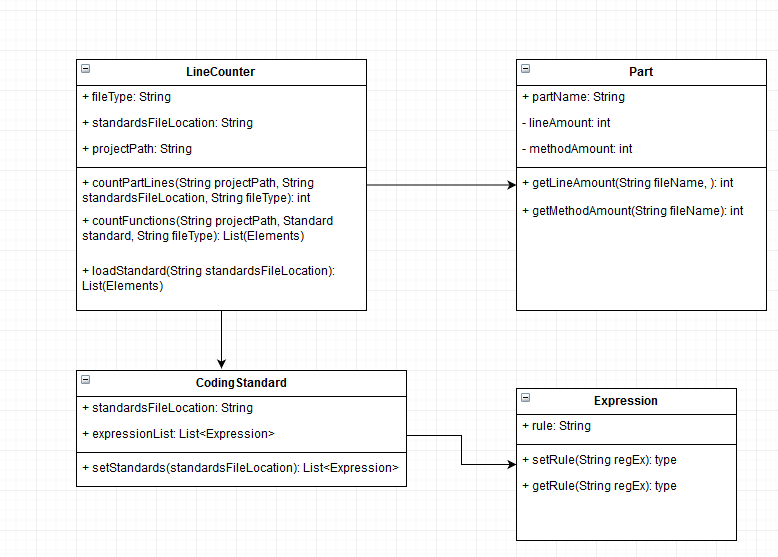
Tarea 2:

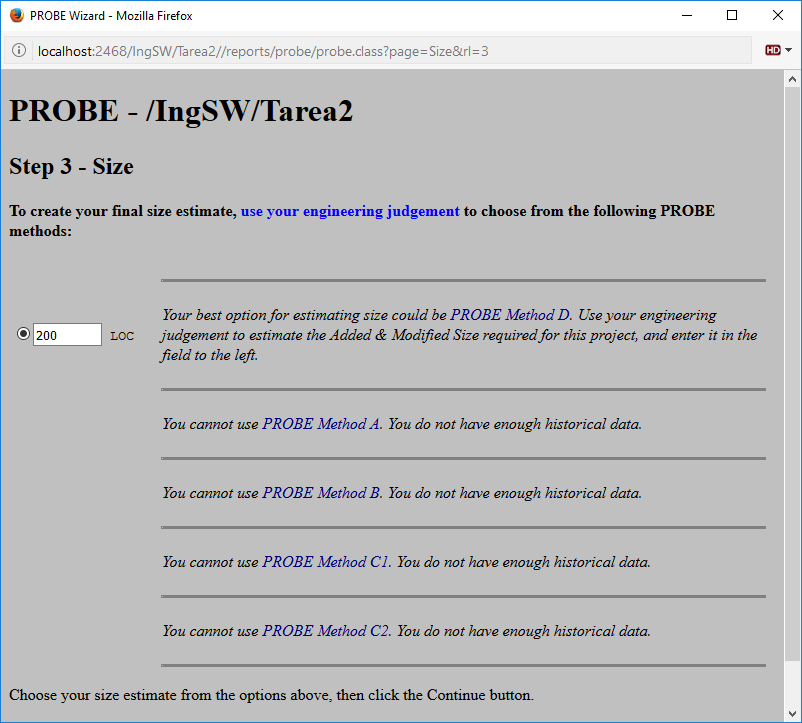
1. Diseño conceptual: se generará un archivo de diagrama de clases para ver cómo se puede solucionar el problema

como resultado de la planeación, salieron los siguientes artefactos:

Análisis conceptual:



Cálculo de tamaño por PROBE (con la ayuda del PROBE wizard de dashboard):

****

el tamaño estimado por el cálculo inicial fue aprox. 198 líneas de código; se redondeó a 200.

entradas:

- ruta de la raíz del proyecto

- tipo de archivo al que se hará el conteo (ext)\*\*

- estándar de codificación

procesos:

- contar líneas del archivo

- contar clases\*

- contar métodos

salidas:

- la cuenta total de líneas de código del programa

- una lista de tamaños (en LOC) y cantidad de items para cada clase\*

\* se entenderá como clase, un archivo.ext

\*\* para el presente, se asumirá que los archivos tienen extensión.java

\*\*\* el estandar de codificación quedará en el código; si queda tiempo, se puede dejar que lea las reglas desde un archivo.properties o archivo.txt

para llevar esto a cabo, se requiere:

prerrequisito: establecer un estándar de codificación (lógica que se usará para contar las líneas de código)

\* establecer el método que lea el fichero raíz, y que devuelva la lista de clases para ese fichero

\* un método que lea un archivo.java y, usando el estándar de codificación, devuelva las líneas de código para ese archivo

\* un método que lea un archiov.java, y usando el estándar de codificación, devuelva la cantidad de métodos

pensamientos aleatorios:

se podría hacer un metamodelo del programa?

qué estándar definir?