```java
package club.westcs.NifflerCeciliaB;

import java.util.Random;
import java.util.Scanner;

public class Niffler {

    //Attributes
        /*
         * Strings
         *      petName- the name the user gives to the pet
         *      masterName- the name of the user
         * Objects
         *      Scanner- get info from user
         *      Random- randomize the start statistics
         *      A Dice object for every trick that is not core
         *          core(eat, sleep, restroom)
         * Boolean
         *      Alive- has the pet died?
         * Int
         *      Sleep
         *      Eat
         *      Restroom
         * String []
         *      Commands- the tricks the pet can do
         *              -must have 11 commands
         *              - must include eat, sleep, restroom, and test death
         */
    private String petName;
    private String masterName;
    private Scanner scan;
    private Random rand;
    private boolean alive;
    private int sleep;
    private int eat;
    private int restroom;
    private int happiness;
    private int gold;
```

```
    private String pronoun;
    private String[] choices = {"give gold", "run", "steal", "help", "eat", "restroom", "sleep", "test death", "play", "go into the suitcase", "buy", "nothing"};
    private Dice giveGoldDice, runDice, stealDice, helpDice, playDice, goIntoTheSuitcaseDice, buyDice;
    private Stopwatch lifeTime;
    private Stopwatch restroomTime;
    private Stopwatch happyTime;
    //Constructor
        /*
         * initialize your attributes
         * don't forget to do the objects first
         * set the core statistics to a random start value
         */
    public Niffler() {
        rand = new Random();
        scan = new Scanner(System.in);
        alive = true;
        petName = getPetName();
        masterName = getMasterName();
        sleep = 100;
        eat = 100;
        restroom = 100;
        happiness = 100;
        gold = rand.nextInt(21) + 20;
        pronoun = setPronoun();
        printStats();
        giveGoldDice = new Dice();
        runDice = new Dice();
        stealDice = new Dice();
        helpDice = new Dice();
        playDice = new Dice();
        goIntoTheSuitcaseDice = new Dice();
        buyDice = new Dice();
        lifeTime = new Stopwatch();
        restroomTime = new Stopwatch();
        happyTime = new Stopwatch();

    }

    //Methods
        /*
         * #1 live()
         *      -while the pet is alive
         *          -call getCommand() doCommand() setAlive()
         * #2 getCommand()      (See the Norse God)
         *          -call printCommands()   (print every String in the commands array)      (Array Notes)
         *          -ask the user what command they want the pet to do
         *          -return the command the user types
         * #3 doCommand(String command)     (See Norse God)
         *              -run a method for the command that the user has called
         *              -call loseStats()
         * #4 printCommands()          (Array Notes)
         *          -Print all the commands in the command[]
         * #5 command for every trick
         *          -print that the pet did the trick
         * #6 setAlive()
         *          -if an core statistic falls at or below 0
         *              -set alive to false
         *          -else if any core statistic is close to 0
         *              -print a warning for that statistic
         *          -else
         *              -print that the pet is doing great
         * #7 setName()
         * #8 setMasterName()
         * #9 reset()
         *      -if the pet has died
         *          -Ask if they want a new pet
         *          -if they say yes
         *              -reset alive, petName, and all core statistics
         * #10 printStats()
         *          -print the core statistics
         * #11 loseStats()
         *          -subtract from each statistic
         */
```

```
113     public void printChoices() {
114         for(int i = 0; i < this.choices.length; i++) {  //array.length returns the number of items in an array
115             System.out.println("#" + (i+1) + ": " +  this.choices[i]); //array[index] gets the item at a given index of an array
116         }
117     }
118 
119     public Stopwatch startHappyTime() {
120         this.happyTime.start();
121         return this.happyTime;
122     }
123 
124     public Stopwatch startLifeTime() {
125         this.lifeTime.start();
126         overflow();
127         return this.lifeTime;
128     }
129 
130     public Stopwatch startRestroomTime() {
131         this.restroomTime.start();
132         return this.restroomTime;
133     }
134 
135     public String getPetName() {
136         System.out.println("What do you want to name your pet Niffler?");
137         petName = scan.nextLine();
138         return this.petName;
139     }
140 
141     public String getMasterName() {
142         System.out.println("What is your name?");
143         this.masterName = scan.nextLine();
144         getPronoun();
145         return this.masterName;
146     }
147 
```

```
148     public void getPronoun() {
149         System.out.println("Are you a girl or a boy?");
150         pronoun = scan.nextLine().toLowerCase();
151         setPronoun();
152     }
153 
154     private String setPronoun() {
155         if(pronoun.equals("girl") || pronoun.equals("Girl")) {
156             this.pronoun = ("her");
157         }
158         else if(pronoun.equals("boy") || pronoun.equals("Boy")){
159             this.pronoun = ("him");
160         }
161         return this.pronoun;
162     }
163 
164     private void printStats() {
165         System.out.println("Eat:" + this.eat);
166         System.out.println("Sleep:" + this.sleep);
167         System.out.println("Restroom:" + this.restroom);
168         System.out.println("Happiness:" +this.happiness);
169         System.out.println("Gold:" + this.gold);
170     }
171 
172     public boolean isAlive() {
173         return this.alive;
174     }
175 
176     private String choose() {
177         String myChoice = "";
178         printChoices();
179         System.out.println("What should " + this.petName + " do?");
180         myChoice = scan.nextLine().toLowerCase();
181         return myChoice;
182     }
183 
```

```
184     public void doChoice(String myChoice) {
185         if(myChoice.equals("give gold")|| myChoice.equals("1")) {
186             if(this.giveGoldDice.roll()) {
187                 System.out.println(this.masterName + " tells " + this.petName + " to get gold for " + this.pronoun + ".");
188                 this.gold = gold + 5;
189                 printStats();
190                 overflow();
191                 choose();
192             }
193             else {
194                 System.out.println(this.petName + " did not give gold, and has " + this.giveGoldDice.getProb() + " chances to getting it next time.");
195                 this.gold = gold + 0;
196                 printStats();
197                 overflow();
198                 choose();
199             }
200         }
201         else if(myChoice.equals("run")|| myChoice.equals("2")) {
202             if(this.runDice.roll()) {
203                 System.out.println(this.masterName + " tells " + this.petName + " to run for " + this.pronoun + ".");
204                 this.happiness = happiness + 100;
205                 printStats();
206                 overflow();
207                 choose();
208             }
209             else {
210                 System.out.println(this.petName + " did not run, and has " + this.runDice.getProb() + " chances to getting it next time.");
211                 this.happiness = happiness + 50;
212                 printStats();
213                 overflow();
214                 choose();
215             }
216         }
217 
```

```
        else if(myChoice.equals("steal") || myChoice.equals("3")) {
            if(this.stealDice.roll()) {
                System.out.println(this.masterName + " tells " + this.petName + " to steal gold for " + this.pronoun + ".");
                this.gold = gold + 7;
                printStats();
                overflow();
                choose();
            }
            else {
                System.out.println(this.petName + " did not steal, and has " + this.stealDice.getProb() + " chances to getting it next time.");
                this.gold = gold + 0;
                printStats();
                overflow();
                choose();
            }
        }
        else if(myChoice.equals("help")|| myChoice.equals("4")) {
            if(this.helpDice.roll()) {
                System.out.println(this.masterName + " tells " + this.petName + " to help others on behalf of " + this.pronoun + ".");
                this.happiness = happiness + 100;
                printStats();
                overflow();
                choose();
            }
            else {
                System.out.println(this.petName + " did not help, and has " + this.helpDice.getProb() + " chances to getting it next time.");
                this.happiness = happiness + 50;
                printStats();
                overflow();
                choose();
            }
        }
        else if(myChoice.equals("eat") || myChoice.equals("5")) {
            System.out.println(this.masterName + " tells " + this.petName + " to eat.");
            this.eat = eat + 100;
            printStats();
            overflow();
            choose();
        }
        else if(myChoice.equals("restroom")|| myChoice.equals("6")) {
            System.out.println(this.masterName + " tells " + this.petName + " to go the restroom.");
            this.restroom = restroom + 100;
            printStats();
            overflow();
            choose();
        }
        else if(myChoice.equals("sleep")|| myChoice.equals("7")) {
            System.out.println(this.masterName + " tells " + this.petName + " to sleep.");
            this.sleep = sleep + 100;
            printStats();
            overflow();
            choose();
        }

        else if(myChoice.equals("test death")|| myChoice.equals("8")) {
            System.out.println(this.masterName + " tells " + this.petName + " to test death.");
            this.alive = false;
            this.restroom = 0;
            this.eat = 0;
            this.sleep = 0;
            this.happiness = 0;
            this.gold = 0;
            reset();
        }

        else if(myChoice.equals("play")|| myChoice.equals("9")) {
            if(this.playDice.roll()) {
                System.out.println(this.masterName + " tells " + this.petName + " to play with " + this.pronoun + ".");
                this.happiness = happiness + 100;
                printStats();
                overflow();
                choose();
            }
            else {
                System.out.println(this.petName + " did not play, and has " + this.playDice.getProb() + " chances to getting it next time.");
```

```
            this.happiness = happiness + 50;
            printStats();
            overflow();
            choose();
        }
    }
    else if(myChoice.equals("go into the suitcase")|| myChoice.equals("10")) {
        if(this.goIntoTheSuitcaseDice.roll()) {
            System.out.println(this.masterName + " tells " + this.petName + " to go into the suitcase.");
            this.happiness = happiness + 100;
            printStats();
            overflow();
            choose();
        }
        else {
            System.out.println(this.petName + " did not go into the suitcase, and has " + this.goIntoTheSuitcaseDice.getProb() + " chances to getting it next time.");
            this.happiness = happiness + 50;
            printStats();
            overflow();
            choose();
        }
    }
    else if(myChoice.equals("buy")|| myChoice.equals("11") || myChoice.equals("buy somethiing")) {
        if(this.buyDice.roll()) {
            System.out.println(this.masterName + " tells " + this.petName + " to go buy something for  " + this.pronoun + ".");
            this.happiness = happiness + 100;
            printStats();
            overflow();
            choose();
        }
        else {
            System.out.println(this.petName + " did not go buy something, and has " + this.buyDice.getProb() + " chances to getting it next time.");
            this.happiness = happiness + 50;
            printStats();
            overflow();
            choose();
        }
    }
```

```
        else if(myChoice.equals("")|| myChoice.equals("12")){
            System.out.println(this.masterName + " tells " + this.petName + " to do nothing.");
            this.happiness = happiness - 5;
            overflow();
            printStats();
            choose();
        }
        else {
            System.out.println(this.masterName + " tells " + this.petName + " to do nothing.");
            this.happiness = happiness - 5;
            overflow();
            printStats();
            choose();
        }

    }

    public void choice(String myChoice) {
        myChoice = choose();
        doChoice(myChoice);
    }

    public void setPetName() {
        this.petName = getPetName();
    }

    public void setMasterName() {
        this.masterName = getMasterName();
    }

    public void setAlive() {
        if(this.alive == false || happiness == 0 || eat == 0 || restroom == 0 || happiness ==0) {
            this.happyTime.stop();
            this.restroomTime.stop();
            this.lifeTime.stop();
            System.out.println(this.petName + " was alive for " + lifeTime.getElapedTime() + " seconds.");
            System.out.println(this.petName + " was happy for " + happyTime.getElapedTime() + " seconds.");
            reset();
```

```
369         }
370         else if(happiness <= 15) {
371             System.out.println("Your Niffler, " + this.petName + " is at " + this.happiness + " level, and might die soon.");
372             this.happyTime.pause();
373         }
374         else if(eat <= 15) {
375             System.out.println("Your Niffler, " + this.petName + " is at " + this.eat + " level, and might die soon.");
376             this.happyTime.pause();
377         }
378         else if(restroom <= 15) {
379             System.out.println("Your Niffler, " + this.petName + " is at " + this.restroom + " level, and might die soon.");
380             this.restroomTime.pause();
381         }
382         else if(sleep <= 15) {
383             System.out.println("Your Niffler, " + this.petName + " is at " + this.sleep + " level, and might die soon.");
384             this.happyTime.pause();
385         }
386         else {
387                 happyTime.unpause();
388                 restroomTime.unpause();
389         }
390         }
391
392     public void live() {
393         while(this.alive) {
394             if(this.alive && this.happiness > 0) {
395                 String choice = choose();
396                 doChoice(choice);
397                 loseStats();
398                 setAlive();
399             }
400             else {
401                 setAlive();
402             }
403         }
404     }
```

```
405
406     public void reset() {
407         if(this.isAlive() == false && this.happiness <= 0) {
408             System.out.println("Your pet has died. Would you like another one?");
409             String yesNo = scan.nextLine().toLowerCase();
410             if(yesNo.contains("yes")) {
411                 this.alive = true;
412                 startHappyTime();
413                 startLifeTime();
414                 startRestroomTime();
415                 getPetName();
416                 getMasterName();
417                 resetStats();
418                 getPronoun();
419                 printStats();
420                 choose();
421             }
422             else {
423                 System.out.println("Thank you for playing!");
424                 this.alive = false;
425                 System.exit(0);
426             }
427         }
428
429     }
430
431     public void setSleep(int sleep) {
432         if(this.sleep < 0) {
433             this.sleep = 0;
434         }
435     }
436
437     public void setEat(int eat) {
438         if(this.eat < 0) {
439             this.eat = 0;
440         }
441     }
442
```

```
443     public void setRestroom(int restroom) {
444         if(this.restroom < 0) {
445             this.restroom = 0;
446         }
447     }
448 
449     public void setHappiness(int happiness) {
450         if(this.happiness < 0) {
451             this.happiness = 0;
452         }
453     }
454 
455     public void setGold(int gold) {
456         if(this.gold < 0) {
457             this.gold = 0;
458         }
459     }
460 
461     public void resetStats() {
462         this.gold = rand.nextInt(21) + 20;
463         this.happiness = 100;
464         this.restroom = 100;
465         this.sleep =100;
466         this.eat = 100;
467     }
468 
469     public void overflow() {
470             if(this.gold > 100) {
471                 this.gold = 100;
472             }
473             else if(this.happiness > 100) {
474                 this.happiness = 100;
475             }
476             else if(this.eat > 100) {
477                 this.eat = 100;
478             }
479             else if(this.restroom > 100) {
480                 this.restroom = 100;
481              }
482              else if(this.sleep > 100) {
483                  this.sleep = 100;
484              }
485      }
486 
487     public void loseStats() {
488         if(this.isAlive() == true) {
489             this.eat -= rand.nextInt(10) + 1;
490             this.sleep -= rand.nextInt(10) + 1;
491             this.restroom -= rand.nextInt(10) + 1;
492             this.happiness -= rand.nextInt(10) + 1;
493         }
494     }
495 }
```