MAGPIE LAB ANSWERS

getResponse()

Did you finish the code?

If not…

What information was missing?

What were you unable to figure out?

If so…

Copy and paste your code here

How similar is your code to the one provided?

Is your code more or less efficient?

What could you have done differently?

What did you learn from comparing your code to the provided answer?

I have indeed finished the code:

public class Magpie2

{

public String getGreeting()

{

return "Hello, let's talk.";

}

public String getResponse(String statement)

{

String response = "";

if (findKeyword(statement, "no", -1) >= 0)

{

response = "Why so negative?";

}

else if (statement.length() == 0)

{

response = "Say something, please.";

}

else if (findKeyword(statement, "mother") >= 0

|| findKeyword(statement, "father") >= 0

|| findKeyword(statement, "sister") >= 0

|| findKeyword(statement, "brother") >= 0)

{

response = "Tell me more about your family.";

}

else if (findKeyword(statement, "cat") >= 0

|| findKeyword(statement, "dog") >= 0

|| findKeyword(statement, "fish") >= 0

|| findKeyword(statement, "turtle") >= 0)

{

response = "Tell me more about your pet.";

}

else if (findKeyword(statement, "robinette", 0) >= 0)

{

response = "He sounds like a pretty dank teacher.";

}

else

{

response = getRandomResponse();

}

return response;

}

private int findKeyword(String statement, String goal, int startPos)

{

String phrase = statement.trim().toLowerCase();

String before = " ";

String after = " ";

int psn = phrase.indexOf(goal, startPos);

while(psn >= 0)

{

before = " ";

after = " ";

if(psn > 0)

{

before = phrase.substring(psn - 1, psn);

}

if(goal.length() + psn < phrase.length())

{

after = phrase.substring(psn + goal.length(), psn + goal.length() + 1); }

if((before.compareTo("a") < 0 || before.compareTo("z") > 0 )&&

(after.compareTo("a") < 0 || after.compareTo("z") > 0))

{

return psn;

}

psn = phrase.indexOf(goal, psn+1);

}

return -1;

}

private int findKeyword(String phrase, String goal)

{

return findKeyword(phrase, goal, 0);

}

private String getRandomResponse()

{

final int NUMBER\_OF\_RESPONSES = 4;

double r = Math.random();

int whichResponse = (int)(r \* NUMBER\_OF\_RESPONSES);

String response = "";

if (whichResponse == 0)

response = "Interesting, tell me more.";

else if (whichResponse == 1)

response = "Hmmm.";

else if (whichResponse == 2)

response = "Do you really think so?";

else if (whichResponse == 3)

response = "You don't say.";

return response;

}

}

My code is incredibly similar to the code provided. While there are minor cosmetic differences, the way the code is handled, processed, and returned is exactly the same.

The code provided is more efficient by 3 lines, as they do not instantiate before and after anywhere outside the while loop and they do it on the same line.

As for what to do differently, I would have had a print function for the getRandomResponse() function so as to mimimize the space it takes up if more lines were added. The perameters would have been what the response would equal.

Otherwise, no major changes would have been made, but I would have changed the goal words because the code is super unrealistic currently, and functions nothing like a chat response system. Bird should be under pet, for example, and the random responses are especially arbitrary and unrealistic.

I learned that variables can be instantiated in while loops, and on the same line.

findKeyword()

Did you finish the code?

If not…

What information was missing?

What were you unable to figure out?

If so…

Copy and paste your code here

How similar is your code to the one provided?

Is your code more or less efficient?

What could you have done differently?

What did you learn from comparing your code to the provided answer?

private int findKeyword(String statement, String goal, int startPos)

{

String phrase = statement.trim().toLowerCase();

String before = " ";

String after = " ";

int psn = phrase.indexOf(goal, startPos);

while(psn >= 0)

{

before = " ";

after = " ";

if(psn > 0)

{

before = phrase.substring(psn - 1, psn);

}

if(goal.length() + psn < phrase.length())

{

after = phrase.substring(psn + goal.length(), psn + goal.length() + 1); }

if((before.compareTo("a") < 0 || before.compareTo("z") > 0 )&&

(after.compareTo("a") < 0 || after.compareTo("z") > 0))

{

return psn;

}

psn = phrase.indexOf(goal, psn+1);

}

return -1;

}

My code is incredibly similar to the code provided. While there are minor cosmetic differences, the way the code is handled, processed, and returned is exactly the same.

The code provided is more efficient by 3 lines, as they do not instantiate before and after anywhere outside the while loop and they do it on the same line.

As for what to do differently, I literally have no issues here. This was done incredibly efficiently and I feel good about this section of code. The time put into this section was well spent and the code is incredibly concise.

I learned that variables can be instantiated in while loops, and on the same line.

Exercise\_03

Paste your code here and explain how it works.

else if (findKeyword(statement, "cat") >= 0

|| findKeyword(statement, "dog") >= 0

|| findKeyword(statement, "fish") >= 0

|| findKeyword(statement, "turtle") >= 0)

{

response = "Tell me more about your pet.";

}

else if (findKeyword(statement, "robinette", 0) >= 0)

{

response = "He sounds like a pretty dank teacher.";

}

This is simply literally the same code from the “mother” “father” section. It plugs into findKeyword() with the parameters of the word you want to find as goal and the statement as what to look for it in, starting from 0. findKeyword() does its job and if it returns anything above -1, which it will if the word is inside the statement, then the appropriate response is printed.

It is simple and concise.