Igor Ribeiro - 140492
POO- Lista 2

Respostas dissertativas

2) A classe iria passar a poder ser instanciada e os métodos abstratos (sem implementação), dariam erro justamente por não terem implementações. Neste caso em específico, teríamos que implementar o método public abstract void move(int passos); e ver se não teria quaisquer outros problemas.

3) Não podemos instanciar classes abstratas pois elas são, essencialmente, a base para as subclasses, que implementam e completam, especificamente, os métodos.

Respostas em código

1)

```
public class RoboABateria extends RoboAbstrato{
    private long energia;
    RoboABateria(String n, int px, int py, short d, long e) {
    public void move(int passos) {
    public String toString() {

    public void recarrega(int qtdeEnergia) {
        this.energia += qtdeEnergia;
    }
}
```

4)

```
public class RoboSimples extends RoboAbstrato {
    RoboSimples(String n, int px, int py, short d) {
    public void move(int passos) {

    public void mudaDirecao(short novaDir) {
        if(novaDir < 45 || novaDir > 315) {
            super.mudaDirecao((short) 0);
        }else if(novaDir >=45 || novaDir <=135) {
            super.mudaDirecao((short) 90);
        }else if(novaDir >=135 || novaDir <=225) {
            super.mudaDirecao((short) 180);
        }else if(novaDir >=225 || novaDir <=315) {
            super.mudaDirecao((short) 270);
        }

    }
}
```

5)

```
public class RoboABateria extends RoboAbstrato{
    private long energia;
    RoboABateria(String n, int px, int py, short d, long e) {
    public void move(int passos) {
        long energiaASerGasta;
        if(direcaoAtual() == 45 || direcaoAtual() == 135 || direcaoAtual() == 225 || direcaoAtual() == 315) {
            energiaASerGasta = passos * 14;
        }else {
            energiaASerGasta = passos * 10;
            }

        if (energiaASerGasta <= energia) {
            switch (direcaoAtual()) {
                case 0:
                    moveX(+passos);
                    break;
                case 45:
                    moveX(+passos);
                    moveY(+passos);
                    break;
                case 90:
                    moveY(+passos);
```

6)

```
package lista2;

public class RoboComMemoria extends RoboAbstrato {
    private int passosX, passosY;
    RoboComMemoria(String n, int px, int py, short d) {

    @Override
    public void move(int passos) {
        switch (direcaoAtual()) {
        case 0:
            passosX+=passos;
            moveX(+passos);
            break;
        case 90:
            passosY+=passos;
            moveY(+passos);
            break;
        case 180:
            passosX-=passos;
            moveX(-passos);
            break;
        case 270:
            passosY-=passos;
            moveY(-passos);
            break;
        }
    }
    public void retornaAOrigem() {
        moveX(-passosX);
        moveY(-passosY);
    }

}
```

7)

```java
public class RoboPesadoABateria extends RoboABateria {
    private double peso;
    RoboPesadoABateria(String n, int px, int py, short d, long e) {
        super(n, px, py, d, e);
        // TODO Auto-generated constructor stub
    }

    public void move(int passos) {
        long energiaASerGasta;
        if(direcaoAtual() == 45 || direcaoAtual() == 135 || direcaoAtual() == 225 || direcaoAtual() == 315) {
            energiaASerGasta = (long) (passos * 1.4 * peso);
        }else {
            energiaASerGasta = (long) (passos * peso);
            }

        if (energiaASerGasta <= energia) {
            switch (direcaoAtual()) {
                case 0:
                    moveX(+passos);
                    break;
                case 45:
                    moveX(+passos);
                    moveY(+passos);
                    break;
                case 90:
                    moveY(+passos);
                    break;
                case 135:
                    moveY(+passos);
                    moveX(-passos);
                    break;
                case 180:
                    moveX(-passos);
                    break;
                case 225:
                    moveX(-passos);
                    moveY(-passos);
```