TP1

1 -

a)

play
addAmount
hasWings
compare
isGreater

b)

Não
Não
Sim, retorna "true" ou "false"
Não
Sim, retorna "true" ou "false"

c)

Nenhum
Um
Nenhum
Três
Um

2 - 13

3 - 5

4 - 1

5 -

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * A cat. Can do some things cats do. Or not.
 *
 * @author Michael Kölling
 * @version 1.0
 */
public class Cat  extends Actor
{
    private boolean tired = false;
    private boolean hungry = false;
    private boolean bored = true;

    /**
     * Walk a bit to the left. 'distance' determines how far to walk. Use small
numbers (1 to 10).
```

```
     */
    public void walkLeft(int distance)
    {
        walk(distance, -10, "cat-walk.png", "cat-walk-2.png");
    }

    /**
     * Walk a bit to the right. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkRight(int distance)
    {
        walk(distance, 10, "cat-walk-right.png", "cat-walk-right-2.png");
    }

    /**
     * Internal walk method. Walk a given distance into a given direction, using
given images.
     */
    private void walk(int distance, int direction, String img1, String img2)
    {
        for (int i=0; i<distance; i++)
        {
            setImage(img1);
            wait(4);
            setLocation(getX() + direction, getY());
            setImage(img2);
            wait(4);
            setLocation(getX() + direction, getY());
        }
        setImage("cat.png");
    }

    /**
     * Do a dance. Cool, Baby!
     * (Dancing makes you tired.)
     */
    public void dance()
    {
        Greenfoot.playSound("music.wav");
        for (int i=0; i<2; i++)
        {
            setImage("cat-dance.png");
            wait(10);
            setImage("cat.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(8);
            setImage("cat.png");
            wait(8);
        }
        for (int i=0; i<5; i++)
```

```
        {
            setImage("cat-dance.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(6);
        }
        setImage("cat.png");
        tired = true;
        hungry = true;
        bored = false;
    }

    /**
     * It's really what the method name says: shout "Hooray".
     */
    public void shoutHooray()
    {
        setImage("cat-speak.png");
        Greenfoot.playSound("hooray.wav");
        wait(20);
        setImage("cat.png");
        bored = false;
    }

    /**
     * Sleep for a while. The parameter determines how long to sleep. Use small
numbers.
     * A value of 1 will sleep for a couple of seconds or so.
     */
    public void sleep(int howLong)
    {
        for (int i=0; i<howLong; i++)
        {
            for (int j=1; j<=4; j++)
            {
                setImage("cat-sleep-" + j + ".png");
                wait(10);
            }
        }
        setImage("cat.png");
        tired = false;
        bored = true;
    }

        /**
         * Eat some pizza!
         */
        public void eat()
        {
        for (int i=0; i<4; i++)
        {
            setImage("cat-eat.png");
```

```
        wait(8);
        setImage("cat-eat-2.png");
        wait(6);
    }
    setImage("cat.png");
    tired = true;
    hungry = false;
}

/**
 * Return true if the cat is alone here.
 */
public boolean isAlone()
{
    int numberOfCats = getWorld().getObjects(Cat.class).size();
    return numberOfCats < 2;
}

/**
 * Return true if the cat is not alone here.
 */
public boolean hasCompany()
{
    return !isAlone();
}

/**
 * Return true if the cat is hungy.
 */
public boolean isHungry()
{
    return hungry;
}

/**
 * Return true if the cat is sleepy.
 */
public boolean isSleepy()
{
    return tired;
}

/**
 * Return true if the cat is bored.
 */
public boolean isBored()
{
    return bored;
}

/**
 * Wait for a given time.
```

```
     */
    public void wait(int time)
    {
        Greenfoot.delay(time);
    }

    public void act()
    {
        eat();
    }
}
```

6 -

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * A cat. Can do some things cats do. Or not.
 *
 * @author Michael Kölling
 * @version 1.0
 */
public class Cat  extends Actor
{
    private boolean tired = false;
    private boolean hungry = false;
    private boolean bored = true;

    /**
     * Walk a bit to the left. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkLeft(int distance)
    {
        walk(distance, -10, "cat-walk.png", "cat-walk-2.png");
    }

    /**
     * Walk a bit to the right. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkRight(int distance)
    {
        walk(distance, 10, "cat-walk-right.png", "cat-walk-right-2.png");
    }

    /**
     * Internal walk method. Walk a given distance into a given direction, using
given images.
     */
    private void walk(int distance, int direction, String img1, String img2)
    {
```

```
    for (int i=0; i<distance; i++)
    {
        setImage(img1);
        wait(4);
        setLocation(getX() + direction, getY());
        setImage(img2);
        wait(4);
        setLocation(getX() + direction, getY());
    }
    setImage("cat.png");
}

/**
 * Do a dance. Cool, Baby!
 * (Dancing makes you tired.)
 */
public void dance()
{
    Greenfoot.playSound("music.wav");
    for (int i=0; i<2; i++)
    {
        setImage("cat-dance.png");
        wait(10);
        setImage("cat.png");
        wait(8);
        setImage("cat-dance-2.png");
        wait(8);
        setImage("cat.png");
        wait(8);
    }
    for (int i=0; i<5; i++)
    {
        setImage("cat-dance.png");
        wait(8);
        setImage("cat-dance-2.png");
        wait(6);
    }
    setImage("cat.png");
    tired = true;
    hungry = true;
    bored = false;
}

/**
 * It's really what the method name says: shout "Hooray".
 */
public void shoutHooray()
{
    setImage("cat-speak.png");
    Greenfoot.playSound("hooray.wav");
    wait(20);
    setImage("cat.png");
```

```
        bored = false;
    }

    /**
     * Sleep for a while. The parameter determines how long to sleep. Use small
numbers.
     * A value of 1 will sleep for a couple of seconds or so.
     */
    public void sleep(int howLong)
    {
        for (int i=0; i<howLong; i++)
        {
            for (int j=1; j<=4; j++)
            {
                setImage("cat-sleep-" + j + ".png");
                wait(10);
            }
        }
        setImage("cat.png");
        tired = false;
        bored = true;
    }

        /**
         * Eat some pizza!
         */
        public void eat()
        {
        for (int i=0; i<4; i++)
        {
            setImage("cat-eat.png");
            wait(8);
            setImage("cat-eat-2.png");
            wait(6);
        }
        setImage("cat.png");
        tired = true;
        hungry = false;
    }

    /**
     * Return true if the cat is alone here.
     */
    public boolean isAlone()
    {
        int numberOfCats = getWorld().getObjects(Cat.class).size();
        return numberOfCats < 2;
    }

    /**
     * Return true if the cat is not alone here.
     */
```

```
    public boolean hasCompany()
    {
        return !isAlone();
    }

    /**
     * Return true if the cat is hungy.
     */
    public boolean isHungry()
    {
        return hungry;
    }

    /**
     * Return true if the cat is sleepy.
     */
    public boolean isSleepy()
    {
        return tired;
    }

    /**
     * Return true if the cat is bored.
     */
    public boolean isBored()
    {
        return bored;
    }

    /**
     * Wait for a given time.
     */
    public void wait(int time)
    {
        Greenfoot.delay(time);
    }

    public void act()
    {
        dance();
    }
}
```

7 -

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * A cat. Can do some things cats do. Or not.
 *
 * @author Michael Kölling
 * @version 1.0
```

```
 */
public class Cat  extends Actor
{
    private boolean tired = false;
    private boolean hungry = false;
    private boolean bored = true;

    /**
     * Walk a bit to the left. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkLeft(int distance)
    {
        walk(distance, -10, "cat-walk.png", "cat-walk-2.png");
    }

    /**
     * Walk a bit to the right. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkRight(int distance)
    {
        walk(distance, 10, "cat-walk-right.png", "cat-walk-right-2.png");
    }

    /**
     * Internal walk method. Walk a given distance into a given direction, using
given images.
     */
    private void walk(int distance, int direction, String img1, String img2)
    {
        for (int i=0; i<distance; i++)
        {
            setImage(img1);
            wait(4);
            setLocation(getX() + direction, getY());
            setImage(img2);
            wait(4);
            setLocation(getX() + direction, getY());
        }
        setImage("cat.png");
    }

    /**
     * Do a dance. Cool, Baby!
     * (Dancing makes you tired.)
     */
    public void dance()
    {
        Greenfoot.playSound("music.wav");
        for (int i=0; i<2; i++)
        {
```

```
            setImage("cat-dance.png");
            wait(10);
            setImage("cat.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(8);
            setImage("cat.png");
            wait(8);
        }
        for (int i=0; i<5; i++)
        {
            setImage("cat-dance.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(6);
        }
        setImage("cat.png");
        tired = true;
        hungry = true;
        bored = false;
    }

    /**
     * It's really what the method name says: shout "Hooray".
     */
    public void shoutHooray()
    {
        setImage("cat-speak.png");
        Greenfoot.playSound("hooray.wav");
        wait(20);
        setImage("cat.png");
        bored = false;
    }

    /**
     * Sleep for a while. The parameter determines how long to sleep. Use small
numbers.
     * A value of 1 will sleep for a couple of seconds or so.
     */
    public void sleep(int howLong)
    {
        for (int i=0; i<howLong; i++)
        {
            for (int j=1; j<=4; j++)
            {
                setImage("cat-sleep-" + j + ".png");
                wait(10);
            }
        }
        setImage("cat.png");
        tired = false;
        bored = true;
```

```
}

    /**
     * Eat some pizza!
     */
    public void eat()
    {
    for (int i=0; i<4; i++)
    {
        setImage("cat-eat.png");
        wait(8);
        setImage("cat-eat-2.png");
        wait(6);
    }
    setImage("cat.png");
    tired = true;
    hungry = false;
}

/**
 * Return true if the cat is alone here.
 */
public boolean isAlone()
{
    int numberOfCats = getWorld().getObjects(Cat.class).size();
    return numberOfCats < 2;
}

/**
 * Return true if the cat is not alone here.
 */
public boolean hasCompany()
{
    return !isAlone();
}

/**
 * Return true if the cat is hungy.
 */
public boolean isHungry()
{
    return hungry;
}

/**
 * Return true if the cat is sleepy.
 */
public boolean isSleepy()
{
    return tired;
}
```

```
    /**
     * Return true if the cat is bored.
     */
    public boolean isBored()
    {
        return bored;
    }

    /**
     * Wait for a given time.
     */
    public void wait(int time)
    {
        Greenfoot.delay(time);
    }

    public void act(int howLong)
    {
        sleep(howLong);
    }
}
```

8 -

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * A cat. Can do some things cats do. Or not.
 *
 * @author Michael Kölling
 * @version 1.0
 */
public class Cat  extends Actor
{
    private boolean tired = false;
    private boolean hungry = false;
    private boolean bored = true;

    /**
     * Walk a bit to the left. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkLeft(int distance)
    {
        walk(distance, -10, "cat-walk.png", "cat-walk-2.png");
    }

    /**
     * Walk a bit to the right. 'distance' determines how far to walk. Use small
numbers (1 to 10).
     */
    public void walkRight(int distance)
```

```
    {
        walk(distance, 10, "cat-walk-right.png", "cat-walk-right-2.png");
    }

    /**
     * Internal walk method. Walk a given distance into a given direction, using
given images.
     */
    private void walk(int distance, int direction, String img1, String img2)
    {
        for (int i=0; i<distance; i++)
        {
            setImage(img1);
            wait(4);
            setLocation(getX() + direction, getY());
            setImage(img2);
            wait(4);
            setLocation(getX() + direction, getY());
        }
        setImage("cat.png");
    }

    /**
     * Do a dance. Cool, Baby!
     * (Dancing makes you tired.)
     */
    public void dance()
    {
        Greenfoot.playSound("music.wav");
        for (int i=0; i<2; i++)
        {
            setImage("cat-dance.png");
            wait(10);
            setImage("cat.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(8);
            setImage("cat.png");
            wait(8);
        }
        for (int i=0; i<5; i++)
        {
            setImage("cat-dance.png");
            wait(8);
            setImage("cat-dance-2.png");
            wait(6);
        }
        setImage("cat.png");
        tired = true;
        hungry = true;
        bored = false;
    }
```

```
    /**
     * It's really what the method name says: shout "Hooray".
     */
    public void shoutHooray()
    {
        setImage("cat-speak.png");
        Greenfoot.playSound("hooray.wav");
        wait(20);
        setImage("cat.png");
        bored = false;
    }

    /**
     * Sleep for a while. The parameter determines how long to sleep. Use small
numbers.
     * A value of 1 will sleep for a couple of seconds or so.
     */
    public void sleep(int howLong)
    {
        for (int i=0; i<howLong; i++)
        {
            for (int j=1; j<=4; j++)
            {
                setImage("cat-sleep-" + j + ".png");
                wait(10);
            }
        }
        setImage("cat.png");
        tired = false;
        bored = true;
    }

        /**
         * Eat some pizza!
         */
        public void eat()
        {
        for (int i=0; i<4; i++)
        {
            setImage("cat-eat.png");
            wait(8);
            setImage("cat-eat-2.png");
            wait(6);
        }
        setImage("cat.png");
        tired = true;
        hungry = false;
    }

    /**
     * Return true if the cat is alone here.
```

```
 */
public boolean isAlone()
{
    int numberOfCats = getWorld().getObjects(Cat.class).size();
    return numberOfCats < 2;
}

/**
 * Return true if the cat is not alone here.
 */
public boolean hasCompany()
{
    return !isAlone();
}

/**
 * Return true if the cat is hungy.
 */
public boolean isHungry()
{
    return hungry;
}

/**
 * Return true if the cat is sleepy.
 */
public boolean isSleepy()
{
    return tired;
}

/**
 * Return true if the cat is bored.
 */
public boolean isBored()
{
    return bored;
}

/**
 * Wait for a given time.
 */
public void wait(int time)
{
    Greenfoot.delay(time);
}

public void act(int howLong, int x, int y, int z)
{
    // sleep = 0
    // eat = 1
    //dance = 2
```

```
        if (x == 0)
        {
            sleep(howLong);
        }
        if (x == 1)
        {
            eat();
        }
        if (x == 2)
        {
            dance();
        }
        if (y == 0)
        {
            sleep(howLong);
        }
        if (y == 1)
        {
            eat();
        }
        if (y == 2)
        {
            dance();
        }
        if (z == 0)
        {
            sleep(howLong);
        }
        if (z == 1)
        {
            eat();
        }
        if (z == 2)
        {
            dance();
        }
    }
}
```

9 -

```
import greenfoot.*;  // (World, Actor, GreenfootImage, Greenfoot and MouseInfo)

/**
 * MyCat is your own cat. Get it to do things by writing code in its act method.
 *
 * @author (your name)
 * @version (a version number or a date)
 */
public class MyCat extends Cat
{
    /**
```

```
     * Act - do whatever the MyCat wants to do.
     */
    public void act(int howLong)
    {

        if(isSleepy()){
            sleep(howLong);
            walkRight(5);
        }
        if(isBored()){
            dance();
            walkRight(5);
        }
        if(isHungry()){
            eat();
            walkRight(5);
        }
        if(isAtEdge()){
            shoutHooray();
            walkLeft(5);
        }

    }
}
```

NOME: Bernardo Mendes de Lima