

TRABAJO PRACTICO N° 1**Parte C: Fundamentos – Herencia, agregación, polimorfismo****Generalidades**

- a) Prepare un directorio denominado apellido_legajo_C (reemplace con sus datos) que contenga 2 subdirectorios denominados cons_4 y cons_5
- b) Ubique en cada uno de ellos la implementación que corresponda según la consigna.
- c) Coloque en el directorio raíz (apellido_legajo_C) el documento con el informe general para esta entrega (siga el mismo formato de la anterior).
- d) Al finalizar, suba un archivo comprimido de la carpeta raíz creada (incluyendo todo su contenido) en la actividad denominada Entrega C, dentro del aula virtual.
- e) En cada consigna, Usted debe diseñar un modelo orientado a objetos, que incorpore las clases mínimas indicadas, e informarlo mediante un diagrama de clases.
- f) Fecha de entrega: 20 de setiembre de 2023

Consigna 4

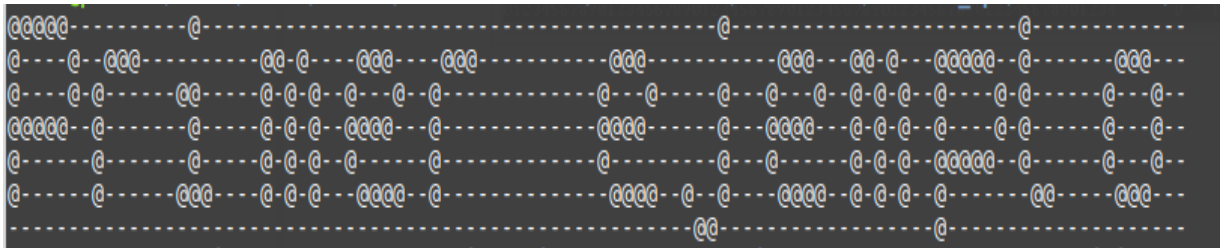
Implementar en lenguaje Python, bajo paradigma orientado a objetos, un programa de utilidad que ofrezca un menú de usuario, con la siguiente funcionalidad:

- Listar los archivos de arte ASCII disponibles en un directorio ubicado en el espacio de la aplicación.
- Seleccionar un archivo en particular mediante el tipeado de su nombre completo.
- Visualizar en la consola el contenido del archivo con el arte ASCII que contiene y un informe de análisis del mismo. Este informe debe indicar:
 - Qué caracteres se usaron en el archivo de arte ASCII
 - Cuántos caracteres hay de cada uno y el total.
- Listar, en formato JSON, las coordenadas (solamente) que corresponden a un carácter indicado por el operador, considerando que [0,0] se encuentra en la esquina superior izquierda.
- [desarrollo opcional] Obtener la línea de texto del mensaje sin formato y su longitud, usando subprocesos y el binario del generador de arte ASCII.
- Terminar la ejecución.

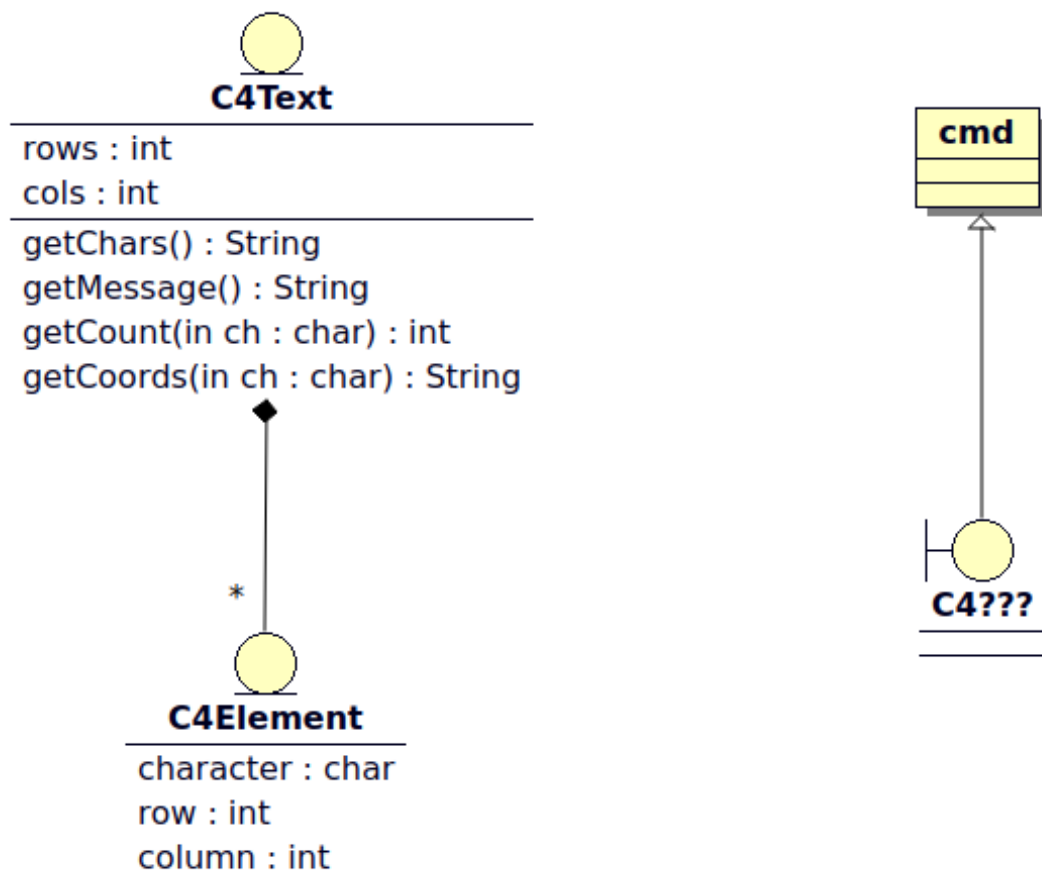
Observaciones y restricciones al diseño e implementación

- Algunos archivos de prueba, para colocar dentro del directorio ya mencionado, se encuentran en el aula virtual (denominados ejemploN.txt), sin embargo Usted puede agregar otros si lo cree conveniente. Para ello, puede usar el generador provisto (denominado *c4ascii.cpp*).
- Los archivos de arte ASCII, responden al siguiente formato interno:
 - Contenido como texto plano.
 - 7 filas de N caracteres ASCII, dependiendo N de la longitud de la línea con el mensaje.

- Un carácter de frente (para el texto del mensaje) y un carácter de fondo
- Por ejemplo:



- Actualmente el generador admite 1 carácter de frente, sin embargo se estima que su funcionalidad será ampliada para usar diferentes caracteres.
- Las opciones de visualización e informe no deben operar sobre el sistema de archivos.
- El aplicativo debe ofrecer mensajes de ayuda cuando ocurran situaciones no previstas o el usuario seleccione una opción que dependa de otra.
- Aplique un esquema de diseño que separe la capa de modelo de las de vista y control.
- Para el manejo de errores y excepciones utilizar subclases de Exception.
- Para la interfaz de usuario, usar formato CLI y reutilización mediante herencia.
- Incluya en su modelo OO las siguientes clases, agregado aquellos elementos que considere adecuados:



Consigna 5

Usando UML para representar el modelo OO y lenguaje C++ para implementación, desarrolle una pequeña herramienta para usar en un sistema de comunicaciones en red, conforme al requerimiento descripto.

- La aplicación deseada responde a una arquitectura de red C/S
 - Los valores son puestos a disposición de los clientes mediante XML-RPC, en un port específico para una IP dinámica, informada al momento del lanzamiento del servidor.
 - No se requiere de interfaz de usuario especial. Es una herramienta del tipo orden o comando de sistema (tanto para el lado el servidor como para el lado cliente). Esto implica considerar el uso de argumentos de línea de comandos.
- El Servidor debe proveer los siguientes servicios:
 - Validar cada petición en base a un identificador de usuario.
 - Permitir a un usuario, iniciar un proceso de generación de números.
 - Generar un número real, con 2 dígitos de precisión en un rango $[-CI, CS]$, donde CI y CS son los valores paramétricos de las cotas superior e inferior del rango, recibidos desde el cliente.
 - Generar un número entero en un rango $[-CI, CS]$ dado por el cliente.
 - Informar el valor de un número ya generado por el usuario, con su marca de tiempo y el rango asociados.
 - Informar la cantidad de números generados por el usuario, su suma total y su promedio.
 - Listar los números generados por el usuario remoto, desde que inició el proceso de generación, incluyendo el tipo, el instante de tiempo en que se generó, el rango de cada uno y al final, el tiempo que ocupó realizar todo el proceso desde que se hizo la primera petición.
- El cliente debe permitir:
 - Solicitar un número entero en un rango indicado explícitamente.
 - Solicitar un número real en un rango indicado explícitamente.
 - Solicitar un número entero sin indicar el rango. En este caso el servidor usa el mismo rango de la petición entera anterior.
 - Solicitar un número real sin indicar el rango. En este caso el servidor usa el mismo rango de la petición real anterior.
 - Solicitar la información asociada a un número generado anteriormente. En este caso, la petición se realiza enviando el ordinal deseado.
 - Solicitar la estadística básica
 - Solicitar el listado de números generados.
 - Mostrar en pantalla cada respuesta obtenida.
 - Guardar la respuesta obtenida, en un archivo usando formato XML, bien formado. El nombre del archivo se provee en cada petición.

Observaciones y restricciones al diseño e implementación

- Si bien son pocos usuarios, cada uno se identifica por un número de 3 dígitos.
- Los identificadores de los usuarios habilitados se encuentran almacenados en un archivo de texto predefinido del lado servidor.
- Un usuario puede acceder sólo a los datos generados por él.
- La marca de tiempo de generación de cada número corresponde a la cantidad de segundos desde que el usuario inició el proceso.
- Contemplar el control de errores en los escenarios básicos de operación usando excepciones.
- Incluya los mensajes al usuario que correspondan, a ayudas de operación o notificación de errores.
- Aplique un diseño de 2 capas con separación entre el modelo y la presentación/control.
- Por simplicidad, se sugiere el uso de la librería XML-RPC para C++, disponible desde la URL <http://xmlrpcpp.sourceforge.net/>.

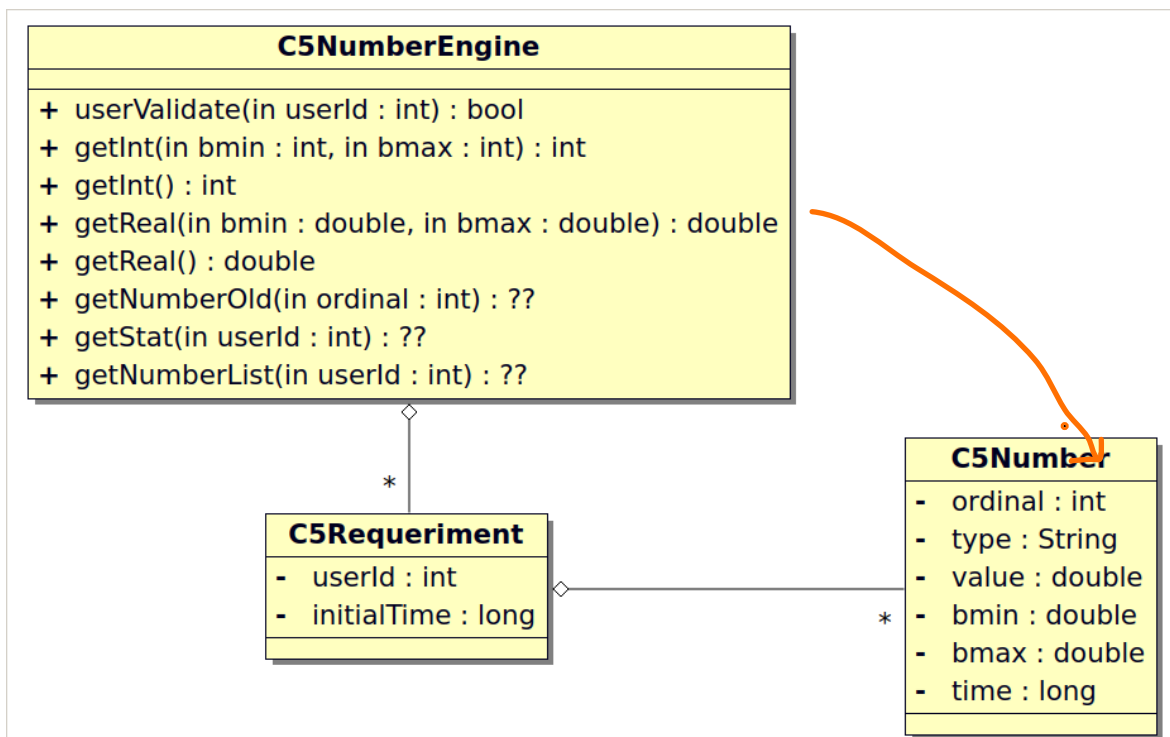
Considere la lectura de la documentación disponible en la página del proyecto.

Revise el video explicativo de clase, sobre el montaje de un aplicativo C/S usando dicha librería, en <https://youtu.be/B4vzzSqODvA> (aprox. desde 4:40 minutos)

Puede ver un aplicativo similar en <https://youtu.be/2an54kpvg3c>

Sin embargo, queda a su preferencia el uso de una librería similar.

- Incluya en su modelo OO las siguientes clases, agregado aquellos elementos que considere adecuados:



Anexo: referencias de consulta y extensión

Se agregan a las ya listadas en la parte B

UML/C++

- <https://pdfcoffee.com/qdownload/aprendiendo-uml-en-24-horas-joseph-schmuller-pdf-free.html>
- <https://docs.nomagic.com/pages/viewpage.action?pageId=38044261>

XML

- https://developer.mozilla.org/es/docs/Web/XML/XML_introduction
- <https://aws.amazon.com/es/what-is/xml/>

JSON

- <https://blog.hubspot.es/website/que-es-json>
- <https://www.freecodecamp.org/espanol/news/que-es-un-archivo-json-ejemplo-de-codigo-en-javascript/>

Arte ASCII /AA

- <https://www.genbeta.com/web/esta-web-convierte-facilmente-cualquier-imagen-ascii-art>
- <https://manytools.org/hacker-tools/convert-images-to-ascii-art/>
- <https://pixlab.io/art>
- <https://web.archive.org/web/20050720002833/http://www.degraeve.com/img2txt.php>
- <https://web.archive.org/web/20141203131509/http://www.jpg2txt.com/convert.html>

Uso de cmd y definición de clases derivadas (Python)

- <https://python-para-impacientes.blogspot.com/2016/07/cmd-construyendo-un-interprete-de.html>
- <http://hugoruscitti.github.io/2010/07/22/2010-07-23-creando-interpretes-interactivos-con-python/>
- <https://bip.weizmann.ac.il/course/python/PyMOTW/PyMOTW/docs/cmd/index.html>

Manejo de Errores y Excepciones mediante Subclases de Exception (Python)

- <http://docs.python.org.ar/tutorial/3/errors.html>
- <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion9/errores.html>
- <https://pythonista.io/cursos/py101/gestion-de-excepciones>