

- All things have a getter `getColor` which returns the thing's color (type `Color`). Bushes are green, rabbits are brown, sloths are gray. Since there's not a predefined `Color.BROWN` constant, you'll need `Color.rgb(165, 125, 8)` instead. (This getter can simply return the desired color — since a particular kind of thing is always the same color, there's no need for an instance variable for the color.)
- All animals have a position (row and column) in the field, getters `getRow` and `getColumn`, and a setter `setPosition` which takes the row and column as parameters and sets the position accordingly.
- Sloths have a sleep counter, which keeps track of how much longer the sloth will sleep before it wakes up.
- All animals have two constructors: one which takes the position (row and column) as a parameter and initializes the instance variables accordingly, and another which just initializes the instance variables to -1. The sloth constructors should also initialize the sleep counter to a random value between 0 and 20. (So different sloths will wake up at different times.)
- All things can be drawn. The `draw` method takes a `GraphicsContext` and the position (x and y coordinates of the upper left corner) and dimensions (width and height) of a rectangle as parameters, and draws the thing (in its color) inside the specified rectangular area. (What method can you use to determine the color? — `draw` is similar to `print` from the tickets example from class) Bushes are drawn as rectangles filling the specified area. Animals are drawn as circles just fitting the specified area. Sleeping sloths are drawn like regular animals, plus the text "Zz" is drawn on top of the sloth's circle. (A sloth is sleeping when its sleep counter is greater than 2.)
- All animals can choose which direction they want to move in. The `getNextMove` method takes a `Field` object as a parameter and returns one of the constants defined in `Direction` (which are integers). (Note that this is a desired direction of movement — the animal does not need to check if it is possible to move in that direction, nor should it change its position.) Rabbits choose a random direction. When a sloth is fully awake (i.e. the sleep counter is 0), the sleep counter is set back to 20 and a random direction is returned. Otherwise the sleep counter is decremented and `Direction.NONE` is returned. Note that the `Field` parameter is not actually used by either rabbit or sloth. (It's there in case some other kind of animal might want to use it.)
- All animals have a `reset` method which clears the animal's memory. Rabbits have no memory to reset, so the method body will be empty. A sloth's memory is its sleep counter, which should be reset to a random value between 0 and 20.

- the language tells you where elements go – “all things”, “all animals”, “sloths”

- variables vs methods – variables store information
 - “have a position”
 - “have a sleep counter, which keeps track of ...”

- All things have a getter `getColor` which returns the thing's color (type `Color`). Bushes are green, rabbits are brown, sloths are gray. Since there's not a predefined `Color.BROWN` constant, you'll need `Color.rgb(165, 125, 8)` instead. (This getter can simply return the desired color — since a particular kind of thing is always the same color, there's no need for an instance variable for the color.)
- All animals have a position (row and column) in the field, getters `getRow` and `getColumn`, and a setter `setPosition` which takes the row and column as parameters and sets the position accordingly.
- Sloths have a sleep counter, which keeps track of how much longer the sloth will sleep before it wakes up.
- All animals have two constructors: one which takes the position (row and column) as a parameter and initializes the instance variables accordingly, and another which just initializes the instance variables to -1. The sloth constructors should also initialize the sleep counter to a random value between 0 and 20. (So different sloths will wake up at different times.)
- All things can be drawn. The `draw` method takes a `GraphicsContext` and the position (x and y coordinates of the upper left corner) and dimensions (width and height) of a rectangle as parameters, and draws the thing (in its color) inside the specified rectangular area. (What method can you use to determine the color? — `draw` is similar to `print` from the tickets example from class) Bushes are drawn as rectangles filling the specified area. Animals are drawn as circles just fitting the specified area. Sleeping sloths are drawn like regular animals, plus the text "Zz" is drawn on top of the sloth's circle. (A sloth is sleeping when its sleep counter is greater than 2.)
- All animals can choose which direction they want to move in. The `getNextMove` method takes a `Field` object as a parameter and returns one of the constants defined in `Direction` (which are integers). (Note that this is a desired direction of movement — the animal does not need to check if it is possible to move in that direction, nor should it change its position.) Rabbits choose a random direction. When a sloth is fully awake (i.e. the sleep counter is 0), the sleep counter is set back to 20 and a random direction is returned. Otherwise the sleep counter is decremented and `Direction.NONE` is returned. Note that the `Field` parameter is not actually used by either rabbit or sloth. (It's there in case some other kind of animal might want to use it.)
- All animals have a `reset` method which clears the animal's memory. Rabbits have no memory to reset, so the method body will be empty. A sloth's memory is its sleep counter, which should be reset to a random value between 0 and 20.

- constructors

- if the class extends another, first line of the constructor must call the superclass constructor to build the inside of the onion, then initialize the instance variables in this layer

- All things have a getter `getColor` which returns the thing's color (type `Color`). Bushes are green, rabbits are brown, sloths are gray. Since there's not a predefined `Color.BROWN` constant, you'll need `Color.rgb(165, 125, 8)` instead. (This getter can simply return the desired color — since a particular kind of thing is always the same color, there's no need for an instance variable for the color.)
- All animals have a position (row and column) in the field, getters `getRow` and `getColumn`, and a setter `setPosition` which takes the row and column as parameters and sets the position accordingly.
- Sloths have a sleep counter, which keeps track of how much longer the sloth will sleep before it wakes up.
- All animals have two constructors: one which takes the position (row and column) as a parameter and initializes the instance variables accordingly, and another which just initializes the instance variables to -1. The sloth constructors should also initialize the sleep counter to a random value between 0 and 20. (So different sloths will wake up at different times.)
- All things can be drawn. The `draw` method takes a `GraphicsContext` and the position (x and y coordinates of the upper left corner) and dimensions (width and height) of a rectangle as parameters, and draws the thing (in its color) inside the specified rectangular area. (What method can you use to determine the color? — `draw` is similar to `print` from the tickets example from class) Bushes are drawn as rectangles filling the specified area. Animals are drawn as circles just fitting the specified area. Sleeping sloths are drawn like regular animals, plus the text "Zz" is drawn on top of the sloth's circle. (A sloth is sleeping when its sleep counter is greater than 2.)
- All animals can choose which direction they want to move in. The `getNextMove` method takes a `Field` object as a parameter and returns one of the constants defined in `Direction` (which are integers). (Note that this is a desired direction of movement — the animal does not need to check if it is possible to move in that direction, nor should it change its position.) Rabbits choose a random direction. When a sloth is fully awake (i.e. the sleep counter is 0), the sleep counter is set back to 20 and a random direction is returned. Otherwise the sleep counter is decremented and `Direction.NONE` is returned. Note that the `Field` parameter is not actually used by either rabbit or sloth. (It's there in case some other kind of animal might want to use it.)
- All animals have a `reset` method which clears the animal's memory. Rabbits have no memory to reset, so the method body will be empty. A sloth's memory is its sleep counter, which should be reset to a random value between 0 and 20.

- methods – abstract vs not – is there a body that makes sense?
 - “all things have a getter `getColor` ... bushes are green, rabbits are brown, ...”
 - color depends on the specific kind of thing → abstract
 - “animals are drawn as circles ...” → not abstract
- methods – extending vs overriding
 - “sleeping sloths are drawn like regular animals, plus...” → call superclass draw plus do other stuff

- All things have a getter `getColor` which returns the thing's color (type `Color`). Bushes are green, rabbits are brown, sloths are gray. Since there's not a predefined `Color.BROWN` constant, you'll need `Color.rgb(165, 125, 8)` instead. (This getter can simply return the desired color — since a particular kind of thing is always the same color, there's no need for an instance variable for the color.)
- All animals have a position (row and column) in the field, getters `getRow` and `getColumn`, and a setter `setPosition` which takes the row and column as parameters and sets the position accordingly.
- Sloths have a sleep counter, which keeps track of how much longer the sloth will sleep before it wakes up.
- All animals have two constructors: one which takes the position (row and column) as a parameter and initializes the instance variables accordingly, and another which just initializes the instance variables to -1. The sloth constructors should also initialize the sleep counter to a random value between 0 and 20. (So different sloths will wake up at different times.)
- All things can be drawn. The `draw` method takes a `GraphicsContext` and the position (x and y coordinates of the upper left corner) and dimensions (width and height) of a rectangle as parameters, and draws the thing (in its color) inside the specified rectangular area. (What method can you use to determine the color? — `draw` is similar to `print` from the tickets example from class) Bushes are drawn as rectangles filling the specified area. Animals are drawn as circles just fitting the specified area. Sleeping sloths are drawn like regular animals, plus the text "Zz" is drawn on top of the sloth's circle. (A sloth is sleeping when its sleep counter is greater than 2.)
- All animals can choose which direction they want to move in. The `getNextMove` method takes a `Field` object as a parameter and returns one of the constants defined in `Direction` (which are integers). (Note that this is a desired direction of movement — the animal does not need to check if it is possible to move in that direction, nor should it change its position.) Rabbits choose a random direction. When a sloth is fully awake (i.e. the sleep counter is 0), the sleep counter is set back to 20 and a random direction is returned. Otherwise the sleep counter is decremented and `Direction.NONE` is returned. Note that the `Field` parameter is not actually used by either rabbit or sloth. (It's there in case some other kind of animal might want to use it.)
- All animals have a `reset` method which clears the animal's memory. Rabbits have no memory to reset, so the method body will be empty. A sloth's memory is its sleep counter, which should be reset to a random value between 0 and 20.

- classes – must be abstract if there's at least one abstract method
 - can be abstract if nothing is just that, only a kind of that