How to implement a functionObject

How to implement a functionObject

- We will here do the basic principle of creating a new functionObject
- We will use the framework from the `forces` functionObject
- Start by copying the `forces` functionObject to your user directory:

```
foam
cp -r --parents src/postProcessing/functionObjects/forces/forces $WM_PROJECT_USER_DIR
cd $WM_PROJECT_USER_DIR/src/postProcessing/functionObjects
mkdir myFunctionObject
mv forces/forces/* myFunctionObject
rm -r forces
cd myFunctionObject
rm *.dep
sed -i 's/forces/myFunctionObject/g' *
```

- Rename the files (this command may differ for different OS's):
  In our case:
  `rename forces myFunctionObject *`
  otherwise, try:
  `rename 's/forces/myFunctionObject/' *`

How to implement a functionObject

- Look at the original `Make/{files,options}` files to create your own:

  ```
  mkdir Make
  ```

  The `Make/files` file should contain:

  ```
  myFunctionObject.C
  myFunctionObjectFunctionObject.C
  LIB = $(FOAM_USER_LIBBIN)/libmyFunctionObject
  ```

  Copy the original `options` file:

  ```
  cp $FOAM_SRC/postProcessing/functionObjects/forces/Make/options Make
  ```

  (we could clean this up, to only include the necessary, if we want to)

- At this point we can compile to make sure that it works:

  ```
  wmake libso
  ```

How to implement a functionObject

- Test it for `icoFoam/cavity`, by adding to the `system/controlDict`:

```
functions
{
forces
{
    type myFunctionObject;
    functionObjectLibs ("libmyFunctionObject.so");
    outputControl timeStep;
    outputInterval 1;
    patches          (movingWall);
    pName            p;
    UName            U;
    rhoName          rhoInf;
    log              yes;
    rhoInf           1000;
    CofR             (0 0 0);
}
};
```

- It should do the same as if we specify (inside `functions{};`:

```
forces
{
type forces;
functionObjectLibs ("libforces.so");
...
}
```

How to implement a functionObject

- Let's have a look at the files:

  ```
  myFunctionObject.C
  myFunctionObject.H
  myFunctionObjectFunctionObject.C
  myFunctionObjectFunctionObject.H
  ```

  The last two are just wrappers to make the code work as a functionObject, so we do not have to change more there (we already changed `forces` in those files to `myFunctionObject`)

- We need to change the files:

  ```
  myFunctionObject.C
  myFunctionObject.H
  ```

How to implement a functionObject

- A simple modification in `myFunctionObject.C`:

```
void Foam::myFunctionObject::writeFileHeader(const label i)
{
    file()<< "This is my header"<< endl;
}
```

- In `myFunctionObject.C`, in member function `void Foam::myFunctionObject::write()`, inside `if (Pstream::master())`:

```
        if (log_)
        {
            Info<< type() << " test" << nl;
        }
        file() << obr_.time().value() << tab << "fileText" << endl;
```

- Have a further look at the files to see how the forces are calculated.
- If you like, clean up all related to the original forces functionObject after investigating the files...

How to implement a functionObject - clean up

- Member data specific to original `forces`:
  All including and after `List<Field<vector> > force_;`
  I.e., to clean up, remove all related to those member data.
- Remove the protected member functions specific to original `forces`:

```
tmp<volSymmTensorField> devRhoReff() const;
tmp<volScalarField> mu() const;
tmp<volScalarField> rho() const;
scalar rho(const volScalarField& p) const;
void applyBins
(
    const vectorField& Md,
    const vectorField& fN,
    const vectorField& fT,
    const vectorField& fP,
    const vectorField& d
);
void writeBins() const;
```

How to implement a functionObject - clean up

- Remove the public member functions specific to original `forces`:

```
virtual void calcForcesMoment();
virtual vector forceEff() const;
virtual vector momentEff() const;
```

- Remove include-files that are no longer needed (trial-and-error)

How to implement a functionObject - further notes

- Your functionObject is a sub-class of `functionObjectFile`:
  `$FOAM_SRC/OpenFOAM/db/functionObjects/functionObjectFile/functionObjectFile.C`
  That is where the output file is created, etc.
- Search for `functionObjectFile.H` in Doxygen to see more sub-classes.