```
      box.initialize();
      plotFrame.clearData();
      displayFrame.setPreferredMinMax(0, 1, 0, 1);
   }

   public void doStep() {
      box.step();
      plotFrame.append(0, box.time, box.nleft);
   }

   public void reset() {
      // clicking reset erases positions of particles
      control.setValue("Number of particles", 64);
      plotFrame.clearData();
      enableStepsPerDisplay(true);
      setStepsPerDisplay(10);
   }

   public static void main(String[] args) {
      SimulationControl.createApp(new BoxApp());
   }
}
```

How long does it take for the system to reach equilibrium? How does this time depend on the number of particles? After the system reaches equilibrium, what is the magnitude of the fluctuations? How do the fluctuations depend on the number of particles? Problems 7.2 and 7.3 explore such questions.

Exercise 7.1 Simple tests of operators and methods

(a) It is frequently quicker to write a short program to test how the operators and methods of a computer language work than to look them up in a manual or online. Write a test class to determine the values of `3/2`, `3.0/2.0`, `(int) (3.0/2.0)`, `2/3`, and `(int) (-3/2)`.

(b) Determine the behavior of the methods `Math.round(arg)`, `Math.ceil(arg)`, and `Math.rint(arg)`.

(c) Write a program to test whether the same sequence of random numbers appears each time the program is run if we use the method `Math.random()` to generate the sequence.

(d) Create an object from class `Random` and use the methods `setSeed(long seed)` and `nextDouble()`. Show that you obtain the same sequence of random numbers if the same seed is used. One reason to specify the seed rather than to choose it at random from the time (as is the default) is that it is convenient to use the same random number sequence when testing a program. Suppose that your program gives a strange result for a particular run. If you notice a error in the program and change the program, you would want to use the same random number sequence to test whether your changes corrected the error. Another reason for specifying the seed is that another user could obtain the same results if you tell them the seed that you used. ■

Problem 7.2 Approach to equilibrium

(a) Use `BoxApp` and `Box` and describe the nature of the evolution of n, the number of particles on the left side of the box. Choose the total number of particles N to be $N = 8$, 16, 64, 400, 800, and 3600. Does the system reach equilibrium? What is your qualitative criterion for equilibrium? Does n, the number of particles on the left-hand side, change when the system is in equilibrium?

(b) The algorithm we have used is needlessly cumbersome, because our only interest is the number of particles on each side. We used the positions only for visualization purposes. Because each particle has the same chance to go through the hole, the probability per unit time that a particle moves from left to right equals the number of particles on the left divided by the total number of particles, that is, $p = n/N$. Modify the program so that the following algorithm is implemented.

   (i) Generate a random number r from a uniformly distributed set of random numbers in the interval $0 \le r < 1$.

   (ii) If $r \le p = n/N$, move a particle from left to right, that is $n \to n - 1$; otherwise, $n \to n + 1$.

(c) Does the time dependence of n appear to be deterministic for sufficiently large N? What is the qualitative behavior of $n(t)$? Estimate the time for the system to reach equilibrium from the plots. How does this time depend on N? ■

Problem 7.3 Equilibrium fluctuations

(a) As a rough measure of the equilibrium fluctuations, visually estimate the deviation of $n(t)$ from $N/2$ for $N = 16$, 64, 400, 800, and 3600? Choose a time interval that is bigger than the time needed to reach equilibrium. How do your results for the deviation depend on N?

(b) A better measure of the equilibrium fluctuations is the mean square fluctuations Δn^2, which is defined as

$$\begin{aligned}\Delta n^2 &= \langle (n - \langle n \rangle)^2 \rangle = \langle n^2 \rangle - 2\langle n \langle n \rangle \rangle + \langle n \rangle^2 \\ &= \langle n^2 \rangle - 2\langle n \rangle^2 + \langle n \rangle^2 = \langle n^2 \rangle - \langle n \rangle^2.\end{aligned} \tag{7.1}$$

The brackets $\langle \cdots \rangle$ denote an average taken after the system has reached equilibrium. The relative magnitude of the fluctuations is $\Delta n / \langle n \rangle$. Modify your program so that averages are taken after equilibrium has been reached. Run for a time that is long enough to obtain meaningful results. Compute the mean square fluctuations Δn^2 for the same values of N considered in part (a). How do the relative fluctuations, $\Delta n / \langle n \rangle$, depend on N? (You might find it helpful to see how averages are computed in Listings 7.3 and 7.4.) ■

From Problem 7.2 we see that $n(t)$ decreases in time from its initial value to its equilibrium value in an almost deterministic manner if $N \gg 1$. It is instructive to derive the time dependence of $n(t)$ to show explicitly how chance can generate deterministic behavior. If