AMERICAN UNIVERSITY OF ARMENIA
College of Science and Engineering
CS 140 Mechanics
PROJECT

PROGRESS EVALUATIONS: Saturday April 30 2022, Friday May 06 2022
FINAL SUBMISSION DEADLINE: Thursday, May 12 2022, not later than 23:59

CONTACTS: Varazdat Stepanyan, varazdat_stepanyan@edu.aua.am
Tsovinar Karapetyan, tsovinar.karapetyan.0510@gmail.com
Suren Khachatryan, skhachat@aua.am

Project Objective

The project aims at design and simulation of a mechanical converter that converts binary representation of a number into its value in the decimal numeral system. The input consists of 8 signals representing bits – 1 if a signal is recorded, and 0 – if the signal is missing. The device generates a spring oscillation the frequency of which corresponds to the magnitude of the recorded input.

Task 1

Write a class Spring that implements the concept of a 1D massless spring and, hence, encapsulates its stiffness **double** *k* with the default value equal 1. Add the following methods:

1. The default constructor and an overloaded constructor that specifies the stiffness.
2. The public getter and a private setter;
3. Overloaded public *move()* methods that return an array of coordinates of an oscillating mass:
   - **double**[] *move*(**double** *t*, **double** *dt*, **double** *x0*, **double** *v0*) – a body of unit mass oscillates during a period **double** *t* starting from $t = 0$ with initial conditions $x(0) = x0$ and $v(0) = v0$. The coordinate is computed per each **double** *dt* time step;
   - **double**[] *move*(**double** *t*, **double** *dt*, **double** *x0*) – a body of unit mass oscillates during a period **double** *t* starting from $t = 0$ with initial conditions $x(0) = x0$ and $v(0) = 0$. The coordinate is computed per each **double** *dt* time step;
   - **double**[] *move*(**double** *t0*, **double** *t1*, **double** *dt*, **double** *x0*, **double** *v0*) – a body of unit mass oscillates from $t = t0$ till $t = t1$ with initial conditions $x(t0) = x0$ and $v(t0) = v0$. The coordinate is computed per each **double** *dt* time step;
   - **double**[] *move*(**double** *t0*, **double** *t1*, **double** *dt*, **double** *x0*, **double** *v0*, **double** *m*) – a body of a specified mass **double** *m* oscillates from $t = t0$ till $t = t1$ with initial conditions $x(t0) = x0$ and $v(t0) = v0$. The coordinate is computed per each **double** *dt* time step;

Task 2

1. Continue with the class Spring and add the following public methods:
   - Spring *inSeries*(Spring *that*) – takes by reference a Spring *that* argument, connects it with **this** Spring in series and returns a new Spring object that represents the equivalent spring;
   - Spring *inParallel*(Spring *that*) – takes by reference a Spring *that* argument, connects it with **this** Spring in parallel and returns a new Spring object that represents the equivalent spring;

2. Write a class SpringArray and implement the following public static methods:
   - Spring *equivalentSpring*(String *springExpr*) – takes a String expression that represents connections of springs of unit stiffness and returns the equivalent spring. The Spring springExpr is a valid expression of balanced braces {} and brackets []. Empty brackets without nested braces and brackets represent a single spring of unit stiffness. Brackets with nested braces and brackets represent springs connected in parallel. Braces with nested braces and brackets represent springs connected in series.
   - Spring *equivalentSpring*(String *springExpr*, Spring[] *springs*) – takes a String expression that represents connections of springs specified by a Spring array Spring[] *springs* and returns the equivalent spring.

Task 3

Write a class FT that implements the concept of Fourier transform / series. It transforms an array of coordinate values at different time moments into an array of the amplitudes of harmonic oscillations. Declare member variables and implement methods as needed.

Task 4

Write a class Converter that aims at conversion of a binary representation of a single byte into its decimal value. Consider a sequence of 8 bits and design a system of springs that implements each of them.

- Add a method that takes as its argument a sequence of 8 binary digits and adds connects the corresponding spring systems into a general system.
- Add a method that connects to the obtained system of spring a body of unit mass and computes its oscillations.
- Add a method that calculates the frequency amplitudes of the oscillations using the implemented Fourier transform.
- Add a method that determines the decimal value of the original binary sequence using the computed frequency amplitudes.

Submission format

1. Before starting the project create a repository to keep there all project deliverables. Share it with Varazdat Stepanyan (varazdat_stepanyan@edu.aua.am), Tsovinar Karapetyan (tsovinar.karapetyan.0510@gmail.com) and. Suren Khachatryan (skhachat@aua.am).
2. The project deadline is Thursday May 12, not later than 23:59. No special submission is needed – the project will be evaluated based on the repository status at the moment of the deadline.
3. Two progress evaluations will be conducted before the final deadline – on Saturday April 30 and Friday May 06. Again, no special submissions are needed – the progress will be evaluated based on the ongoing repository status. The progress includes and not limited to understanding, study, reading, design, development, testing, analysis, etc.
4. This is an individual project. Therefore, the individual contributions must be explicitly stated in the project docs.
5. You are welcome to use / reuse / consult external sources, including open-source code, electronic and hard-copy texts, videos, group discussions, instructor / TA assistance, etc. All such sources must be explicitly acknowledged / referenced. Any unreferenced use of external sources will violate the rules of the academic integrity.