Work on project. Stage 1/6: Validate the answer

Project: Guess the Animal

Validate the answer

Hard 58 minutes **171** users solved this stage. Latest completion was **about 14 hours ago**.

Description

You have probably played some version of this game before: one player thinks of a person, an animal, or something else, and another one tries to guess what it is based on the answers. In this project, we will create a similar text game where a computer tries to guess an animal that the user has thought of. One of the first versions of this game was written by Arthur Luehrmann at Dartmouth College in the 1960s.

In the first stage, we will implement greetings and goodbyes, as well as the formation of a question based on the name of the animal.

82 / 85 Prerequisites

- Computer programming 20
- Paradigms 20
- Introduction to Java 20
- Basic literals 20
- Overview of the basic program 20

Show all

Join a study group for the project Guess the Animal

Discuss your current project with fellow learners and help each other.

Objectives

The computer should greet the user based on the time of the day. “Good morning” should be used from 5:00 am to 12:00 pm, “Good afternoon” is best for the period between 12:00 pm and 6:00 pm, and “Good evening” is used after 6 p.m. You can also get creative and use “Hi, Night Owl” for late night or “Hi, Early Bird” for early morning.

After greeting the user, the program should ask for an animal. The animal should be entered in the following format: a/an + name of the animal, for example, “an elephant”. The user may or may not specify the article. If the user specifies the undefined article it should be kept as is. Otherwise, the program should determine it according to the rules. For simplicity, let's assume that all words beginning with a vowel take the article "an", and those that begin with a consonant take the article "a". The defined article should be replaced by undefined.

Here are examples:

```
cat => a cat
a cat => a cat
the cat => a cat
ape => an ape
an ape => an ape
the ape => an ape
an elephant => an elephant
a unicorn => a unicorn
a xantic sargo => a xantic sargo
an xeme => an xeme
```

After entering the name of the animal, the program should formulate the question `"Is it a/an <animal>?"`. The question must use the article specified by the user or determined by the program.

Then, the user should answer this question. The computer must perceive these responses as positive: *y, yes, yeah, yep, sure, right, affirmative, correct, indeed, you bet, exactly, you said it*. The negative answer could be: *n, no, no way, nah, nope, negative, I don't think so, yeah no*. The letters can be in any case and there can be a period or an exclamation mark at the end of the statement. In case the user's answer is not clear, the program should ask the user to clarify.

With the clarification question, you don't have to stick to one formal phrase. We are creating a game, so let your fantasy fly! Make the computer ask again in a different way each time. The only condition is that the phrase should contain "yes or no". Check out some example phrases below, but you are free to invent your own phrases:

"I'm not sure I caught you: was it yes or no?"
"Funny, I still don't understand, is it yes or no?"
"Oh, it's too complicated for me: just tell me yes or no."
"Could you please simply say yes or no?
"Oh, no, don't try to confuse me: say yes or no."

After accepting the answer, the program should say goodbye to the user. To make our program more lively, let there be several ways to do it. If you need some examples of cool goodbye phrases, watch a Youtube video 33 ways to say "Bye"

Examples

The greater-than symbol followed by a space `> ` represents the user input. Note that it's not part of the input.

Example 1

```
Good morning!

Enter an animal:
> cat
Is it a cat?
> Yeah
You answered: Yes

Have a nice day!
```

Example 2

```
Good afternoon!

Enter an animal:
> unicorn
Is it an unicorn?
> Oops..
Come on, yes or no?
> Nope
You answered: No

See you soon!
```

Example 3

```
Good evening!

Enter an animal:
> a unicorn
Is it a unicorn?
> Sure!
You answered: Yes

Bye!
```

Report a typo

HINT by **HeyMilkshake (β)**

Don't go straight creating rules that strip punctuation from user input. If you take a look at the test files, there are two rules that need to be respected:
The user can put a point
An exclamation mark is also allowed.
To be considered a positive or negative reply, the only allowed punctuation is a simple Period (.) or a simple Exclamation Mark (!)
any other combination of punctuation should be treated as an Unclear reply.

Was this hint helpful? Yes No Report

Write a program

Code Editor IDE

CONNECTION STATUS

× IDE is not installed or is not running (not opened)

Please open a JetBrains IDE or install and configure it from this page ↗

Solve in IDE

src/animals/Main.java

```
package animals;

import java.time.LocalTime;
import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class Main {
    static LocalTime MORNING_START = LocalTime.of(5, 1);
    static LocalTime MORNING_END = LocalTime.of(12, 0);
    static LocalTime DAY_START = LocalTime.of(12, 1);
    static LocalTime DAY_END = LocalTime.of(18, 0);
    static final Pattern STARTS_WITH_VOWEL = Pattern.compile("^[aeiouy]", Pattern.CASE_INSENSITIVE);

    public static void main(String[] args) {
        System.out.println(checkTimeOfDay() + "\n");
        System.out.println("Enter an animal:");
        Scanner scanner = new Scanner(System.in);
        String userAnimal = scanner.nextLine();
        System.out.println("Is it " + checkAnimal(userAnimal) + "?");
        String userAnswer = scanner.nextLine();
        System.out.println(checkYesNoAnswer(checkIfAnswerHasDot(userAnswer)) + "\n");
        System.out.println(sayBye());
    }

    public static String checkTimeOfDay() {
        LocalTime now = LocalTime.now();
        if (now.isAfter(MORNING_START) && now.isBefore(MORNING_END)) {
            return "Good morning!";
        } else if (now.isAfter(DAY_START) && now.isBefore(DAY_END)) {
            return "Good day!";
        } else {
            return "Good evening!";
        }
    }

    public static String checkAnimal(String userAnimal) {
        String[] words = userAnimal.split(" ");
        Pattern pattern = Pattern.compile("^the\\s|^an\\s|^a\\s|^THE\\s|^AN\\s|^A\\s");
        Matcher matcher = pattern.matcher(userAnimal);

        if (words[0].equals("the") && !STARTS_WITH_VOWEL.matcher(words[1]).find()) {
            return userAnimal.toLowerCase().replace("the", "a");
        } else if (words[0].equals("the") && STARTS_WITH_VOWEL.matcher(words[1]).find()) {
            return userAnimal.toLowerCase().replace("the", "an");
        } else if (!matcher.find() && STARTS_WITH_VOWEL.matcher(words[0]).find()) {
            return "an " + userAnimal.toLowerCase();
        } else if (!matcher.find() && !STARTS_WITH_VOWEL.matcher(words[0]).find()) {
            return "a " + userAnimal.toLowerCase();
        }
        return userAnimal.toLowerCase();
    }

    public static String checkYesNoAnswer(String userAnswer) {
        List<String> yesAnswers = new ArrayList<>(Arrays.asList("y", "yes", "yeah",
                "yep", "sure", "right", "affirmative", "correct", "indeed", "you bet", "exactly", "you said it"));
        List<String> noAnswers = new ArrayList<>(Arrays.asList("n", "no", "no way", "nah", "nope",
                "negative", "I don't think so, yeah no"));
        List<String> clarificationQuestion = new ArrayList<>(Arrays.asList("I'm not sure I caught you: was it yes
or no?",
                "Funny, I still don't understand, is it yes or no?",
                "Oh, it's too complicated for me: just tell me yes or no.",
                "Could you please simply say yes or no?",
                "Oh, no, don't try to confuse me: say yes or no."));
        boolean isAnswerYes = yesAnswers.stream().anyMatch(s -> s.equals(userAnswer.toLowerCase()));
        boolean isAnswerNo = noAnswers.stream().anyMatch(s -> s.equals(userAnswer.toLowerCase()));
        if (isAnswerYes) {
            return "You answered: Yes";
        } else if (isAnswerNo) {
            return "You answered: No";
        } else {
```

```
            Collections.shuffle(clarificationQuestion);
            return clarificationQuestion.stream().findAny().get();
        }
    }

    public static String sayBye() {
        List<String> goodByes = new ArrayList<>(Arrays.asList("See you soon!", "Bye!", "Good bye", "Have a nice
day!"));
        Collections.shuffle(goodByes);
        return goodByes.stream().findAny().get();
    }

    public static String checkIfAnswerHasDot(String userAnswer) {
        Pattern pattern = Pattern.compile("[.,!]+$");
        Matcher matcher = pattern.matcher(userAnswer);
        if (matcher.find()) {
            return userAnswer.toLowerCase().substring(0, userAnswer.length() - 1).trim();
        }
        return userAnswer.toLowerCase().trim();
    }
}
```

✓ Correct.

It was a tricky task, but you nailed it!

14 users liked this problem. 0 didn't like it. **What about you?**

😍 🙂 😐 🙁 😡

Continue

Solutions (38)

Comments (15) Hints (3) Useful links (0) Solutions (38) Show discussion