

```
Woo
private Grid g;
private Cazador kats;
private InputStreamReader isr;
private BufferedReader in;
private numPrey; (private)
boolean exit; (private)
private String predators;
private String prey;
private String notifis;
private String rules;
CONSTRUCTOR
Woo (int s): s = sidelength, populate g
METHODS:
newGame () void: starts a new game
int[] ask (): asks for coordinates,
takes in commands
void checkGrid (int xcor, int ycor): goes
to chosen coordinates & checks
int checkStatus (): -1 = lost, 1 = won,
0 = playing, 2 = force exit
HELPERS:
void checkCommands: executes commands
inputted (if valid)
boolean specialCaseBoo: (int xcor, int ycor)
special case:
int checkBox (int xcor, int ycor): presence
of prey/predator/empty
void checkSurround (int[] p, int[] g): looks
in surroundings using checkBox (x,y)
boolean noExit (): checks if the player
has force exited the game.
```

```
Grid
private Animal[][] -grid;
private int sidelength;
private Object[][] -used Cors;
CONSTRUCTOR:
Grid (int s): new grid with sidelength s
METHODS:
String toString (): Prints -grid on screen
String toString (String diag): for diagnostics
int populate (Grid s): populates -grid
void initPopulateUsed (): pop. -used Cors
ACCESSORS
int getSideLength ();
Animal[][] getArray ();
Object getCor (int xcor, int ycor);
Object[][] getUsedCors ();
MUTATORS
Object setUsedCors (Integer xcor,
Integer ycor, Integer a);
```

```
Prey
private int CP;
ArrayList<String> names;
ArrayList<String> susNames;
ArrayList<String> stories;
ArrayList<String> endStories
CONSTRUCTOR
Prey (): random Prey
String toString (): returns name
ACCESSORS
int getCP ();
String getStory ();
String getSusName ();
String getEnd ();
void setSusName (String str);
```

```
Cazador
private int HP;
private int catchPoints;
private int status;
private String nombre;
CONSTRUCTOR
Cazador (String name)
ACCESSORS
int getHP ();
int getCP ();
int getStatust ();
MUTATORS
void setHP (int p)
void setCP (int p)
```

```
Animal
protected int numPreds;
protected int numPreys;
protected String name;
protected String story;
protected String sus Name;
protected String endStory;
abstract String endStory;
abstract String setSusName ();
```

```
INTERFACE
Real Animals
METHODS
String toString ();
String getStory ();
String getSusName ();
String getEnd ();
```

```
Empty
CONSTRUCTOR
public Empty (): sets name to
empty 5-character string
public Empty (int xcor, int ycor)
METHODS
String toString (): returns the
empty string (5spaces)
String getSusName ()
void setSus Name (String str)
```

\* Animal is the superclass  
of Prey, Predator and  
Empty.

```
Predator
private int damage;
ArrayList<String> names;
ArrayList<String> susNames;
ArrayList<String> stories;
ArrayList<String> endStories;
ArrayList<Integer> damages;
CONSTRUCTOR:
Predator (int damageNum): sets damage
String toString (): returns name
ACCESSORS
String getStory ();
int getDamage ();
String getSusName ();
String getEnd ();
void set Sus Name (String str);
```

IMPLEMENTED