

2007 AP[®] COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

2. This question involves reasoning about the code from the Marine Biology Simulation case study. A copy of the code is provided as part of this exam.

A `PounceFish` is a type of fish that looks for prey and then "pounces" on it. A `PounceFish` can see only a limited distance in its forward direction. If the `PounceFish` sees another fish, it rushes forward and eats the nearest one that it sees, ending up in the location where its prey was originally located. If the `PounceFish` does not see another fish, it acts as a `Fish`.

The `PounceFish` class is shown below.

```
public class PounceFish extends Fish
{
    private int range;    // the distance that a PounceFish can see; range > 0

    /** Looks ahead range locations in current direction
     *  @return the nearest fish in that direction within range (if any);
     *          null if no such fish is found
     */
    private Fish findFish()
    {    /* to be implemented in part (a) */    }

    /** Acts for one step in the simulation
     */
    public void act()
    {    /* to be implemented in part (b) */    }

    // There may be fields, constructors, and methods that are not shown.
}
```

2007 AP[®] COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

The following diagrams show an example environment containing a PounceFish (represented by P) and other fish (represented by F1, F2, etc.). The direction of the PounceFish is indicated by the character ">" showing that, in this example, the PounceFish is facing east. If the PounceFish can see 2 or more locations ahead in its forward direction, it will see fish F3 as shown in the first diagram and will move to that location to eat it, causing F3 to die as shown in the second diagram.

Environment before the PounceFish acts

		North							
			0	1	2	3	4	5	
West	0				F1				
	1	F2		P>		F3	F4		East
	2			F5					
	3								
		South							

Environment after the PounceFish acts

		North							
			0	1	2	3	4	5	
West	0				F1				
	1	F2				P>	F4		East
	2			F5					
	3								
		South							

If the PounceFish in the first diagram above could see only 1 location ahead, it would not see any prey and therefore would act as an ordinary fish.

2007 AP® COMPUTER SCIENCE A FREE-RESPONSE QUESTIONS

- (a) Write the `PounceFish` method `findFish`. If any fish are located within `range` locations in the direction that the `PounceFish` is currently facing, the method returns the nearest of these. Otherwise, the method returns `null`.

Complete method `findFish` below.

```
/** Looks ahead range locations in current direction
 *  @return the nearest fish in that direction within range (if any);
 *          null if no such fish is found
 */
private Fish findFish()
```

- (b) Override the `act` method for the `PounceFish` class. A `PounceFish` attempts to find a fish that it can eat. If it finds such a fish, the `PounceFish` eats it (causing it to die) and moves to its location. If the `PounceFish` does not find a fish that it can eat, it acts as an ordinary fish.

In writing `act`, assume that `findFish` works as specified, regardless of what you wrote in part (a).

Complete method `act` below.

```
/** Acts for one step in the simulation
 */
public void act()
{
    if ( ! isInEnv() )
        return;

    // Write your code below.
```