Technical Merit: 100
Documentation: 100
Quality of code: 100
Program runs: 95

49.5

Harrison Zheng

CMPT 220

Professor Arias

14 May 2020

Project 2 Final Writeup

Abstract

For my project, I designed and programmed an application that stores data about turtles. The application was written in the Java programming language in the NetBeans IDE. It includes a Java database that stores information on individual turtles and a user interface made with Swing to read and write to the database. In this report, I describe in detail the program that I created and cover other important information pertaining to it.

Introduction

My inspiration for this project came from my fascination with turtles and my desire to help in both research and conservation efforts for endangered turtle species. I also saw this project as an opportunity to learn how databases work and get some experience using them so that I can be more prepared to take courses that involve database programming in the near future. In the next section, I explain what my program does and how specific users interact with it. Then, I elaborate on the problem that my program addresses and other work that has been done to address the same or similar problems. Afterwards, I explain how the program should be used. Finally, I conclude by summarizing the goals accomplished by my program and list the sources that I consulted while working on this project.

Detailed System Description

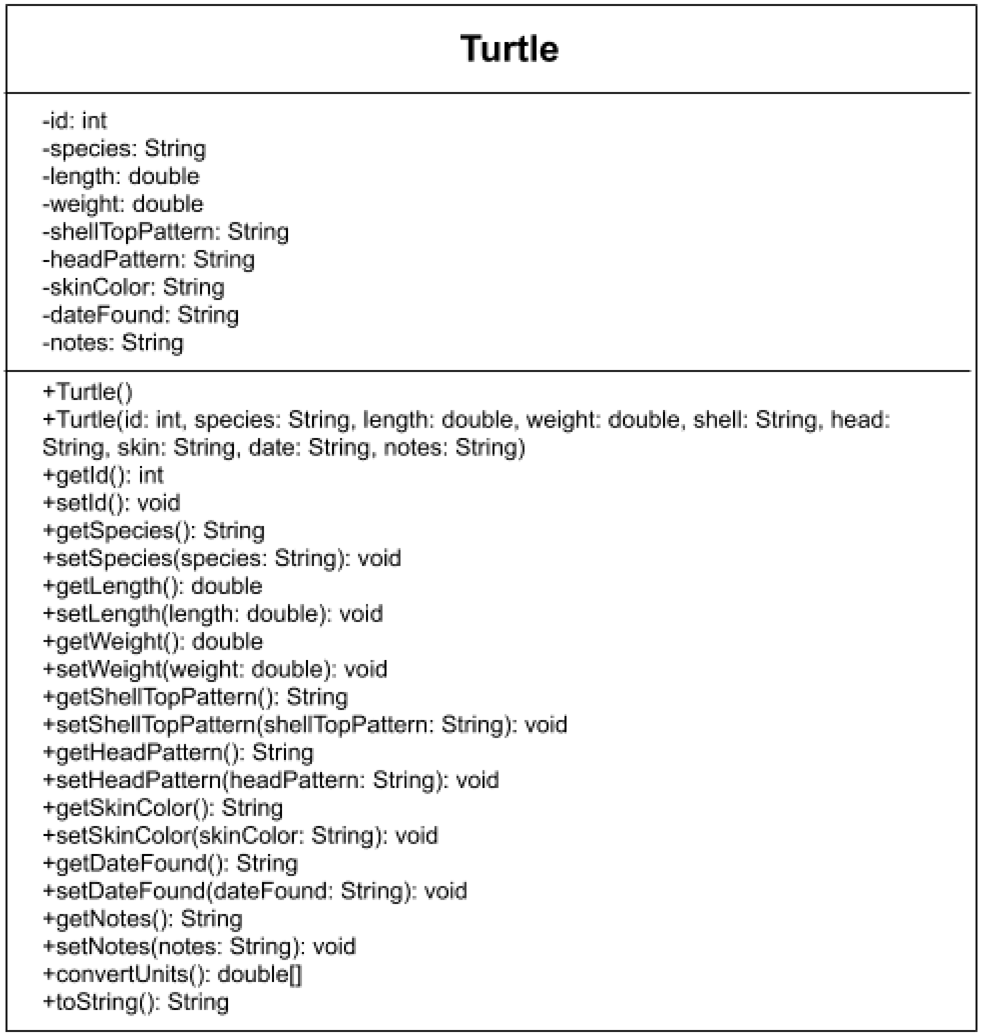
My program was designed to help scientists and wildlife conservationists keep track of individual turtles that they come across while completing their field work. Once they spot a turtle, they can use the program's graphical user interface (GUI) to enter information that is commonly used to identify turtles. Specifically, this includes an identifying number, the species the turtle belongs to, its length, its weight, the pattern on the top of its shell, the pattern on its head and neck, its skin color, the date found it was found, and any additional information worth noting. All the information entered by the user is then captured by a method in the *GUI* class and handed off to a method in the *Add* class. The *Add* class's method writes a SQL statement with the information that it is given and then executes it to create an entry in the program's database. If the statement is successfully executed, the program displays the new entry along with all the others in the database as rows in the GUI's table.

Entries remain in the database until the user decides to modify or delete them. The user can modify an entry by entering the updated information into the program. This information is captured by another method in the *GUI* class, which then hands it off to the *Save* class. Similar to the *Add* class, the *Save* class writes a SQL statement and then executes it to modify the database entry with the ID number specified by the user. If the user decides they want to delete an entry, they simply need to enter the ID associated with it. A separate method in the *GUI* class then gives the ID to the *Delete* class, which writes and executes a SQL statement that deletes the entry with the specified ID. Changes to information in database entries and deletions are reflected in the GUI's table after each action is performed.

Since the scientific community almost always uses the metric system to take measurements, I designed my program to take the length and weight of turtles in centimeters and kilograms. I realized, though, that an American who does not often use metric units may become annoyed while reading from the database. To solve this problem, I implemented a method in the *GUI* class that takes an ID number from the user and hands it off to the *Convert* class, which then find the entry with the specified ID and converts its length and weight into inches and pounds and returns the values to be displayed in the GUI.

Another class in my program that some users may find useful is the *Turtle* class. This class contains data fields that can hold the same information (species, length, date found, etc.) as an entry in the database. It also has constructors that can be used to create instances of the *Turtle* class with specific traits, getter and setter methods for returning and changing values in an instance, a method that converts an instance's length and weight to inches and pounds, and a method that returns a string representation of an instance. Unlike the other classes in my program, the *Turtle* class does not read from or write to the database as it was made for users who prefer to manipulate data through coding themselves. It is important to note that the *Add*, *Save*, *Delete*, and *Turtle* classes in my program are sub-classes of the *GUI* class because sometimes they need to use SQL objects and other data declared in the *GUI* class. The *GUI* class itself is actually a sub-class of Java Swing's JFrame class. More information on all of the classes in my program can be found in the UML diagrams on the next two pages.

Turtle Class UML Diagram

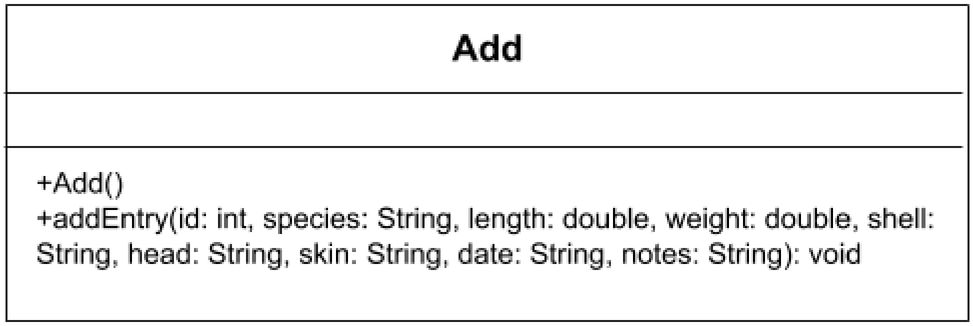


Turtle

-id: int
-species: String
-length: double
-weight: double
-shellTopPattern: String
-headPattern: String
-skinColor: String
-dateFound: String
-notes: String

+Turtle()
+Turtle(id: int, species: String, length: double, weight: double, shell: String, head: String, skin: String, date: String, notes: String)
+getId(): int
+setId(): void
+getSpecies(): String
+setSpecies(species: String): void
+getLength(): double
+setLength(length: double): void
+getWeight(): double
+setWeight(weight: double): void
+getShellTopPattern(): String
+setShellTopPattern(shellTopPattern: String): void
+getHeadPattern(): String
+setHeadPattern(headPattern: String): void
+getSkinColor(): String
+setSkinColor(skinColor: String): void
+getDateFound(): String
+setDateFound(dateFound: String): void
+getNotes(): String
+setNotes(notes: String): void
+convertUnits(): double[]
+toString(): String

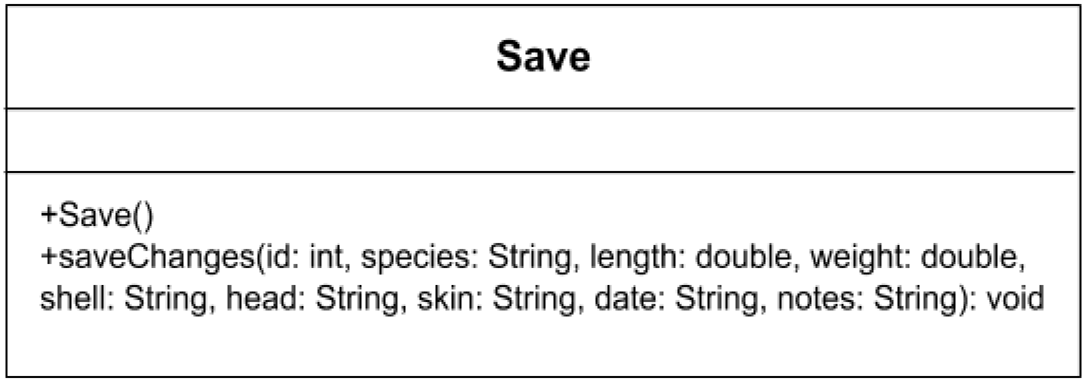
Add Class UML Diagram



Add

+Add()
+addEntry(id: int, species: String, length: double, weight: double, shell: String, head: String, skin: String, date: String, notes: String): void

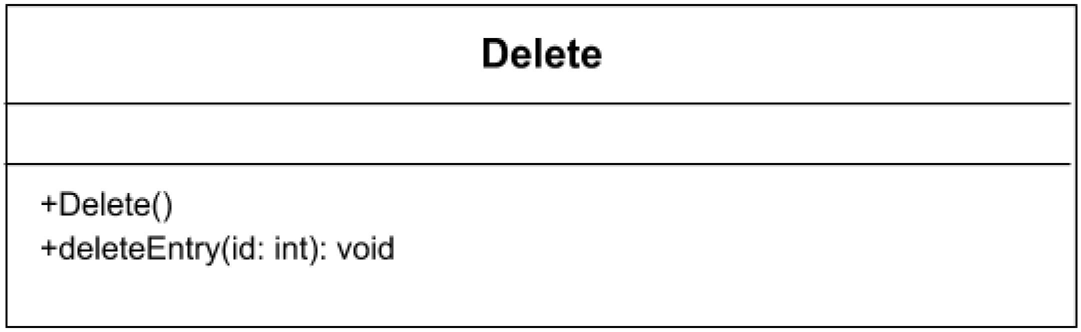
Save Class UML Diagram



Save

+Save()
+saveChanges(id: int, species: String, length: double, weight: double, shell: String, head: String, skin: String, date: String, notes: String): void

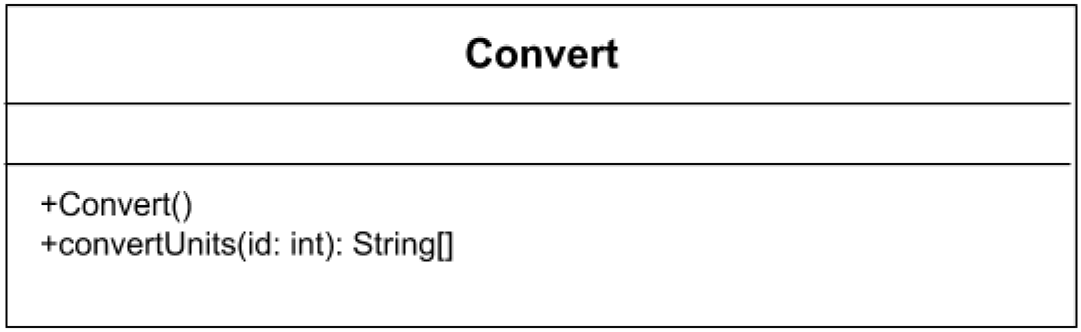
Delete Class UML Diagram



Delete

+Delete()
+deleteEntry(id: int): void

Convert Class UML Diagram



Convert

+Convert()
+convertUnits(id: int): String[]

Requirements

Turtles have been roaming Earth since the time of the dinosaurs. But over the last few decades, turtle populations around the world have declined significantly as a result of poaching, pollution, and climate change. Today, over 120 turtle species are listed as either vulnerable or endangered. The purpose of my program is to help scientists save these creatures from extinction by monitoring individual turtles. Doing this allows them to pinpoint which human activities cause the most amount of harm to a population and determine what should be done to mitigate the effects of these activities. Keeping track of individuals allows scientists to find other commonalities within turtle populations as well, such as their diets and where the females like to nest. This information may help government and wildlife organizations improve their approach to protecting turtles.

Literature Survey

My program was partially inspired by the turtle identification guide on Discoverlife.org. The guide helps people identify the species that a turtle belongs to by having them choose options from a list of identifying features, a couple of which I directly included in my program. In addition to showing me how to identify turtles, the guide also gave me some ideas for how the layout of my program's GUI should look like. A website I found that seeks to address the same problem as my program, though, is Seaturtlestatus.org. The site contains an interactive database of sea turtles spotted around the world by members of an organization called The State of the World's Sea Turtles (SWOT) and sea turtle researchers. The large database is accessible to all users because according to the website, the SWOT team "believe[s] that sharing data about sea turtles in a publicly-available global database is critical to effective sea turtle conservation".

While my program and SWOT’s database were both designed with the goal of helping turtle conservation efforts, their database achieves this on a much larger scale than my program, which is better suited for monitoring local turtle populations.

User Manual

Upon running my program, the user will see a window pop up with the title “Turtle Records” at the top. Below that will be a table with three rows already filled in to serve as examples of what kinds of information the user may enter, labeled data fields where the user will enter all of their data (ID, species, length, etc.), and four buttons that perform various functions. To add an entry to the database, the user enters information about a turtle into the appropriate data fields and then clicks the “Add” button. To revise an entry in the database, the user enters the new information into the appropriate data fields and then clicks the “Save” button. To delete an entry from the database, the user enters the ID of the entry they want to delete and then clicks the “Delete” button. To convert the length and weight of an entry into inches and pounds, the user enters the entry’s ID and then clicks the “Convert Units” button. When the user is done using the program, they can simply close the window to terminate it.

Conclusion

The purpose of my program is to help scientists track turtle populations and determine whether their numbers are healthy or action needs to be taken to protect them from going extinct. Although my program is not perfect and there are many applications and websites already in use that serve the same purpose as my program, I believe that going through the long process of building it has expanded my understanding of databases and how to use them like I hoped it would. Completing this project has also greatly improved my problem-solving skills as I had to fix numerous errors in my code that I had never encountered before. This newfound knowledge is something that I am sure I will find valuable as I continue to learn about the field of computer science and create programs.

References/Bibliography

"Java Tutorial - Creating a Java Gui Application with DataBase - Coding Part." *YouTube*, uploaded by Aghiad Bakry, 2 Apr. 2016, www.youtube.com/watch?v=AzeJEEeGl_w&list=PLhWlag_D0srReZOq8xtmQQWpSBtwAPhfk&index=12&t=1187s. Accessed 14 May 2020.

"JDBC - Select Records Example." *Tutorials Point*, www.tutorialspoint.com/jdbc/jdbc-select-records.htm. Accessed 14 May 2020.

"SQL Server Basics." *SQLServerTutorial.net*, www.sqlservertutorial.net/sql-server-basics/. Accessed 14 May 2020.

SWOT Team. "Online Map & Sea Turtle Database." *The State of the World's Sea Turtles*, www.seaturtlestatus.org/online-map-data. Accessed 14 May 2020.

"Turtles." *Discover Life*, edited by John Pickering, www.discoverlife.org/mp/20q?guide=Turtles. Accessed 14 May 2020.

"Working with the Java DB (Derby) Database." *NetBeans*, netbeans.org/kb/docs/ide/java-db.html. Accessed 14 May 2020.