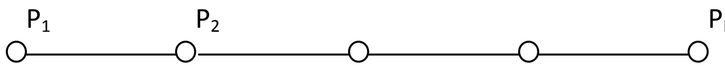
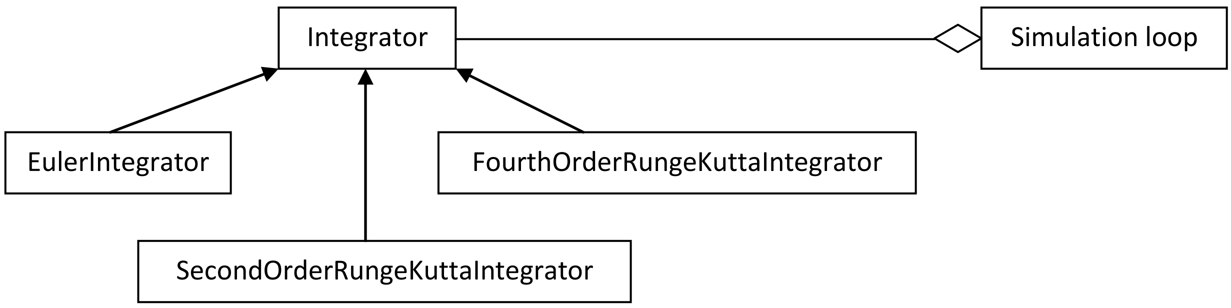
Exercise 3

Implement the simulation of a pendulum constructed from a chain of springs:



1. The parameters of the particles $P_1,\dots,P_N$, which form the knots of the chain, are stored in the file particle.txt in the following form:
   x_1 y_1 m_1
   x_2 y_2 m_2
   …
   x_N y_N m_N
   where x_i, y_i are the initial particle coordinates and m_i is the particle mass.
2. The particle P_1 is fixed and cannot move.
3. The force of Earth's gravity $(0, -m_i g)$ acts on all particles.
4. If particles P_i and P_j are connected with a spring, the following force acts on the particle P_i:
   $$f_i(p, v) = \left(k_s\left(L_{ij} - L_0\right) + k_d\left(v_j - v_i\right) \cdot r_{ij}\right) r_{ij}$$
   where v_i is the velocity of the i-th particle, $L_{ij} = \|p_j - p_i\|$, $r_{ij} = \frac{p_j - p_i}{\|p_j - p_i\|}$, $p_i = (x_i, y_i)$, $p = (p_1, \dots, p_N)$, $v = (v_1, \dots, v_N)$, L_0 is the spring rest length, k_s is the spring stiffness, and k_d is the damping coefficient.
5. In this exercise, the springs have no mass and $k_s = 50$, $k_d = 1$, $L_0 = 1$ for each spring.
6. The simulation is based on Newton's second law:
   $$\begin{cases} p_i'(t) = v_i(t) \\ v_i'(t) = f_i\left(p(t), v(t)\right)/m_i + (0, -g) \end{cases}$$
   or, in the matrix form,
   $$z'(t) = F(z(t)),$$
   where
   $$z(t) = \begin{bmatrix} p_1(t) & \cdots & p_N(t) \\ v_1(t) & & v_N(t) \end{bmatrix},$$
   $$F(z(t)) = \begin{bmatrix} v_1(t) & \cdots & v_N(t) \\ f_1\left(p(t), v(t)\right)/m_1 + (0, -g) & & f_N\left(p(t), v(t)\right)/m_N + (0, -g) \end{bmatrix}.$$
7. The following time integration methods should be implemented:
   a. The Euler method:
   $$z(t + \Delta t) = z(t) + \Delta t F(z(t))$$
   b. The second order Runge-Kutta method:
   $$k_1 = F\left(z(t)\right)$$
   $$k_2 = F\left(z(t) + \frac{\Delta t}{2} k_1\right)$$
   $$z(t + \Delta t) = z(t) + \Delta t k_2$$
   c. The fourth order Runge-Kutta method:
   $$k_1 = F\left(z(t)\right)$$
   $$k_2 = F\left(z(t) + \frac{\Delta t}{2} k_1\right)$$
   $$k_3 = F\left(z(t) + \frac{\Delta t}{2} k_2\right)$$
   $$k_4 = F(z(t) + \Delta t k_3)$$
   $$z(t + \Delta t) = z(t) + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
8. In this exercise, $\Delta t = 0.01$ s.

9. In order to compute the particle positions for the next frame, 10 time integration steps should be applied. The particle positions for each frame should be printed to the file frame_NNNN.txt, where NNNN is the 4-digit frame index going from 0 to 499. The format of this file is as follows:
   x_1 y_1
   x_2 y_2
   …
   x_N y_N
   500 frames should be generated in total. Then, the motion of the pendulum can be visualized using the gnuplot command `load "animate_pendulum.plt"`, which creates a gif-file.
10. All time integration methods should be accessible using the common interface `Integrator`, which is utilized in the simulation loop:



11. The particular integration method should be selected based on the command line argument, which can take the following values:
    a. Euler
    b. Runge-Kutta-2
    c. Runge-Kutta-4

   For example, if the program is launched as “Exercise3 Runge-Kutta-4”, then the fourth order Runge-Kutta integration should be utilized.
12. The entry point and the simulation loop are in the file Exercise3.cpp.