**Exercise 1:**

A screenshot of a computer

Description automatically generated

Purpose: To create terminal based applications that draw a polygon.

Class relationships:

* Drawing is used to link the Point, Line, and Polygon classes.
* A polygon is composed of at least 3 lines.
* A line is composed of 2 points.

**Exercise 2:**

A screenshot of a computer

Description automatically generated

**Exercise 3:**

A small retail shop that sells tools requires an application to manage inventory of different types of tools it sells.

Inventory →Tool

The store owner wants to be able to modify the store’s inventory by adding new tools, and deleting tools. The owner also wants to be able to search the inventory for tools by tool name, and by tool id. Currently, the information about tools available in the shop and suppliers is stored in two text files: items.txt, and suppliers.txt.

Inventory →Tool

* addTools()
* deleteTools()
* searchToolsById()
* searchToolsByName()

Supplier

* addSupplier()

The order and type of data given in these files are:

Items.txt: (id; description or name of tool; quantity in stock; price; supplier id number)

Suppliers.txt: (id; company name; address; sales contact)

Tools → FileHandler

* readItems()
* readTools()
* exportData()

The owner would also like to check the quantity of each item in stock. If the quantity of each item in stock goes below 40 items, then the program should automatically generate an order line for that item. The order line will have the supplier information and the required quantity for that item (The default quantity ordered by each item = 50 – number of existing items). All items ordered each day should be included in an order which has a randomly generated 5-digit id, and the date that was ordered. The order should be written to a text file called orders.txt.

OrderLine → Order

* makeOrder()
* checkAvailability()
* generateID()
* getDate()