Design Of Pen

Contents

[1. Design a Pen 2](#_Toc177561510)

[Overview 2](#_Toc177561511)

[Step 1: Gather the Requirements 2](#_Toc177561512)

[Step 2: Clarify Requirements 2](#_Toc177561513)

[Step 3: Identify the nouns 2](#_Toc177561514)

[Step 4: Visualize Pen 2](#_Toc177561515)

[Step 5: Create a class diagram 3](#_Toc177561516)

Code and Notes are @ <https://github.com/nishithjain/Design_Pen>

# Design a Pen

## Overview

* What classifies as a Pen? Are we designing digital pen or normal pen?
  + Anything that can write on paper is a Pen.

## Step 1: Gather the Requirements

* What type of pen are we designing? Ball Point Pen, Fountain, Gel
* Some pen will have refill and some pen will not have refill.
* Fountain pen will directly have ink.
* Ink can be of different colors.
* Every pen will have brand, price.
* Some pens, we will be able to refill and for some pens we will not be able to refill.
* Refill has Nib, Fountain pen will directly have the Nib.
* Nib can be of different radius.
* Ink can be of different types. (Sparkel, fragrance, waterproof)

## Step 2: Clarify Requirements

* Ask questions on behaviors/edge cases/future modification.

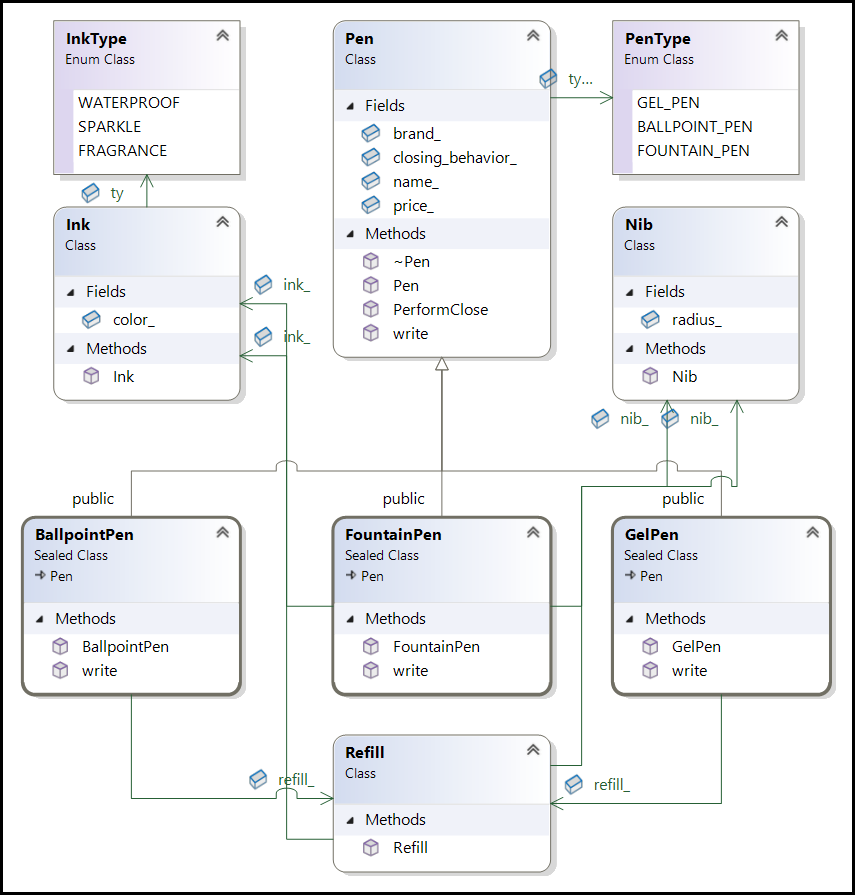
## Step 3: Identify the nouns

* Identify nouns in the requirement. 90% of the nouns in the requirement will be our classes.
* Nouns identified in the above requirements are
  + Pen
  + Ink
  + Nib
  + Refill
  + Color
  + Type (Ink, Nib, Pen)
  + Brand

## Step 4: Visualize Pen

* Visualize Pen and create classes from outside to inside.
* Often entities for which there are **multiple types** are there, these entities are defined as **abstract** class with a type attribute.
  + This type helps create factory.
* Pen is the out most part.
* Pen contains Refill. So, create a class for Refill next.
* Refill contains Nib. So, create a class for Nib.
* Refill contains Ink. So, create a class for Ink.

## Step 5: Create a class diagram

* Expectation is to show classes for core entities in the system.
* No need to mention controller classes, services, repositories.
* We need to mention use of design patterns.
* Write() -> Pens can write in different ways.
  + Multiple ways to implement write() -> Strategy Design Pattern
* Based on input, create an object of the Pen -> Factory Design Pattern.

