різними значеннями x, y та z для перевірки коректності розрахунків та виведення результатів. 	<p>Написати програму на Java, яка обчислює значення L та N за заданими формулами.</p> <p>Вимоги:</p> <ul style="list-style-type: none"> -використати метод класу Math для обчислень ступенів, тригонометричних функцій тощо; -перевірити результати, шляхом виконання тестування з різними значеннями x, y та z для перевірки коректності розрахунків та виведення результатів.

Частина 2. Концепція об'єктно-орієнтованого програмування

Завдання 3

Варіант 1	Варіант 2
(для всіх студентів, хто в списку йде за непарним номером (1,3,5,7 і т д))	(для всіх студентів, хто в списку йде за парним номером (2,4,6,8 і т д))
<p>Написати програму на Java, в якій буде створено ієрархію класів для геометричних фігур. Наприклад, базовий клас «Фігура» і підкласи «Коло», «Прямокутник», «Трикутник». Кожен клас повинен включати методи для обчислення площі/периметру фігури. Використати наслідування для спрощення коду.</p>	<p>Написати програму на Java, в якій буде створено ієрархію класів для різних типів транспортних засобів. Наприклад, базовий клас «Транспорт» і підкласи «Автомобіль», «Велосипед», «Мотоцикл». Кожен клас повинен включати методи для обчислення максимальної швидкості (або споживання палива). Використати наслідування для спрощення коду.</p>
<p>Рекомендації:</p> <ul style="list-style-type: none"> -клас «Фігура» може бути абстрактним, та може містити абстрактні методи: calculateArea()/calculatePerimeter(). -кожен підклас («Коло», «Прямокутник», «Трикутник») повинен реалізувати абстрактні методи calculateArea() та calculatePerimeter() згідно з їхніми геометричними формулами; - кожен підклас повинен мати властивості, що визначають фігуру (наприклад, радіус для кола, довжину та ширину для прямокутника, сторони для трикутника). 	<p>Рекомендації:</p> <ul style="list-style-type: none"> - клас «Транспорт» може бути абстрактним, та може містити атрибути, такі як швидкість, вага, колір тощо. - може містити конструктор, геттери та сеттери для атрибутів; - кожен підклас («Автомобіль», «Велосипед», «Мотоцикл») повинен реалізувати абстрактні методи та можуть додавати додаткові атрибути та методи, специфічні для кожного типу транспортного засобу; - для демонстрації створення об'єктів кожного з підкласів та виклику їхніх методів можна створити клас TransportTest;

<ul style="list-style-type: none"> - кожен підклас може містити конструктори для ініціалізації цих властивостей. - для демонстрації створення об'єктів кожного з підкласів та виклику їхніх методів можна створити клас GeometryTest; - необхідно додати перевірку вхідних даних у конструкторах підкласів, щоб запобігти створенню фігур з некоректними параметрами; - обов'язково виведення на консоль результатів обчислень площі та периметру для кожної фігури. 	<ul style="list-style-type: none"> - обов'язково виведення на консоль інформації про кожен транспортний засіб.
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------

ЗАВДАННЯ 4*

(завдання додаткове, для бажаних отримати додаткові бали (оцінюється в 3 бали). Вимоги щодо оформлення такі ж як і для інших завдань)

Написати програму на Java, в якій буде створено три класи: «Book», «Library», та «Librarian». Кожен клас представляє окремий аспект системи управління бібліотекою. Клас «Book» представляє книги з назвою та автором, клас «Library» представляє бібліотеку зі списком книг, а клас «Librarian» представляє бібліотекаря, який відповідає за управління книгами.

Код повинен представити структуру для моделювання бібліотеки. Основне завдання коду - додавання книг до бібліотеки, зберігання їх та відображення списку книг на консолі. Задача полягає в створенні програми для керування книгами у бібліотеці та відображенні інформації про них.

Рекомендації:

1)Клас «Book»:

- повинен містити принаймні два атрибути: title (назва) та author (автор);
- потрібно надати геттери та сеттери для управління зазначеними атрибутами;
- можна реалізувати метод toString() для виведення інформації про книги.

2)Клас «Library»:

- повинен містити колекцію книг, наприклад, використовуючи ArrayList<Book>.
- методи можуть включати: addBook, removeBook, findBookByTitle, listAllBooks, тощо.

- можна реалізувати додатково пошук книг за автором або жанром.

3)Клас «Librarian»:

- може включати методи для взаємодії з «Library», наприклад, manageBookAddition(), manageBookRemoval().
- може містити методи для взаємодії з користувачами, такі як checkOutBook()/returnBook().

При виконанні завдання, студент повинен використати підготовлений список з 5-7 реальних книг з назвами та авторами для заповнення бібліотеки.

При демонстрації взаємодії класів, студент повинен підготувати приклад, який продемонструє, як об'єкти класів можуть взаємодіяти між собою, наприклад, як Librarian викликає методи Library для додавання або видалення книг.

ВИМОГИ ДО ОФОРМЛЕННЯ ЗВІТУ З ЛАБОРАТОРНОЇ РОБОТИ

Звіт оформлюється студентом в редакторі MS Word:

- весь шрифт тексту звіту лише Times New Roman, розмір - 14, міжрядковий інтервал – 1, абзац – 1,25.
- поля звіту: ліве - 25 мм, верхнє та нижнє - 20 мм, праве 10 мм;
- нумерація сторінок (для титульного – особливий, тобто нумерація не вказується, однак враховується) – по центру нижнього поля;
- титульний лист оформлюється згідно «Додаток 1»;
- завдання на лабораторну роботу разом із інформацією щодо варіанту студента обов'язково вказується у звіті;
- текстова частина звіту оформлюється лише в текстовому форматі (заміна текстової частини звіту на відповідні скріншоти – не дозволяється);
- студентом надається детальний опис послідовності виконання для кожного завдання;
- скріншоти етапів виконання завдань та результатів повинні бути розбірливі та змістовні (оформлюються згідно «Додаток 2»);
- у звіті обов'язково вказується код програми (лістинг коду на білому фоні);
- обов'язково наявність висновків до кожного завдання.

УВАГА!

1. Звіт необхідно надіслати викладачеві до **27.11.2023 року (до 23:59)**.
2. Назва файлу звіту має бути чітко оформлена, за зразком: **ЛР4_ТР-11_Петренко (doc або docx)**
3. Відправляти звіти потрібно на пошту: nadezhdadovzhenko@gmail.com
4. Тема листа має бути такою: **ЛР4_ТР-11_Петренко**

Вимоги щодо захисту Звіту з Лабораторної роботи

Лабораторні роботи, які вчасно здано (надіслано) викладачеві, підлягають перевірці та допускаються до захисту.

Під час захисту – студент обов'язково вмикає камеру та мікрофон (всі лабораторні роботи записуються викладачем).

На захисті студент: демонструє виконання кожного із завдань в інтегрованому середовищі IntelliJ IDEA (або окремі завдання – за запитом від викладача); відповідає на 2-3 запитання (кожне з яких оцінюється певною сумою балів).

Викладач оцінює демонстрацію виконаних завдань, відповідність вимогам, якість підготовки звіту та правильність відповідей на запитання. Запитання на захисті ставляться виключно в рамках викладених лекцій та контексту виконаних завдань.

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

**Навчально-науковий інститут атомної і теплової енергетики
Кафедра цифрових технологій в енергетиці**

ЗВІТ

з лабораторної роботи № 4

з дисципліни «Програмування на мові Java»

**ТЕМА 4. «РОЗРОБКА ПРОГРАМ В СЕРЕДОВИЩІ INTELLIJ IDEA ДЛЯ
ДОСЛІДЖЕННЯ КОНЦЕПЦІЇ МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА
ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ»**

Варіант № ____

Виконав:
Студент групи ТР-11
Іваненко М.П.

Дата здачі _____

Приклад оформлення скріншотів

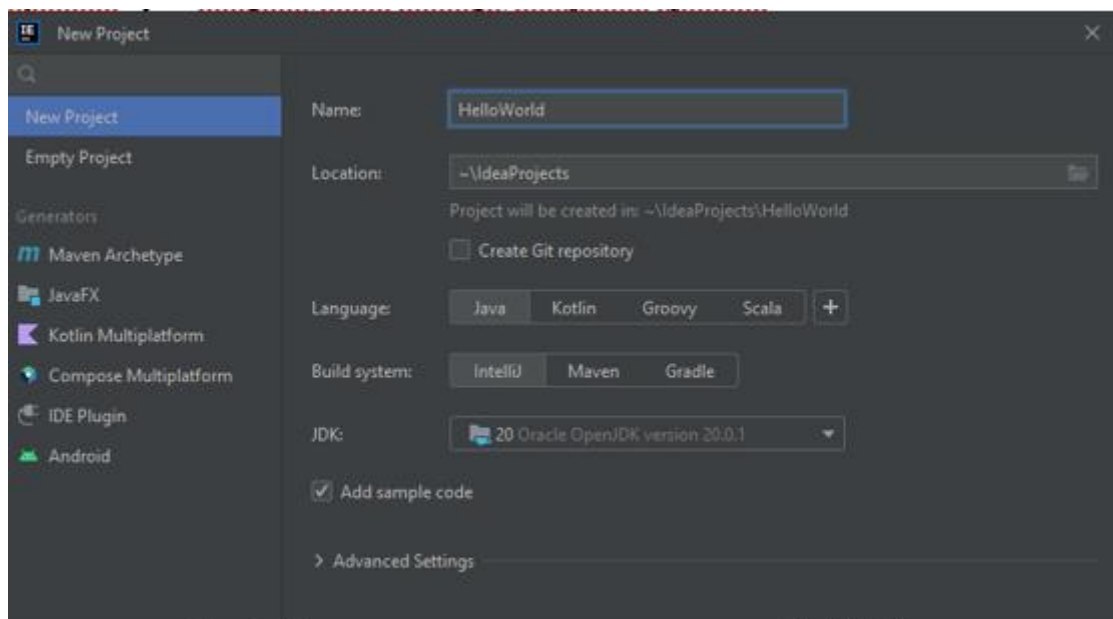


Рис.1. Приклад створення нового проєкту в IntelliJ Idea

Усі скріншоти у звіті називаються виключно рисунками та обов'язково повинні бути пронумеровані за порядком та мати відповідні назви (наприклад: Рис.1. Приклад створення нового проєкту в IntelliJ Idea).

Підпис не може відриватись від самого рисунку (розміщуватись на іншій сторінці). Номер та назва розміщуються одразу під рисунком.

Вимоги щодо оформлення рисунків та підписів: абзацний відступ відсутній; вирівнювання — по центру; шрифт — звичайний; крапка в кінці назви рисунку не ставиться.