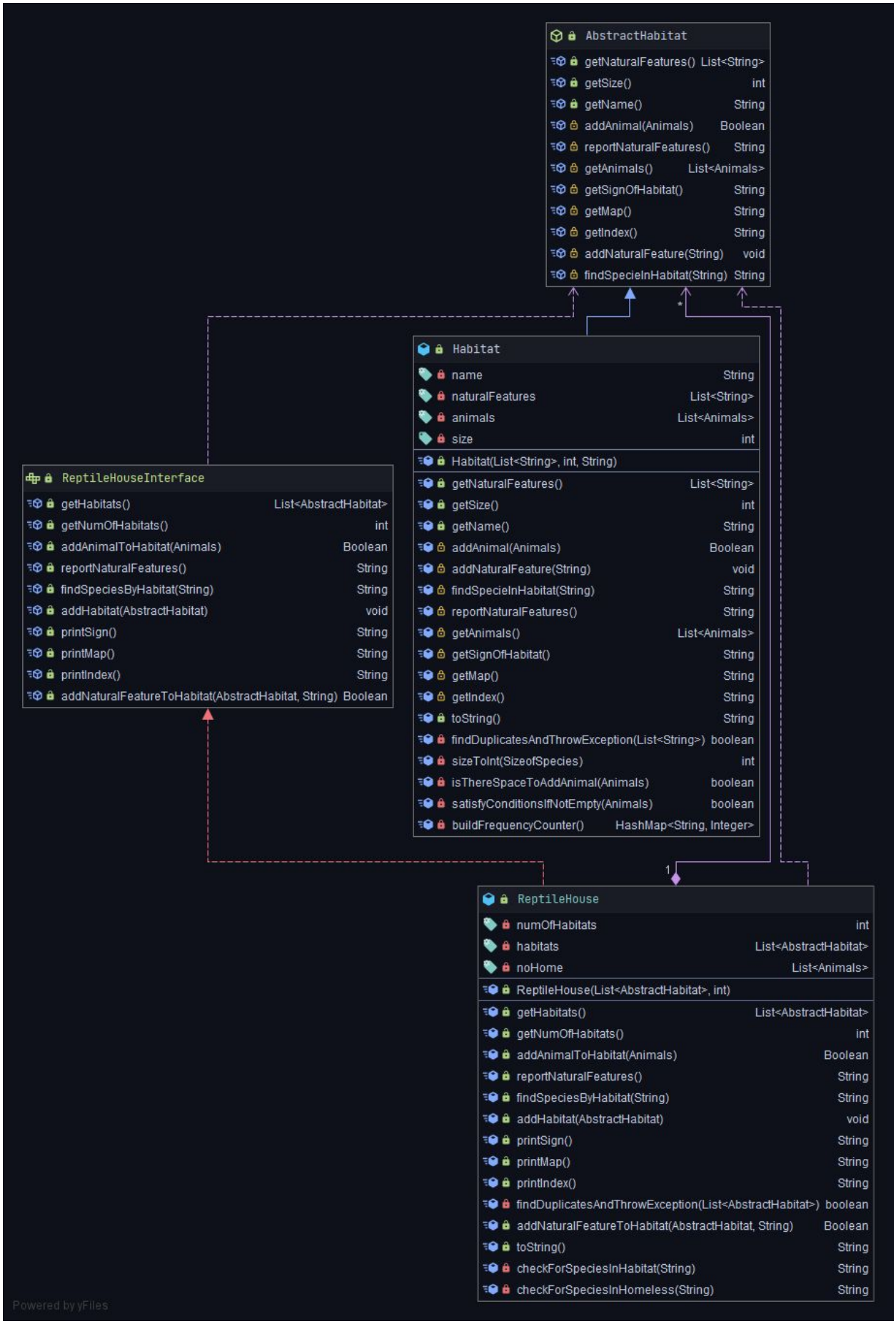
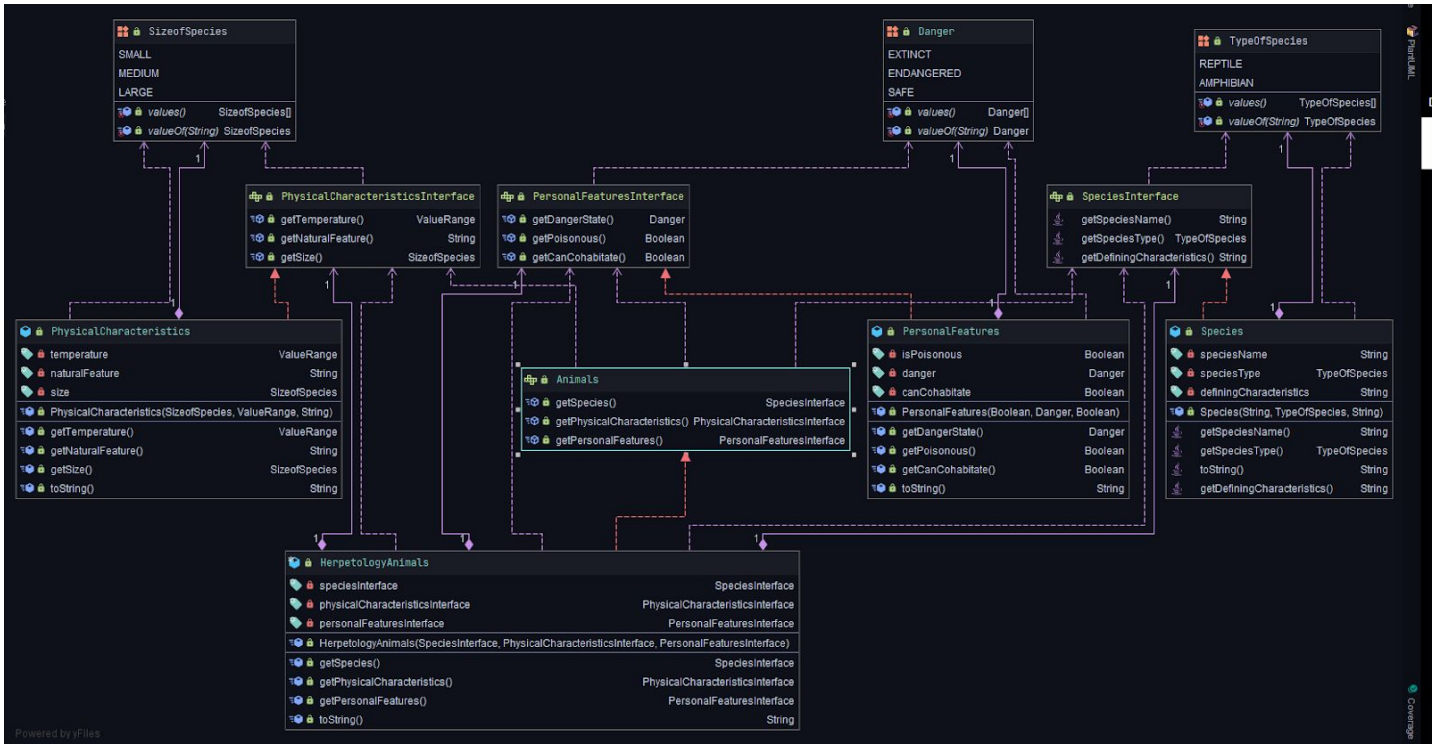
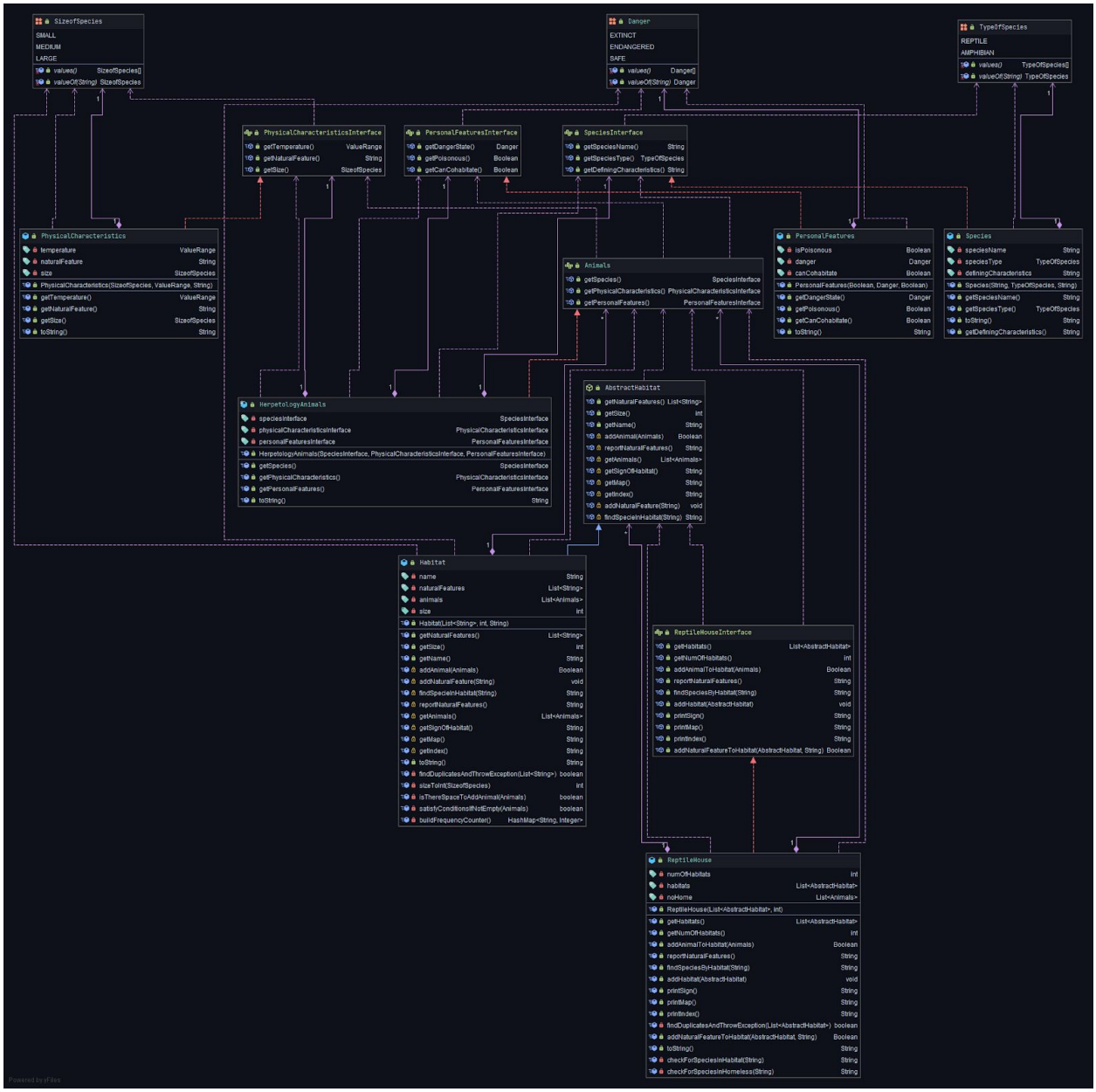
NEW UML DIAGRAM


AbstractHabitat
getNaturalFeatures() List<String>
getSize() int
getName() String
addAnimal(Animals) Boolean
reportNaturalFeatures() String
getAnimals() List<Animals>
getSignOfHabitat() String
getMap() String
getIndex() String
addNaturalFeature(String) void
findSpecieInHabitat(String) String
Habitat
name String
naturalFeatures List<String>
animals List<Animals>
size int
Habitat(List<String>, int, String)
getNaturalFeatures() List<String>
getSize() int
getName() String
addAnimal(Animals) Boolean
addNaturalFeature(String) void
findSpecieInHabitat(String) String
reportNaturalFeatures() String
getAnimals() List<Animals>
getSignOfHabitat() String
getMap() String
getIndex() String
toString() String
findDuplicatesAndThrowException(List<String>) boolean
sizeToInt(SizeofSpecies) int
isThereSpaceToAddAnimal(Animals) boolean
satisfyConditionsIfNotEmpty(Animals) boolean
buildFrequencyCounter() HashMap<String, Integer>
ReptileHouseInterface
getHabitats() List<AbstractHabitat>
getNumOfHabitats() int
addAnimalToHabitat(Animals) Boolean
reportNaturalFeatures() String
findSpeciesByHabitat(String) String
addHabitat(AbstractHabitat) void
printSign() String
printMap() String
printIndex() String
addNaturalFeatureToHabitat(AbstractHabitat, String) Boolean
ReptileHouse
numOfHabitats int
habitats List<AbstractHabitat>
noHome List<Animals>
ReptileHouse(List<AbstractHabitat>, int)
getHabitats() List<AbstractHabitat>
getNumOfHabitats() int
addAnimalToHabitat(Animals) Boolean
reportNaturalFeatures() String
findSpeciesByHabitat(String) String
addHabitat(AbstractHabitat) void
printSign() String
printMap() String
printIndex() String
findDuplicatesAndThrowException(List<AbstractHabitat>) boolean
addNaturalFeatureToHabitat(AbstractHabitat, String) Boolean
toString() String
checkForSpeciesInHabitat(String) String
checkForSpeciesInHomeless(String) String
*
1
Powered by yFiles

House Package Class Diagram



Animal Package Diagram



Combined Diagram

Testing

Testing Species Class		**Expected Value**
Species Name	getSpeciesName()	“frog”
Species Type	getSpeciesType()	REPTILE
Constructor disallows type not defined in enum	Species(“frog”,Mammal)	IllegalArgument Exception

Species species = new Species(“frog”, TypeOfSpecies.REPTILE)

Testing Physical Char Class		**Expected Value**
Temperature Range	getTemperature()	0 - 123
Natural Feature	getNaturalFeature()	“water”
Size	getSize()	“small”

PhysicalCharacteristics physical = new
PhysicalCharacteristics(ValueRange.of(0,123),”water”,”small”);

Testing Personal Feat Class		**Expected Value**
Get Danger Level	getDangerState()	EXTINCT
Check Poisonous	getPoisonous()	false
Constructor disallows true on cohabitable if extinct and poisonous is true	PersonalFeatures(true,true,true)	IllegalArgumentException

PersonalFeatures pf = new PersonalFeatures(true,false,false)

Testing HerpetologyAnimals Class		**Expected Value**
Test toString Method	toString()	“frog,reptile,small,water,true,false,false”

HerpetologyAnimals hAnimals = new HerpetologyAnimals(Species species, PhysicalCharacteristics physical, PersonalFeatures pf)

Testing Habitat		**Expected Value**
Natural Features	getNaturalFeatures()	[“water”,”grass”,”beach”]
Check Size	getSize()	20
Habitat Name	getName()	“Loc A”
Add animal to Habitat	addAnimal(hAnimals,hAnimals.getPhysicalCharacteristics.getSize())	“frog”
Constructor disallows negative values	Habitat([“water”,”grass”,”beach”],-20,”Loc A))	IllegalArgumentException
Constructor disallows array length less than 1	Habitat([],-20,”Loc A))	IllegalArgumentException

AbstractHabitat habitat = new Habitat([“water”,”grass”,”beach”],20,”Loc A”)

Reptile House		**Expected Value**
Add animal to habitat	addAnimalToHabitat(hAnimals)	“frog”
Generate Natural Features	reportNaturalFeatures()	“water,grass,beach,LocA”
Find By Habitat	findSpeciesByHabitat(“frog”)	“Loc A”
Test species not in habitat	findSpeciesByHabitat(“snake”)	“Not currently in a habitat”
Test species not found	findSpeciesByHabitat(“lion”)	“Not in reptile house”

Add Habitat	addHabitat(AbstractHabitat habitat2)	2
Constructor disallows negative value	ReptileHouse([habitat],-1)	IllegalArgumentException
Constructor disallows length of array > num Of Habitats	ReptileHouse([habitat,habitat-2],1)	IllegalArgumentException
Constructor disallows length of array < 1	ReptileHouse([],2)	IllegalArgumentException

ReptileHouseInterface rHouse = new ReptileHouse([habitat],2)