**Лабораторна робота номер 0.** Hello world

 Встановити Intelliji Idea та Java jdk останньої версії. Cтворити maven проект та розробити в цьому проекті типову програму Hello world.  Програма повинна видавати на екран напис Hello world та закінчувати свою роботу.

**Лабораторна робота номер 1.** CVS. GIT

 Під'єднати до intellij Idea систему CVS. А саме GIT.  Створити аккаунт в хмарному середовищі github, під'єднати свій проект в intellij Idea до свого аккаунту github та завантажити  нульову лабораторну роботу на github аккаунт.

*Починаючи з цієї лабораторної роботи на титульній сторінці в паперовому виді вказувати посилання на проект лабораторних робіт GitHub   з цієї дисципліни.*

*Обов'язково перевірте, щоб ваш проект лабораторних робіт на GitHub    мав вільний доступ усіх бажаючих до вашого коду.  Обмеження доступу до ваших лабораторних робіт буде розцінено, як помилка при виконанні лабораторної роботи*

**Лабораторна робота номер 2. Основи.**

 Розробити програму, що дозволить вам створити, як з клавіатури  так і рандомно матрицю цілих чисел типу int заданої ширини та висоти(ввести з клавіатури), але не більше 20 на 20. Створити можливість пошуку в цій матриці мінімального і максимального елементу  та розрахунок середнього арифметичного. Програма може бути написана в одному класі, обов'язково розбиття на методи. Обов'язкове використання клавіатури, під час вибору ручного чи рандомного створення матриці. Створення системи зчитування з клавіатури зробити будь-яким способом, наприклад завдяки класу Scanner. Scanner являє собою найпростішу систему сканування клавіатури. Диапазон рандомних чисел для створення елементів матриці повинен зверігатись в спеціальних константах.

*Як завдання підвищеної складності додати  розрахунок середнього геометричного елементів матриці.*

**Лабораторна робота номер 3. ООП.**

 Створити програму що буде створювати та обробляти комплексний об'єкт під назвою університет(university). Програма повинна складатися з трьох частин: модель вид та контролер згідно з парадигмою mvc (Model View Controller).  Кожній з цих груп повинна відповідати package з  відповідною назвою.  В моделі повинні знаходитись усі класи що відповідають за структурні підрозділи університету. Серед них:   університет, факультет, кафедра, група, студент, людина (Human). Усі вони повинні містити  назву  типу string та голову типу Human. Студент також повинен бути породжений від  Human.  Human повинен мати поля ім'я, прізвище, побатькові та стать.  Усі поля повинні бути строковими окрім поля стать. Стать повинна використовувати спеціальний  enum  типу Sex(стать).

В цій лабораторній роботі  група View Нам не потрібна.

 Що стосується групи контроллер (controller)  то вона повинна містити менеджери що дозволяють нам створити відповідні підрозділи наприклад  StudentCreator, FacultyCreator, GroupCreator  та інші, кожен з яких повинен використовувати можливості нижчого за рівнем   створювача. Програма повинна також містити клас Run, в якому буде знаходитись точка входу та методи, що повинні дати можливість створити університет.   Процес створення університету повинен бути зроблений в методі createTypycalUniversity.

  В програмі активно рекомендується використовувати абстрактні класи та інтерфейси

**Лабораторна робота номер 4. JUnit. Json**

 Додати до лабораторної роботи 3 можливість запису університету  у формат json, запис цього формату у файл, зчитування цього формату файлу, та створення об'єкту з текстового формату json.  В  проекті повинен бути зроблений JUnit тест,  який буде виглядати наступним чином:  створити об'єкт університет(oldUniversity),  в якому в кожному підрозділі маються два підрозділи нижчого рівня. Наприклад на факультеті дві кафедри, на кожній кафедрі дві групи, на кожній групі два студенти.  Цей об'єкт повинен бути записаний в файл у форматі json. Потім з цього файлу зчитаний та відновлений як newUniversity. В тесті повинні бути порівняні newUniversity та oldUniversity за допомогою методу equals.  Якщо все зроблено правильно то університети повинні бути еквівалентні, а метод equals повинен повернути True.  Для запису та зчитування університету у форматі json повинен бути зроблений клас JsonManager.  Для безпосереднього перетворення університету  у формат json та його відновлення цього формату, можливо використання сторонніх бібліотек наприклад Gson, Jackson  чи будь-яких інших.

Для початку розробки лабораторної роботи номер 4 повністю скопіювати програмний код лабораторної роботи номер 3. Не змішувати ці роботи не в якому разі.

**Лабораторна робота номер 5. Black Jack**

 Створити гру Блек Джек (Відома як 21).  Гра карткова. Використовується  одна стандартна колода на 52 картки. Детальні правила гри можна прочитати у відкритому доступі або на Вікіпедії. Коротко про правила гри: грає людина проти комп'ютера. Людині видається дві карти. Мета гри набрати найбільшу кількість балів, але не більше 21. Після отримання двох карток людина має зробити вибір, або зупинити гру і залишити отриману кількість балів, або взяти ще одну картку. Якщо кількість балів на руках перевищує 21 то людина автоматично програє. Людина може в будь-який час зупинитись. Згідно з теорією гри жодна комбінація з двох карток не може перевищувати 21 бал. Відповідність картки та її цінності в балах написана нижче.

 Ваша програма повинна займати наступну стратегію брати картки до тих пір поки сума балів не буде де менше або дорівнювати 17.  Видача карток повинна здійснюватися завдяки механізму рандомайзер.  Для скороченого відображення цінності картки використовуємо 2  частини 1 відповідає за ранг карти 2-10,j,q,k,a,   інша відповідає за масть   Diamond,  Heart,  Spade,  Club.  Тобто назви карт будуть виглядати на кшталт 5S, 10H, AS.  Гра повинна бути зроблена з використанням усіх принципів ооп та мати зручний та дружелюбний інтерфейс.

Будь які картки від 2 до 10 дорівнюють від 2 до 10 балів відповідно. J=10 Q=10 K=10 A=1(завжди).  Що стосується різновидів гри, цінності карток можуть бути іншими, проте в цій лабораторній роботі прошу використовувати cаме ці цінності. Починаючи з цієї лабораторної роботи намагатись активно використовувати Stream API де це можливо та має сенс.

**Лабораторна робота номер 6. Jdbc**

 Створити базу даних в будь  якому сервері баз даних. Створити таблицю з переліком студентів вказати їх прізвище, ім'я, по батькові, день народження номер залікової книжки та ID.

Створити програму  що буде дозволяти   виводити на екран  інформацію про студентів  які народилися в тому чи іншому місяці року.  Програма повинна завдяки системі jdbc під'єднатися до вашої бази даних та робити до неї запроси.  Вимог до розробки бази даних немає.  Програма ж має бути написана за усіма стандартами ООП.  Та може бути спроектована за двох принципів:

- при будь-якій ситуації буде забиратися весь перелік студентів, а вже на стороні java буде зроблено пошук  необхідного

 - SQL запрос буде сформований згідно запросу який зробив користувач і вже сервер управління баз даних буде вирішувати, які самі студенти народилися в тому чи іншому місяці.

 У висновку обов'язково пояснити чому вибрали той чи інший принцип, які в нього переваги та недоліки.  Оцінка не залежить від того який сервер управління баз даних вибрали.  Перелік студентів зробити не менше 20 людей. Місяць червень зробити місяцем, коли в жодного зі студентів немає дня народження.

 SQL код створення бази даних розмістити проекті 6 лабораторної роботи  в файлі database в пекеджі resources.  Для використання цієї лабораторної роботи рекомендується активно використовувати знання отримані на дисципліні що стосуються  розробки баз даних.

 До паперового звіту обов'язково додати принтскрин з програми  в якій ви дивитесь інформацію  вашого  сервера управління баз даних, де показати створену таблицю, її ім'я та загальні відомості бази даних, наприклад назва, ім'я, назва користувача адміністратора, пароль тощо.  Для роботи з сервером управління баз даних рекомендуємо використовувати програмне забезпечення компанії jetbrains datagrip. Або вбудовану панель користування базами даних, що міститься у середовищі intellij Idea,  яка на сьогоднішній день підтримує майже всі сервери управління баз даних.