

## **Problem statement:**

Two cats and three mice are in a house with 10 rooms.

- The cats and mice move from room to room randomly, as long as they are alive.
- When a cat enters a room:
  - If a mouse is in that room, the cat eats the mouse.
  - Then the cat sleeps in that room for a while.
- When a mouse enters a room:
  - The mouse sometimes leaves immediately,
  - but otherwise it sleeps in that room for a while.

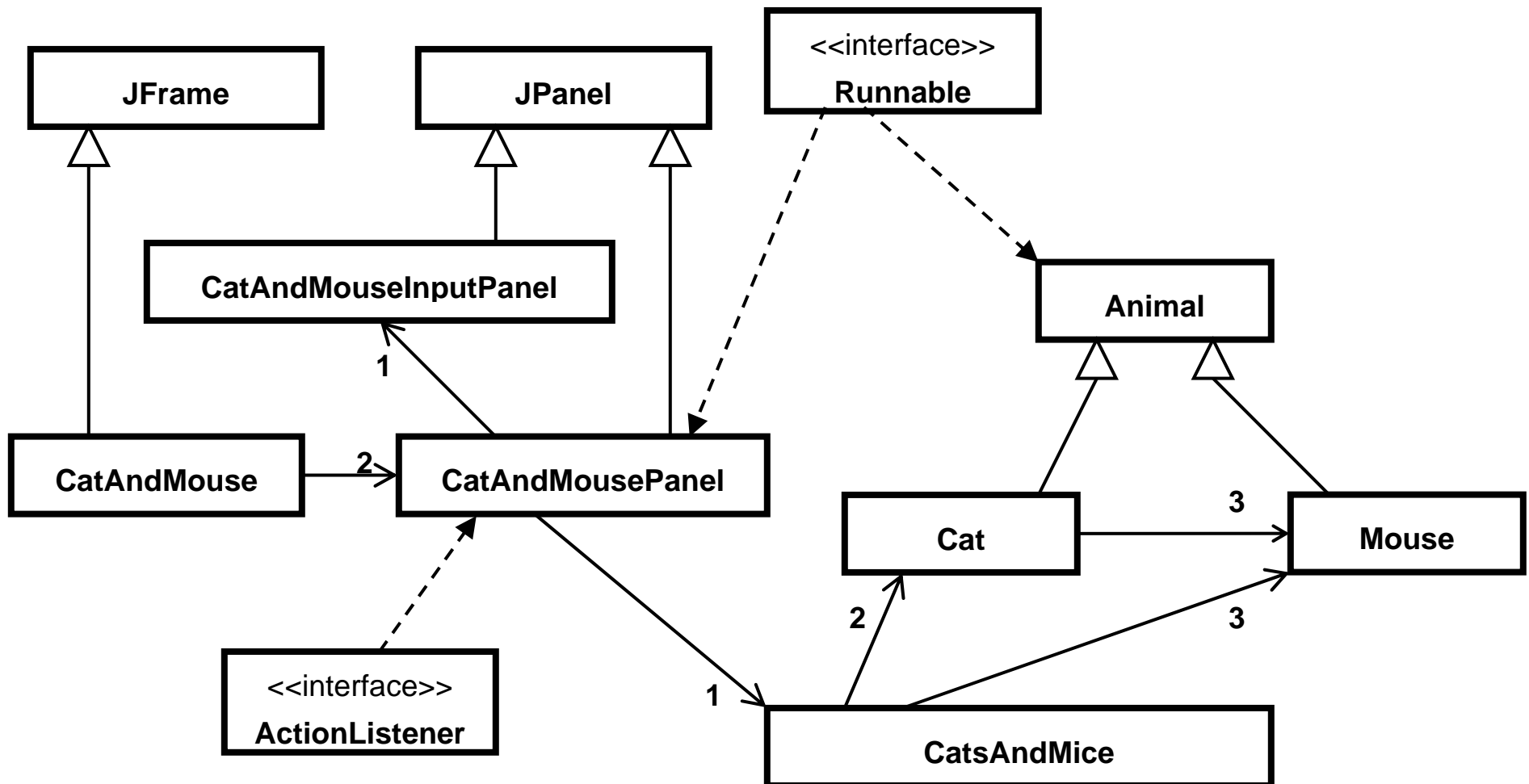
Notes:

1. The rooms are numbered from 1 to 10.
2. If there is more than one mouse in the room, the cat eats all of them.
3. Each mouse has a “skittishness” that is a number between 0 and 1. When a mouse enters a room, it leaves the room immediately with probability per its skittishness. For example, if a mouse’s skittishness is 0.2, then with probability 0.2 the mouse immediately leaves the room that it enters and with probability 0.8 it sleeps in that room for a while.
4. Each animal has a “timeToSleep” that is the number of milliseconds that the animal sleeps in a room before moving to another randomly chosen room.
5. When an animal moves to a new (randomly chosen) room, the new room should really be new (i.e., not the same room in which the animal currently is in).

## Problem statement (repeated for your convenience):

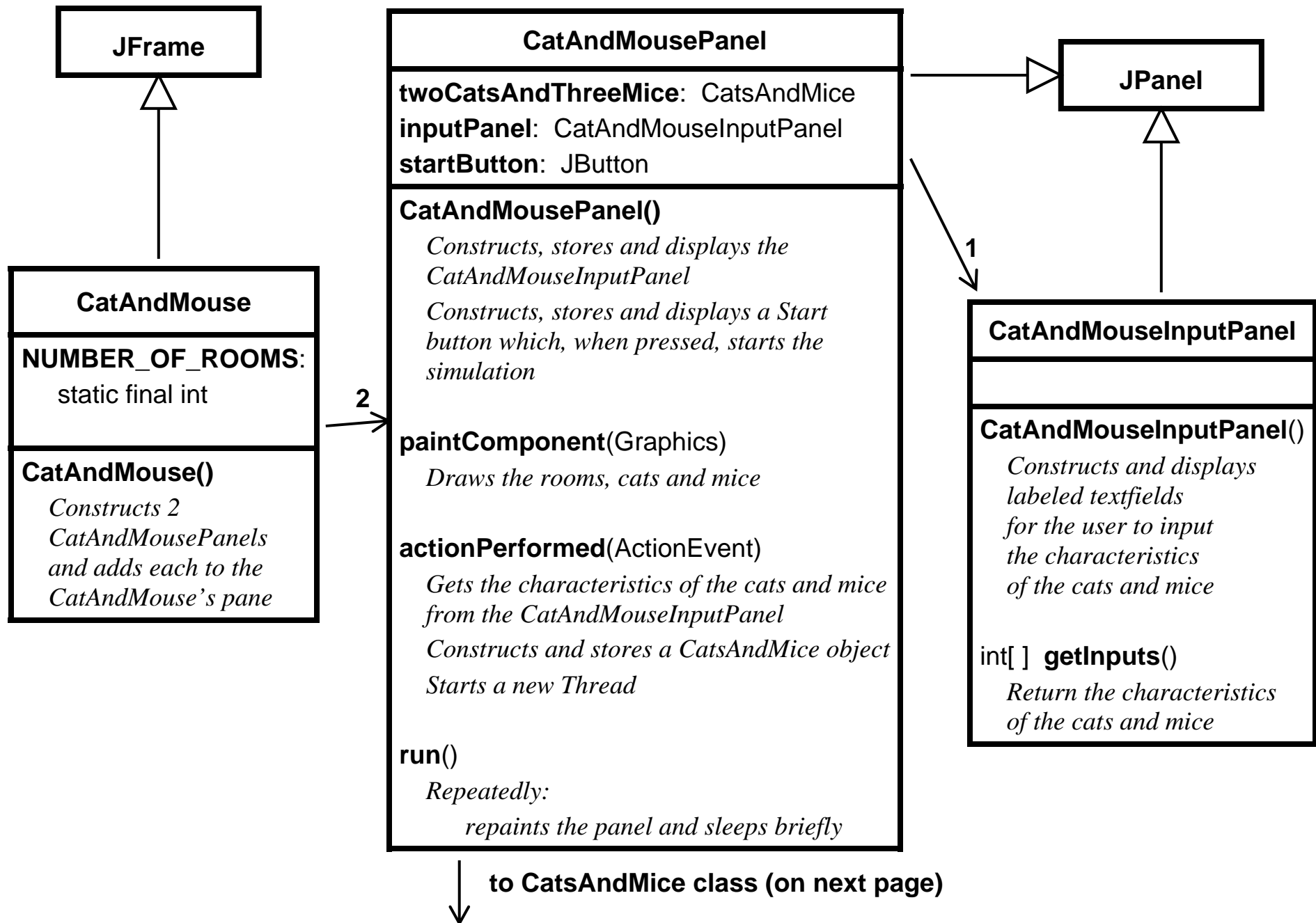
Two cats and three mice are in a house with 10 rooms. The cats and mice move from room to room randomly, as long as they are alive. When a cat enters a room, if a mouse is in that room, the cat eats the mouse. Then the cat sleeps in that room for a while. When a mouse enters a room, the mouse sometimes leaves immediately, but otherwise it sleeps in that room for a while.

## Classes and associations



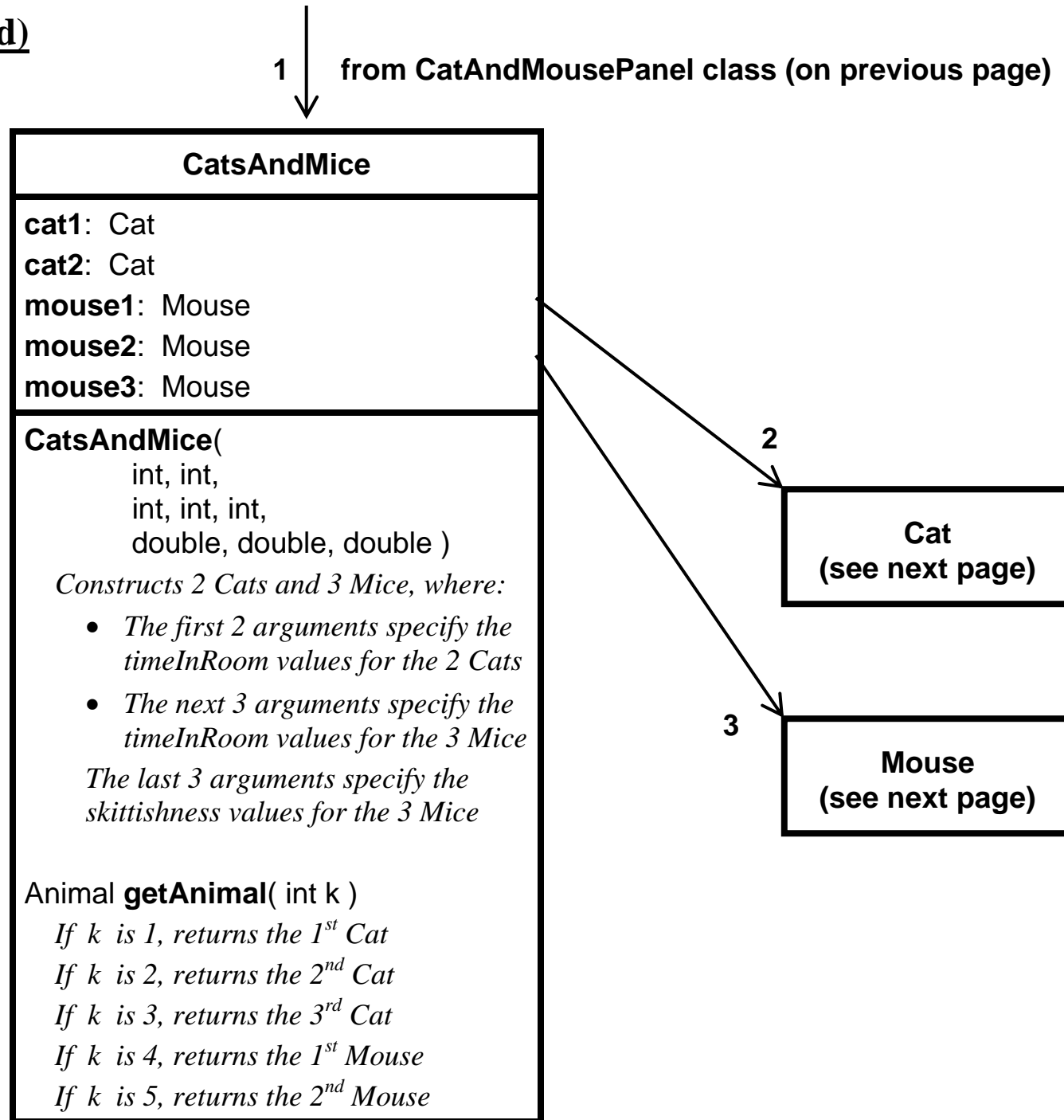
## Classes:

We have implemented *all* the classes on this page



## Classes (continued)

You implement only the *constructor* of this class



## Classes (continued) You implement *all* of these classes

