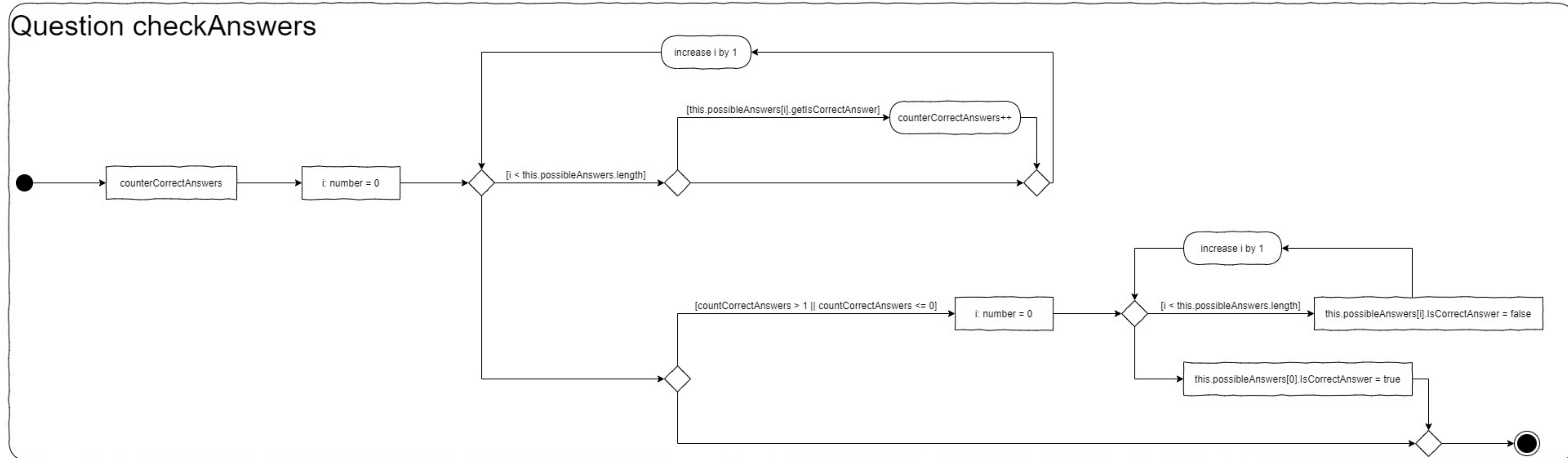


Input: Kimi no Na wa y

Output: Richtig



generateNewQuestion

```

    graph LR
      Start(( )) --> Prompt1[["prompt to question, default: "Welcher der folgenden Begriffe bezeichnet keine große Saurierperiode der Weltgeschichte?"]]
      Prompt1 --> NewQuestionText[["newQuestionText: string"]]
      NewQuestionText --> ParseInt[["parseInt(prompt to question, default)"]]
      ParseInt --> AnswerCount[["answerCount: number"]]
      AnswerCount --> Decision1{ }
      Decision1 -- "[isNaN(answerCount) || answerCount > 5 || answerCount <= 0]" --> AnswerCount2[["answerCount = 2"]]
      AnswerCount2 --> Decision2{ }
      Decision2 --> NewQuestion[["newQuestion: Question"]]
      NewQuestion --> INumber[["i: number = 0"]]
      INumber --> Decision3{ }
      Decision3 -- "[i < answerCount]" --> Prompt2[["prompt to question, default: "Wie lautet die " + (i + 1) + ' Antwort?'  
"Markieren sie die richtige Antwort mit einem Leerzeichen und 'y' an letzter Stelle)  
"z.B. '28 y' in Hinweis. Setzen sie dieses Flag bei keiner der Antwortmöglichkeiten, wird automatisch die 1. als richtig markiert"  
default: "26"]]
      Prompt2 --> NewAnswerText[["newAnswerText: string"]]
      NewAnswerText --> Decision4{ }
      Decision4 -- "newAnswerText.slice(-2) == 'r'" --> NewAnswerTextSub[["newAnswerText = newAnswerText.substring(0, newAnswerText.length - 2)"]]
      NewAnswerTextSub --> NewPossibleAnswersTrue[["newQuestion.possibleAnswers[i] = new Answer(newAnswerText, true)"]]
      Decision4 --> NewPossibleAnswersFalse[["newQuestion.possibleAnswers[i] = new Answer(newAnswerText, false)"]]
      NewPossibleAnswersTrue --> Decision5{ }
      NewPossibleAnswersFalse --> Decision5
      Decision5 --> IncreaseI[["increase i by 1"]]
      IncreaseI --> Decision3
      Decision3 -- "[i < answerCount]" --> CheckAllAnswers[["newQuestion.checkAllAnswers"]]
      CheckAllAnswers --> Dispatch[["dispatch"]]
      Dispatch --> PushNewQuestion[["preDefinedQuestions.push(newQuestion)"]]
      PushNewQuestion --> End(( ))
  
```

The diagram illustrates the process of generating a new question. It starts with a prompt to the user, which is then parsed to determine the number of possible answers. If the number is not a valid integer or is outside the range of 1 to 5, it defaults to 2. The process then generates a new question object. A loop is used to generate multiple questions, with the number of questions determined by the 'answerCount' variable. The loop generates a prompt for each question, which is then used to generate a new answer text. The answer text is then used to generate a new possible answer for the question. The process continues until all questions have been generated, at which point the questions are pushed into a predefined list.

```

graph LR
    Start(( )) --> Init[randomQuestionInd: number = Math.floor(Math.random() * preDefinedQuestions.length)]
    Init --> Select[randomQuestion: Question = preDefinedQuestions[randomQuestionInd]]
    Select --> Shuffle[preDefinedQuestions.splice(randomQuestionInd, 1)]
    Shuffle --> Len[randomQuestionAnswersLength: number = randomQuestion.possibleAnswers.length]
    Len --> InitAns[randomQuestionAnswersAsString: string = ""]
    InitAns --> Dispatch[dispatch]
    Dispatch --> LoopStart[i: number = 0]
    LoopStart --> LoopCond{i < randomQuestionAnswersLength}
    LoopCond --> LoopBody[randomQuestionAnswersAsString += "Antwort: (" + 1) + " - " + randomQuestion.possibleAnswers[i].answerText + "<br>"]
    LoopBody --> LoopInc[increase i by 1]
    LoopInc --> LoopCond
    LoopCond --> Parse[parseInt(prompt to question, default)]
    Parse --> UserAns[userAnswer: number]
    UserAns --> ValidCond{[!NaN(userAnswer) || userAnswer > randomQuestionAnswersLength || userAnswer <= 1]}
    ValidCond --> SetOne[userAnswer = 1]
    SetOne --> ValidCond
    ValidCond --> EvalCond{ }
    EvalCond --> OutRicht[output: "richtig"]
    EvalCond --> OutFalsch[output: "falsch"]
    OutRicht --> Correct[correctAnswers++]
    OutFalsch --> Correct
    Correct --> GetChoice[getUserChoice]
    GetChoice --> End(( ))

```