Monster Maker HW

For this homework please construct the class ***Monster.java*** and the derived classes ***CookieMonster.java*** and ***Dragon.java***

MonsterMaker.java is a test file. A sample run is provided at the end of this document.

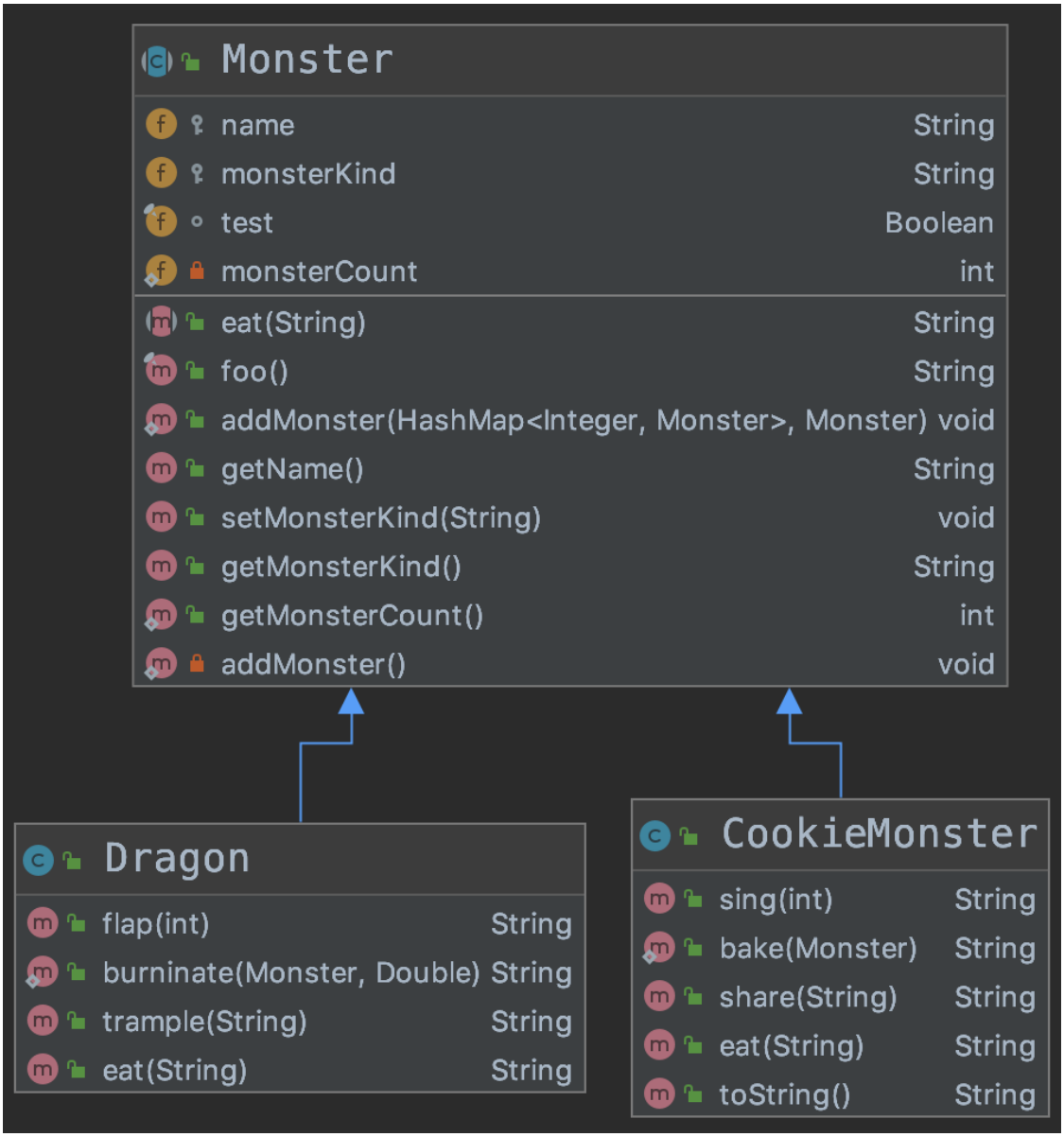

Figure 1: IntelliJ Derived UML diagram

Monster.java

Fields / Data Members / variables

name and monster kind are **protected**
monsterCount is **private** and **static**

constructors

The default constructor uses **this** to call the parameterized constructor with a default value of "Trogdor"

The parameterized constructor sets the **name** field and uses addMonster to increment monster count.

Methods

eat is an **abstract** method (I want to see if you know what this means)

addMonster(HashMap, Monster) adds a Monster object to the provided HashMap. the key is the current size of the HashMap plus 1. The first monster added would be monster 1, etc. (this isn't a great way to do this but I want to see if you know how to get the size of a hashmap)

addMonster() is a private static method that increments **monsterCount**

getName and **getMonsterKind** are standard getters

getMonsterCount is a static method

CookieMonster.java

Constructor

The default constructor calls the parametrized constructor with the default name of “cookieM” and the kind of “Blue Monster”.

The parametrized constructor takes two String variables, name and kind. The inherited constructor is used to set name. monsterKind is set in the CookieMonster instance

Methods

sing takes an int, **mins**. If **mins** is odd and less than 15, the following string is returned (assuming default values):
“cookieM sings C IS FOR COOKIE!! for 7 minutes”
where 7 is the value passed in for **mins**.
Otherwise the String (again assuming default values):
“cookieM says it is not time for signing. Maybe in 41 minutes.”

bake is a **static** method that takes a **Monster** object m.
If m is of type CookieMonster the following is returned:
“cookieM bakes cookies in an oven”
where cookie is the name stored in the Monster object.
otherwise
“Trogdor uses flame breath to bake a cake”
Where Trogdor is the name stored in the monster object

share takes a String, **name**. if **name** is equal to the name stored in the current object:
“cookieM does not share any cookies =-(“
is returned (where cookieM is the value stored in the instance variable **name**)
otherwise the String:
“cookieM shares cookies with Trogdor”
is returned. “cookie” is the name stored in the current object, “Trogdor” is the value for name that was passed in.

eat takes a String, food.
if food is “cookie”
“OM NOM NOM!”
is returned
otherwise
“C is for cookie not avocado”
where “avocado” is the value that was passed in.

Dragon.java

Constructor

The default constructor calls the parametrized constructor with the default name of “trogdor” and the kind of “wingely”.

The parametrized constructor takes two String variables, name and kind. The inherited constructor is used to set name. Kind is set in the Dragon instance

Methods

flap takes an **int**. If flap is even or less than 20 the following String is returned:
“trogdor flaps its tiny wings 7 times”
where 7 is the number that was passed in and “**trogdor**” is the value stored in the instance variable **name**.
otherwise the following is returned
“trogdor can’t even”

burninate is a **static** method. burninate takes a **Monster** object and a **Double**.
if the Monster is of type Dragon the following is returned
“trogdor burninates 12.5 peasants”
Where “trogdor” is the name that is stored in the passed in **Monster** object and 12.5 is the value passed in for the **Double**.
otherwise the following is returned
“cookie eats 12.5 cookies”
where cookie is the name stored in the passed in **Monster** object and 12.5 is the value passed in for the **Double**.

trample takes a String, **thing**.
If **thing** holds the word “cottage” the following is returned
“the cottage is trampled by trogdor”
where “**cottage**” is the thing passed in and “**trogdor**” is the name stored in the instance.
otherwise the following is returned:
“the wall resists trogdor”
were “**wall**” is the value passed in for thing and “**trogdor**” is the name stored in the instance

eat takes a String, **food**
if **food** holds “peasant” the following **String** is returned
“burna-licious”
otherwise the following is returned
“Dragons don’t eat cookies”
where “**cookies**” is the value passed in for the parameter **food**.

Test run

=-=-=-=-= Checking Static Methods =-=-=-=-=

I don’t know what kind of monster that is

Trogdor6 burninates 12.5 peasants

=-=-=-=-= Checking Dragons =-=-=-=-=

Dragons don't eat peasant

Dragons don't eat cookie

Trogdor6 can't even

Trogdor6 flaps its tiny wings 7 times

Trogdor6 flaps its tiny wings 12 times

the cottage is trampled by Trogdor6

the wall resists Trogdor6

=-=-=-=-= Looping =-=-=-=-=

=-=-=-=-= Checking Static Methods =-=-=-=-=

HankMcCoy5 bakes cookies in an oven.

I don't know what this thing eats

=-=-=-=-= Checking CookieMonster Methods =-=-=-=-=

cookie!! OM NOM NOM!

C is for cookie, not avocado

HankMcCoy5 says it is not time for singing. Maybe in 10 minutes

HankMcCoy5 sings C IS FOR COOKIE!! for 7 minutes

HankMcCoy5 says it is not time for singing. Maybe in 41 minutes

HankMcCoy5 does not share any cookies =-(

HankMcCoy5 shares cookies with Trogdor

=-=-=-=-= Looping =-=-=-=-=

=-=-=-=-= Checking Static Methods =-=-=-=-=

I don’t know what kind of monster that is

Trogdor4 burninates 12.5 peasants

=-=-=-=-= Checking Dragons =-=-=-=-=

Dragons don't eat peasant

Dragons don't eat cookie

Trogdor4 can't even

Trogdor4 flaps its tiny wings 7 times

Trogdor4 flaps its tiny wings 12 times

the cottage is trampled by Trogdor4

the wall resists Trogdor4

=-=-=-=-= Looping =-=-=-=-=

=-=-=-=-= Checking Static Methods =-=-=-=-=

HankMcCoy3 bakes cookies in an oven.

I don't know what this thing eats

=-=-=-=-= Checking CookieMonster Methods =-=-=-=-=

cookie!! OM NOM NOM!

C is for cookie, not avocado

HankMcCoy3 says it is not time for singing. Maybe in 10 minutes

HankMcCoy3 sings C IS FOR COOKIE!! for 7 minutes

HankMcCoy3 says it is not time for singing. Maybe in 41 minutes

HankMcCoy3 does not share any cookies =-(

HankMcCoy3 shares cookies with Trogdor

=-=-=-=-= Looping =-=-=-=-=

=-=-=-=-= Checking Static Methods =-=-=-=-=

I don’t know what kind of monster that is

Trogdor2 burninates 12.5 peasants

=-=-=-=-= Checking Dragons =-=-=-=-=

Dragons don't eat peasant

Dragons don't eat cookie

Trogdor2 can't even

Trogdor2 flaps its tiny wings 7 times

Trogdor2 flaps its tiny wings 12 times

the cottage is trampled by Trogdor2

the wall resists Trogdor2

=-=-=-=-= Looping =-=-=-=-=

=-=-=-=-= Checking Static Methods =-=-=-=-=

HankMcCoy1 bakes cookies in an oven.

I don't know what this thing eats

=-=-=-=-= Checking CookieMonster Methods =-=-=-=-=

cookie!! OM NOM NOM!

C is for cookie, not avocado

HankMcCoy1 says it is not time for singing. Maybe in 10 minutes

HankMcCoy1 sings C IS FOR COOKIE!! for 7 minutes

HankMcCoy1 says it is not time for singing. Maybe in 41 minutes

HankMcCoy1 does not share any cookies =-(

HankMcCoy1 shares cookies with Trogdor

=-=-=-=-= Looping =-=-=-=-=

the final count of Monsters: 6