

Immutability

Lecture Question

Restriction: No state is allowed in this question. Specifically, the keyword "var" is banned

Question: In a package named "functions" write a class named VoteCounter with the following features:

- Has a constructor that takes a List of Strings representing all votes that have been cast
- A method named "castVote" that takes a String as a parameter representing a vote for a candidate and returns a new VoteCounter with this vote added to the List of votes
- A method named "submitVotes" that takes a String representing a candidate to vote for and an Int representing the number of votes to cast for that candidate and returns a new VoteCounter with these votes added
- A method named "countVotes" that takes a String representing a candidate and returns the number of votes cast for that candidate

Testing: In a package named "tests" create a class named "TestVoteCounter" as a test suite that tests all the functionality listed above

Example

Restriction: No state is allowed in this question. Specifically, the keyword "var" is banned

Question: In a package named "functions" write an object named NumberGuesser with a method named guessTheNumber with the following features:

- Plays a number guessing game where there is a hidden number that needs to be found
- Takes 2 Doubles and function Double to Boolean as parameters representing:
 - The lower bound of the number to guess
 - The upper bound of the number to guess
 - A function that tells you if you need to search higher or lower
 - This function returns true if the hidden number is higher than the input number, false otherwise
- Returns The hidden number to 2 decimal places

Testing: In a package named "tests" create a class named "TestNumberGueser" as a test suite that tests all the functionality listed above

Example

Restriction: No state is allowed in this question. Specifically, the keyword "var" is banned

Question: In a package named "functions" write an object named Numbers with a method named exponentiate with the following features:

- Takes a Double and an Int and returns the Double raised to the power of the Int
- Your method should run in $O(\log(n))$ time where n is the value of the exponent

Testing: In a package named "tests" create a class named "TestExponent" as a test suite that tests all the functionality listed above

Example

Restriction: No state is allowed in this question. Specifically, the keyword "var" is banned

Question: Add a method to the VoteCounter class named winer that takes no parameters and returns the candidate with the most votes

Lecture Question

Restriction: No state is allowed in this question. Specifically, the keyword "var" is banned

Question: In a package named "functions" write a class named VoteCounter with the following features:

- Has a constructor that takes a List of Strings representing all votes that have been cast
- A method named "castVote" that takes a String as a parameter representing a vote for a candidate and returns a new VoteCounter with this vote added to the List of votes
- A method named "submitVotes" that takes a String representing a candidate to vote for and an Int representing the number of votes to cast for that candidate and returns a new VoteCounter with these votes added
- A method named "countVotes" that takes a String representing a candidate and returns the number of votes cast for that candidate

Testing: In a package named "tests" create a class named "TestVoteCounter" as a test suite that tests all the functionality listed above

Lecture Question

Example:

```
val vc1: VoteCounter = new VoteCounter(List())  
val vc2: VoteCounter = vc1.castVote("A")  
val vc3: VoteCounter = vc2.castVote("A")  
val vc4: VoteCounter = vc3.submitVotes("A", 3)  
val vc5: VoteCounter = vc4.submitVotes("B", 10)
```

```
assert(vc5.countVotes("A") == 5)  
assert(vc5.countVotes("B") == 10)
```

Tip: Since the voting methods return new VoteCounters you can chain method calls together that can make your tests easier to write.

```
val vc0: VoteCounter = new VoteCounter(List()).castVote("A").castVote("A").submitVotes("A", 3).submitVotes("B", 10)
```