Bluegrass Community and Technical College

Programming Requirements Document

Stately Information – Array of Objects

NARRATIVE DESCRIPTION

The Student Example discussed in this week's lecture (podcast) will be extremely useful as a model for this assignment.

Included with this assignment in Blackboard is a class named State.java. It's original UML diagram can be found below. The State class follows this design.

Some Driver
main (args: String[ ]): void

State
- stateName : String - stateCapital : String - stateBird : String - stateFlower : String
+ State ( ): + State (String stateName, String stateCapital, String stateBird, String stateFlower):
+ getStateName( ) : String + getStateCapital( ) : String + getStateBird( ) : String + getStateFlower( ) : String + setStateName(String color) : void + setStateCapital(String color) : void + setStateBird(String color) : void + setStateFlower(String color) : void + toString( ) : String

The class currently does not have any security or data validation for parameters.

Step 1: Update State Class and Add Security (R1 – R2 Requirements)

You are to update the class to include 2 new pieces of data:

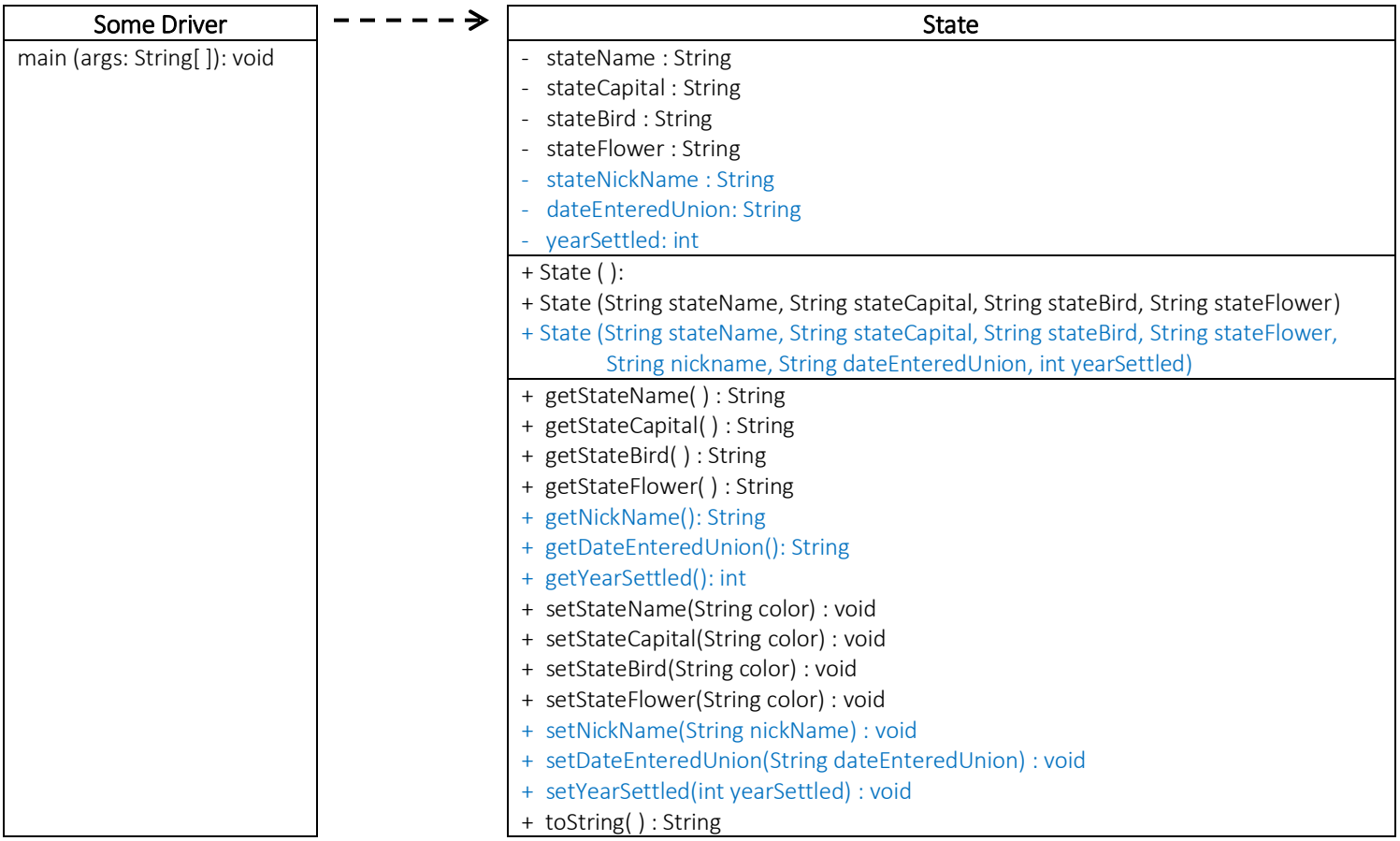
(1) the date the state entered the union (state was ratified)

- This is a String in the format YYYY-MM-DD. For example, 1819-12-14 means the state entered the union on December 14, 1819.

(2) the year the state was first settled

- This an integer. For example, 1702 means the state had its first settlement in 1702.

The newly designed UML diagram for the State class is:



Add security to the class by validating parameters where feasible. Your instructor is looking to see if you can apply security on your own. For example, the year a state was settled is an integer which can be validated between 1600 and the current year (allowing for future states).

There is a Calendar class in the java.util package. To extract the current year from your computer's date, use the method below. This method returns an integer value so you can assign that value to an integer variable.

```
Calendar.getInstance().get(Calendar.YEAR)
```

To use the method above, include the following import statement in your class:

```
import java.util.Calendar;
```

Step 2: Create an Array of Objects (R3 Requirement)

While this assignment could be completed without arrays, the purpose of this assignment is to practice using an array of objects. A data file, named StateData.txt, is provided to give you data to use in the program for the array. You are required to use an array of State objects and not a 2D-array of string/integers. The Student class example explained in the lecture (podcast) this week will be a great model to follow.

Sample data from the array is:

```
Alabama,Montgomery,Yellowhammer,Camellia,Yellowhammer State,1819-12-14,1702
Alaska,Juneau,Willow Ptarmigan,Forget-me-not,The Last Frontier,1959-01-03,1784
```

Arizona,Phoenix,Cactus Wren,Saguaro Cactus Blossom,Grand Canyon State,1912-02-14,1776

To create the array of State objects, do the following for each state (record) in the file:

- Read a record from the StateData.txt file. Each record has 7 fields:
  - *state name*
  - *state capital*
  - *state bird*
  - *state flower*
  - *state nickname*
  - *the date the state entered the union*
  - *the year the state was originally settled*
- Instantiate a State object using the fields above
- Store the State object in the array

Step 3: Use the Array of Objects (R4 – R5 Requirements)

Display all the state names and ratified date in a nicely formatted manner using the array of State objects (and not directly from the file and not while the array is being created). This shows me you can use an array of objects after it is created.

```
STATE NAME                Ratified Date

Alabama                   12-14-1819
Alaska                    01-03-1959
Arizona                   02-14-1912
…etc.
```

Below the list of states, create the following statistical data where XX and YY are counts for each category:

```
Number of states settled before 1700:    XX
Number of states with any type of Rhododendron as their state flower:  YY
```

Comments

In a real-world setting this could be done using file processing without any arrays at all. Remember – This exercise was designed for me to assess your manipulation of arrays. Please complete the steps above as indicated. Use separate loops for step 2 and step 3. In a real-world, we would not want this duplication of code, however, for this assignment I am interested in seeing you (1) create and load an array and (2) use an array to calculate results.

Zip your .java file, the State class, and the States.txt file together and submit in Blackboard for grading.

Restrictions:

You must use an array of objects to hold the state data and not a single and/or two-dimensional array of primitive data type.

SOFTWARE REQUIREMENTS

R1: Update the State class to include the two new data times, their accessors, and their mutators.
R2: Add security into the State class so that parameters are validated. In the constructor, assign default values (null strings or zeroes) if parameter data is invalid. In the mutators, do not change instance data items if the parameters are invalid, leave as is.
R3: Create an array of state objects using the data from the StateData.txt file.
R4: Display the state names and date each state entered the union (was ratified) displayed in aligned columns. The date is displayed as mm-dd-yyyy (note: the data in the file is in the form yyyy-mm-dd).
R5: Calculate and display the number of states who were originally settled before 1700.
R6: Calculate and display the number of states who have some type of rhododendron as a state flower.

Security Considerations

Classes should validate method parameters, use private instance data items, and only provide accessors and mutators where needed and necessary. Array out-of-bound errors can cause security vulnerabilities.

Special Notes

None provided for this assignment.

Change Request Form

Students who wish to obtain written permission to alter the assignment or to use features/statements/structures before they are introduced in class, must complete a *Change Request Form* (link in Blackboard in left-hand navigation bar) and follow all guidelines provided there.