



Você deve fazer um sistema que simule um Tamagotchi, controlando algumas ações definidas a seguir. Assim sendo, você mostrará na tela informações do que está acontecendo com o animalzinho digital e solicitará ao usuário que tome decisões de acordo com o que acontece com o animal no decorrer de sua vida.

Um Tamagotchi tem um nome, uma idade e um peso. Ele tem a capacidade de comer, dormir e se exercitar. Neste sentido, ele pode:

- **Sentir sono:** quando o Tamagotchi sentir sono, duas opções devem ser dadas ao usuário:
 - Dormir
 - Permanecer acordado

Quando o Tamagotchi avisa que está com sono e permanece acordado 5 vezes seguidas, a próxima vez que ele sentir sono ele dorme automaticamente. Sempre que o Tamagotchi dorme ele aumenta sua idade em 1 dia. Quando o Tamagotchi chega a 15 dias, ele morre.

- **Sentir fome:** quando o Tamagotchi sentir fome, algumas opções devem ser dadas ao usuário. São elas:
 - Comer muito
 - Comer pouco
 - Não comer

Comer pouco é o normal do Tamagotchi, ele aumenta 1 quilo a cada vez que come pouco. Um Tamagotchi que come muito aumenta 5 quilos, e logo deve dormir. Um Tamagotchi que sente fome e não come emagrece 2 quilos a cada vez. Se o Tamagotchi ultrapassar os 20 quilos ele explode. Se o Tamagotchi chegar a zero quilos, ele fica desnutrido e morre.

- **Ficar entediado:** quando o Tamagotchi fica entediado, ele deve se exercitar. Neste caso, devem ser apresentadas duas opções de treino para o Tamagotchi:
 - Correr 10 minutos

- Caminhar 10 minutos

Depois de correr por 10 minutos, o Tamagotchi automaticamente emagrece 4 quilos e come muito. Depois de caminhar, o Tamagotchi sempre emagrece 1 quilo e fica com fome.

Como escolher os desejos do Tamagotchi?

Os desejos do Tamagotchi (sentir sono, sentir fome e ficar entediado) acontecem aleatoriamente. Use o método `Math.random()` para realizar um sorteio entre os valores desejados. Pesquise o funcionamento do método `Math.random()` e utilize no seu trabalho da forma que achar pertinente.

Seu programa deve simular a vida toda do Tamagotchi, que começa com 0 dias e 1 quilo. Mostre, a cada ação realizada, o estado do Tamagotchi.

Seu programa deve ter, no mínimo, as seguintes classes: Tamagotchi, Principal e Teclado.

- *classe Tamagotchi*: possui todas as informações (atributos) e métodos necessários para um Tamagotchi, como construtores, métodos de acesso e métodos auxiliares para o desenvolvimento do trabalho;
- *classe Principal*: possui o método `main`. No método `main`, deve ser inserido o código que faz a lógica para que o sistema funcione da forma descrita;
- *classe Teclado*: possui métodos para viabilizar a entrada de dados do usuário pelo teclado, para interação com o usuário. Esta classe está pronta e será disponibilizada pelo professor.

Fique atento aos seguintes itens:

1. *Comente o seu código*, informando no mínimo o que cada método de suas classes faz. Não esqueça de colocar seu nome em **TODAS** as classes que você for implementar (isto **exclui** a classe Teclado).
2. *Idente seu código*. Códigos não identados são mais difíceis de entender, difíceis de encontrar erros, além de receberem desconto na nota final. Pesquise sobre indentação.
3. *Utilize impressões na tela* para informar o que está acontecendo no programa
4. *Faça seu código com clareza*.

Outras informações:

- O trabalho deverá ser implementado utilizando o BlueJ. Para utilizar outra IDE, fale **ANTES** com o professor, para verificar a possibilidade ou não
- Deve ser realizado **individualmente**
- Você deverá entregar uma pasta compactada com todos os arquivos do projeto criado no BlueJ
- Trabalhos copiados de colegas e/ou não realizados pelo aluno receberão zero (todos os envolvidos)
- Identifique-se claramente no nome do projeto do BlueJ (nome completo)

Avaliação:

- Antes de mais nada:

```
if(código não compila){
    System.out.println("Sem avaliação...");
    nota = zero;
}else
    System.out.println("O que você fez será avaliado!");
```

- A avaliação será realizada da seguinte forma:
 - *Código comentado*: desconto de 1,0 ponto por classe não comentada
 - *Código indentado*: desconto de 1,0 ponto por classe não indentada
 - *Classes conforme enunciado*: 1,5 pontos
 - *Interface com o usuário (mensagens e criatividade)*: 2,5 pontos
 - *Corretude do programa (lógica, ações, mensagens, resultados)*: 6,0 pontos

O que devo entregar?

Juntamente com sua implementação, você deve enviar um vídeo de 3 minutos no máximo, fazendo uma execução do seu programa, realizando as operações implementadas e narrando o que foi feito e o resultado. Reforçando: MÁXIMO 3 MINUTOS! Faça uma captura de tela narrada, é bem simples 😊

Você deve entregar um arquivo compactado contendo os arquivos do seu projeto (não apenas os fontes, entregue todo o projeto) e o vídeo.

Atenção!

Trabalhos copiados da internet ou de outros colegas levarão nota zero automaticamente (todos os envolvidos). Além disso, trabalhos entregues sem o

vídeo solicitado ou com o vídeo de mais de 3 minutos terão desconto de 50% na nota final da atividade.

Boa sorte!

Dicas:

- Comece pelas classes mais simples. Não tente fazer o mais difícil primeiro.
- Faça um trabalho simples mas correto. Não é necessário criar um programa com centenas de linhas de código. Mais vale um programa que funciona e é simples do que um que é demasiadamente complexo e apresenta falhas.
- Implemente e vá testando suas classes. Não deixe para compilar e verificar se funcionam no final de toda a implementação. Se existirem erros, serão mais difíceis de serem encontrados.
- Pense no programa antes de começar a implementar. Comece a codificar somente quando entender exatamente como ele deve funcionar.
- Não invente regras. Elas estão todas claras no decorrer deste documento.
- Não se assuste com a primeira impressão do trabalho. Você tem plenos conhecimentos e total capacidade de implementá-lo. Basta calma e atenção.
- Não deixe para fazer nos últimos dias.
- Reserve um tempo por dia para pensar no trabalho e implementar alguma parte.