LAB ASSIGNMENT 2

NOOKS AND CRANNIES

You deftly press a few keys and rouse your laptop from its mechanical slumber. Bower pauses for a moment, one eyebrow raised in sarcastic interrogation. The innards of the computer whir in an audible crescendo, its struggle akin to the wheeze of an asthmatic pap. Both of you stoically await the likelihood of turned heads and smirks. This hunk of junk was all the technological assistance HQ was willing to provide for your excursion - at least for starters. Being a potential bust, nobody back home was too keen on sending heavy duty rigs unless the leads would prove worthwhile. Bower tells you she's been hard at work over at the town library, searching for bits of information that could match the vague hints given by the letters.

Speaking of which, another one of the damned things had landed in the mail back in the city, and Tillyard had promised to send a scanned copy to everyone out in the field, ASAP; but first things first. She tells you to access some maps of Whaler's End she had dug up earlier, and filter sections of them according to prominent landmarks, public areas and older neighborhoods. Categorize them neatly so she can compare them with the letters and see if this one-gas-station town could be your lucky winner. Adding to this, make sure to sort the data neatly according to those three categories, dispose of outdated information and retrieve some photos of the more peculiar monuments. You are interested in mapCatalogueNumber, stateOfDeterioration, mapType, and yearsOfStorage

* Your map sorter must be able to add, modify and dispose of unwanted maps. It must also be able to list all maps, and only maps of a type. All commands must be in the form:
  + add mapCatalogueNumber, stateOfDeterioration, mapType, yearsOfStorage
  + update mapCatalogueNumber, newStateOfDeterioration, newMapType, newYearsOfStorage
  + delete mapCatalogueNumber
  + list
  + list newMapType
* You decide to just store your maps statically, can't be bothered to do it properly now. (static memory allocation)
* You decide to use multiple files to store your code, no point jumbling everything together. (domain, repository, controller, UI in different modules)
* You have to make sure you don't re-add the same file or delete a non-existing one. The application should clearly say "No!" in that case.

Example test run:

* add 123, abc, def, 456
* list
* exit

Expected output: (formating is ignored)

[possibly some text here]123[possibly some text here]abc[possibly some text here]def[possibly some text here]456[possibly some text here]