Abstract data types "part II" unit:

- Nested classes (Wednesday)
- Generics (**today** and Friday, plus lab Thursday)
- Back to ADTs: Stacks and Queues (next week Monday and Wednesday)

Today (and Friday):

- Writing generic methods
- Writing generic classes
- Example: Homemade generic linked list class (including lab retrospective)
- ~~How generics are implemented~~
- Generics and arrays
- Example: Extending pred/prey

```c
struct A_t
{
   char ** strs;
   int x;
   int * y;
};
```

```c
typedef struct A_t A;

A* a[20];
```

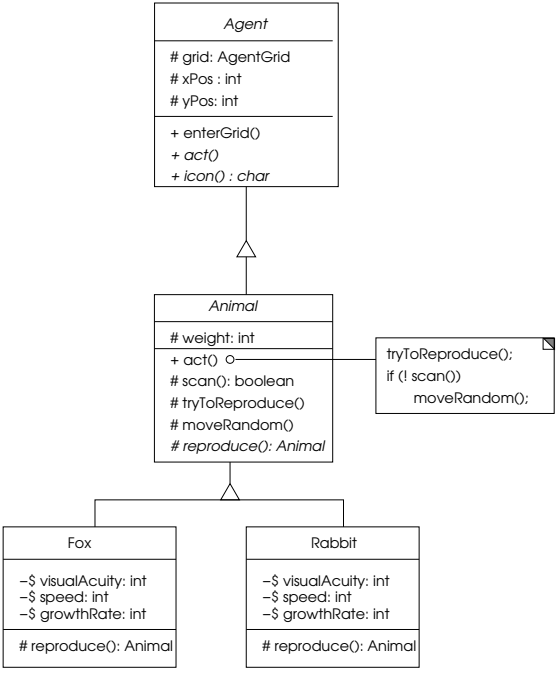
```
a[4]->str[*(a[3]->y = & (** a).x ] [2]
```

Consider a class with the following header:

```
public class Container<T> {
```

Which of the following is something that **cannot** occur inside class Container?

A. An instantiation of `T` ✓
B. A method with return-type `T`
C. An instance variable of type `T`
D. A method with a parameter of type `T`



Problem: Suppose some animal species are are "herd" animals; a set of individuals from this species exist together as a herd, and they cannot move to a space unless it is within three spaces of another member of the herd (not just any other member of the same species; there might be more than one herd of a given species, and an individual must stay by its herd). New individuals become members of their parent's herd.

Other animal species are "pack" animals; a set of individuals from such a species exist together as a pack. Whenever one eats another animal it shares half of the prey's weight with some of the other members of the pack. New individuals become members of their parent's pack.

Modify this design so that new species classes that are herd or pack classes can share code (for example, `Buffalo` and `Antelope` might both be herd classes, so they should share code relevant to herd information or behavior; `Jackal` and `Wolf` might be pack classes and need to share code; there might be some code that both herd and pack animals can share).

Coming up:

- **Due Wed, Mar 25.** *Read Savitch 14.2 (pg 782–799).*
- **Due Thrus, Mar26.** *Read prelab and take quiz*
- **Due Fri, Mar 27.** *Do Project 5, bit vector*