Learning Goals
• Practice inheritance
• Have fun simulating critters!
Critters!

hares

foxes
import turtle
import math
import random

#========================================
# globals
RADIUS = 300
NUM_FOXES = 6
NUM_HARES = 10

#========================================
# functions

def random_position():
    """Sample uniformly in circle""
    r = RADIUS * math.sqrt(random.uniform(0,1))
    theta = random.uniform(0, 2*math.pi)
    x = r * math.cos(theta)
    y = r * math.sin(theta)
    return x, y

def random_heading():
    """Sample uniformly""
    return random.uniform(0, 360)  # degrees

#========================================
# classes
class Critter(turtle.Turtle):
    """A Critter"
    code here...

class Fox(Critter):
    """A Fox (fast darkorange square)"
    code here...

class Hare(Critter):
    """A Hare (slow tan circle)"
    code here...

#========================================
# main
def main():
    
    # initialize field
    turtle.dot(2*RADIUS, "darkgreen")

    # initialize critters
    critters = []
    for i in range(NUM_FOXES):
        critters.append(Fox("Fox " + str(i)))
    for j in range(NUM_HARES):
        critters.append(Hare("Hare " + str(j)))

    # move critters indefinitely
    while True:
        for critter in critters:
            critter.move()
class Critter(turtle.Turtle):
    """A Critter""

    def __init__(self, name):
        turtle.Turtle.__init__(self)
        # set name

        # set random position and heading
        self.penup()

    def move(self):
        """Move the critter""

        # move forward one step
        if self.distance(0,0) >= RADIUS:
            # invalid, go back, pick new random heading

Turtle Methods you may find useful
- def speed(speed) – set speed
- def shape(name) – set shape
- def color(name) – set color
- def setposition(position) – set position
- def setheading(heading) – set heading
- def forward(distance) – move forward by distance

Critter Specifications
- A Critter has a name.
- On a move, a Critter moves forward by speed. If the resulting location is invalid, a Critter turns around, moves back, and picks a new random heading.
- A Fox moves fast (speed = 5), is “darkorange” and “square”
- A Hare moves slow (speed = 1), is “tan” and “circle”

Complete according to specifications.
Demo critters.py!
class Critter(turtle.Turtle):
    """A Critter""

def __init__(self, name):
    turtle.Turtle.__init__(self)
    self.name = name  # set name

    # set random position and heading
    self.penup()
    self.setposition(random_position())
    self.setheading(random_heading())

def move(self):
    """Move the critter""
    # move forward one step
    self.forward(self.speed())
    if self.distance(0,0) >= RADIUS:
        # invalid, go back, pick new random heading
        self.right(180)
        self.forward(self.speed())
        self.setheading(random_heading())

class Fox(Critter):
    """A Fox (fast dark orange square)""

def __init__(self, name):
    Critter.__init__(self, name)
    self.speed(5)
    self.shape("square")
    self.color("darkorange")

class Hare(Critter):
    """A Hare (slow tan circle)""

def __init__(self, name):
    Critter.__init__(self, name)
    self.speed(1)
    self.shape("circle")
    self.color("tan")

Turtle Methods you may find useful
• def speed(speed) – set speed
• def shape(name) – set shape
• def color(name) – set color
• def setposition(position) – set position
• def setheading(heading) – set heading
• def forward(distance) – move forward by distance

Critter Specifications
• A Critter has a name.
• On a move, a Critter moves forward by speed. If the resulting location is invalid, a Critter turns around, moves back, and picks a new random heading.
• A Fox moves fast (speed = 5), is "darkorange" and "square"
• A Hare moves slow (speed = 1), is "tan" and "circle"

Complete according to specifications. Demo critters.py!
What’s coming next…

• Tuesday: Projects showcase!
• After break
  – Work on your project (milestone + final project)
  – Labs are just for working and getting help on projects
• Class material: The limits of computation!