4 Types

* Drive
* Folder
* Zip file
* Text

File System (If A contains B, A is parent of B)

* **Drive** (drive cannot be contained)
  + File
    - Text
    - Zip -> folders files (text or zip)
  + Folder -> Folder/File (Zip , txt)

Every entity has the following properties: (FileSystem Interface)

* **Type** – The type of the entity (one of the 4 type above).
* **Name** - An alphanumeric string. Two entities with the same parent cannot have the same name. Similarly, two drives cannot have the same name.
* **Path** – The concatenation of the names of the containing entities, from the drive down to and including the entity. The names are separated by ‘\’.
* A text file has a property called **Content** which is a string.
* **Size** – an integer defined as follows:
* For a text file – it is the length of its content.
* For a drive or a folder, it is the sum of all sizes of the entities it contains.
* For a zip file, it is one half of the sum of all sizes of the entities it contains.

File System Operations:

1. **Create** – Creates a new entity.

Arguments: Type, Name, Path of parent.

Exceptions: Path not found; Path already exists; Illegal File System Operation (if any of the rules a-d above is violated).

1. **Delete** – Deletes an existing entity (and all the entities it contains).

Arguments: Path

Exceptions: Path not found.

1. **Move** – Changing the parent of an entity.

Arguments: Source Path, Destination Path.

Exceptions: Path not found; Path already exists, Illegal File System Operation.

1. **WriteToFile** – Changes the content of a text file.

Arguments: Path, Content

Exceptions: Path not found; Not a text file

Tasks:

1. Come up with the design for this system. Full implementation is not required, but only to the level which you feel is a “proof of concept”.
2. Show a sketch of implementation of the **Move** operation.
3. Explicitly implement the property **Size**.