



Bundeswettbewerb
Informatik

33. BUNDESWEITBEWERB INFORMATIK

BUNDES
WEIT
INFORMATIK
NACHWUCHS
FÖRDERN



Digitale Technologien durchdringen unseren Alltag. Ein Leben ohne Handy und Computer, ohne Navigationssystem im Auto und ohne digitale medizinische Diagnostik ist heutzutage kaum noch vorstellbar. Auch für den Umbau der Energieversorgung und die Entwicklung integrierter Mobilitätssysteme brauchen wir Informatik.

Das Wissenschaftsjahr 2014 „Die digitale Gesellschaft“ befasst sich mit der Frage, was die Digitalisierung für unser Leben bedeutet. Dabei setzen wir uns sowohl mit den Chancen als auch mit den Herausforderungen auseinander, die mit der Entwicklung der digitalen Technologien verbunden sind. Ziel ist, technologische Zukunftsprozesse und gesellschaftliche Veränderungen mitzugestalten.

Die Bundesweiten Informatikwettbewerbe bieten eine gute Gelegenheit, auf besondere Weise in die faszinierende Welt der Informatik einzutauchen. Im Bundeswettbewerb Informatik können Schülerinnen und Schüler der Bedeutung von Informatik nachspüren und sich mit technologischen Zusammenhängen vertraut machen. Auch die anderen Informatikwettbewerbe wie der Informatik-Biber wecken Neugier für dieses Fach. Über die große Resonanz freue ich mich sehr. Über 200.000 Jugendliche haben im vergangenen Jahr am Informatik-Biber teilgenommen. Alle, die durch den Informatik-Biber Lust auf die Welt der Bits und Bytes bekommen haben, können ihr Interesse und ihr Wissen beim Bundeswettbewerb Informatik vertiefen.

Die Informatikwettbewerbe tragen dazu bei, die große Bedeutung von Informatik für die Entwicklung unserer Gesellschaft zu vermitteln. Den Teilnehmerinnen und Teilnehmern wünsche ich viel Spaß und gute Ideen.

Prof. Dr. Johanna Wanka
Bundesministerin für Bildung und Forschung

Gesellschaft für Informatik e.V. (GI)

Die Gesellschaft für Informatik e.V. (GI) ist mit rund 20.000 Mitgliedern die größte Fachgesellschaft der Informatik im deutschsprachigen Raum. Ihre Mitglieder kommen aus allen Sparten der Wissenschaft, aus der Informatikindustrie, aus dem Kreis der Anwender sowie aus Lehre, Forschung, Studium und Ausbildung. In der GI wirken Männer und Frauen am Fortschritt der Informatik mit, im wissenschaftlich-fachlich-praktischen Austausch in etwa 120 verschiedenen Fachgruppen und mehr als 30 Regionalgruppen. Ihr gemeinsames Ziel ist die Förderung der Informatik in Forschung, Lehre und Anwendung, die gegenseitige Unterstützung bei der Arbeit sowie die Weiterbildung. Die GI vertritt hierbei die Interessen der Informatik in Politik und Wirtschaft.

www.gi.de

Fraunhofer-Verbund IuK-Technologie

Als größter europäischer Forschungsverbund für Informations- und Kommunikationstechnik (IuK) versteht sich der Fraunhofer-Verbund IuK-Technologie als Anlaufstelle für Industriekunden auf der Suche nach dem richtigen Ansprechpartner in der anwendungsorientierten IT-Forschung. Die Vernetzung der 5000 Mitarbeiter in bundesweit 18 Instituten ermöglicht die Entwicklung übergreifender branchenspezifischer IT-Lösungen, oft zusammen mit Partnern aus der Industrie, sowie anbieterunabhängige Technologieberatung. Entwickelt werden IuK-Lösungen für die Geschäftsfelder Medizin, Automotive, Produktion, Digitale Medien, E-Business, E-Government, Finanzdienstleister, Sicherheit sowie IT und Kommunikationssysteme. InnoVisions – Das Zukunftsmagazin des Fraunhofer Verbundes IuK-Technologie informiert Sie über aktuelle Forschungsprojekte auf www.innovisions.de. Weitere Informationen über den Fraunhofer IuK-Verbund gibt es auf www.iuk.fraunhofer.de

Max-Planck-Institut für Informatik

Eine der größten Herausforderungen der Informatik ist die robuste und intelligente Suche nach Information, die grundlegendes Verständnis und automatische Organisation der gewünschten Inhalte voraussetzt. Das Max-Planck-Institut für Informatik widmet sich seit seiner Gründung 1990 diesen Fragestellungen. Das Spektrum der Forschung reicht von allgemeinen Grundlagen der Informatik bis hin zu konkreten Anwendungsszenarien und umfasst Algorithmen und Komplexität, Automatisierung der Logik, Bioinformatik und Angewandte Algorithmik, Computergrafik, Bildverarbeitung und multimodale Sensorverarbeitung sowie Datenbanken und Informationssysteme.

Das Max-Planck-Institut für Informatik unterstützt nachhaltig junge Forscher, die am Institut die Möglichkeit bekommen, ihr eigenes Forschungsgebiet und ihre eigene Gruppe zu entwickeln. Das Institut wirkt seit mehr als 20 Jahren auf Weltklassenniveau durch Publikationen und Software und durch seine jetzigen und ehemaligen Forscher, die Führungsrollen in Wissenschaft und Industrie übernommen haben.

www.mpi-inf.mpg.de



Unter der Schirmherrschaft des Bundespräsidenten

Von der Kultusministerkonferenz empfohlener Schülerwettbewerb



GEFÖRDERT VOM



Die Partner

Zusätzlich zur Förderung durch das Bundesministerium für Bildung und Forschung und seine Träger erfahren die Bundesweiten Informatikwettbewerbe (BWINF) und insbesondere der Bundeswettbewerb Informatik weitere Unterstützung durch viele Partner. Sie stiften Preise und bieten vor allem spannende Informatik-Workshops für Wettbewerbsteilnehmer an.



Die BWINF-Partner wünschen allen Teilnehmerinnen und Teilnehmern des 33. Bundeswettbewerbs Informatik viel Erfolg!

Bundeswettbewerb Informatik

Der Bundeswettbewerb Informatik (BwInf) wurde 1980 von der Gesellschaft für Informatik e.V. (GI) auf Initiative von Prof. Dr. Volker Claus ins Leben gerufen. Ziel des Wettbewerbs ist, Interesse an der Informatik zu wecken und zu intensiver Beschäftigung mit ihren Inhalten und Methoden sowie den Perspektiven ihrer Anwendung anzuregen. Der Bundeswettbewerb Informatik ist der traditionsreichste unter den Bundesweiten Informatikwettbewerben (BWINF), zu denen auch Informatik-Biber und das deutsche Auswahlverfahren zur Internationalen Informatik-Olympiade gehören. BWINF wird vom Bundesministerium für Bildung und Forschung gefördert; die Träger sind GI, Fraunhofer-Verbund IuK-Technologie und Max-Planck-Institut für Informatik. Die Bundesweiten Informatikwettbewerbe gehören zu den bundesweiten Schülerwettbewerben, die von den Kultusministerien der Länder empfohlen werden. Der Bundeswettbewerb Informatik steht unter der Schirmherrschaft des Bundespräsidenten.

Die Gestaltung des Wettbewerbs und die Auswahl der Sieger obliegen dem Beirat; Vorsitzender: Prof. Dr. Till Tantau, Universität Lübeck. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt der Aufgabenausschuss; Vorsitzender: Prof. Dr. Peter Rossmanith, RWTH Aachen. Die BWINF-Geschäftsstelle mit Sitz in Bonn ist für die fachliche und organisatorische Durchführung zuständig; Geschäftsführer: Dr. Wolfgang Pohl.

Drei Runden

Der Wettbewerb beginnt jedes Jahr im September, dauert etwa ein Jahr und besteht aus drei Runden. In der ersten und zweiten Runde sind die Wettbewerbsaufgaben zu Hause selbstständig zu bearbeiten. Dabei können die Aufgaben der ersten Runde mit guten grundlegenden Informatikkenntnissen gelöst werden; die Aufgaben der zweiten Runde sind deutlich schwieriger. In der ersten Runde ist Gruppenarbeit zugelassen und erwünscht. In der zweiten Runde ist dann eigenständige Einzelarbeit gefordert; die Bewertung erfolgt durch eine relative Platzierung der Arbeiten. Die bis zu dreißig bundesweit Besten der zweiten Runde werden zur dritten Runde, einem Kolloquium, eingeladen. Darin führt jeder Gespräche mit Informatikern aus Schule und Hochschule und bearbeitet im Team zwei Informatik-Probleme.

Juniorliga

Für Schüler, die nicht älter als 16 Jahre und nicht in der Qualifikationsphase der gymnasialen Oberstufe sind, werden zwei leichtere Aufgaben gestellt, die Junioraufgaben. Stammt eine Einsendung von Schülern, die alle die Altersgrenze für Junioraufgaben erfüllen, nehmen die darin bearbeiteten Junioraufgaben in der Juniorliga teil (wenn auch andere Aufgaben bearbeitet sind, nimmt die vollständige Einsendung zusätzlich in der Hauptliga teil). Die Juniorliga wird getrennt bewertet, Preise werden separat vergeben.

Biber goes BwInf
Teilnehmerinnen und Teilnehmer am Informatik-Biber sollen dazu angeregt werden, auch beim 33. Bundeswettbewerb Informatik mitzumachen. Deshalb werden Schulen, die unter ihren BwInf-Erstteilnehmern mindestens fünf ehemalige Biber-Teilnehmer nachweisen können, mit einem besonderen „Biber-goes-BwInf“-Schulpreis ausgezeichnet. Die für den Preis gewerteten Schülerinnen und Schüler erhalten als Anerkennung einen „Biber goes BwInf“-USB-Stick. Weitere Informationen zu dieser Aktion gibt es online: für Schüler (bwinf.de/bgb/schueler) und Lehrkräfte (bwinf.de/bgb/lehrer)



Die Chancen

Preise

In allen Runden des Wettbewerbs wird die Teilnahme durch eine Urkunde bestätigt. In der ersten Runde werden auf den Urkunden erste und zweite Preise sowie Anerkennungen unterschieden; mit einem Preis ist die Qualifikation für die zweite Runde verbunden. In der zweiten Runde gibt es erste, zweite und dritte Preise; jüngere Teilnehmer haben die Chance auf eine Einladung zu einer Schülerakademie. Ausgewählte Gewinner eines zweiten Preises erhalten einen Buchpreis des Verlags O'Reilly; erste Preisträger werden zur dritten Runde eingeladen, die im Herbst 2015 von der TU Darmstadt ausgerichtet wird.

Die dort ermittelten Bundessieger werden in der Regel ohne weiteres Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen. Zusätzlich sind für den Bundessieger, aber auch für andere besondere Leistungen Geld- und Sachpreise vorgesehen.

Informatik-Olympiade

Ausgewählte Teilnehmerinnen und Teilnehmer können sich in mehreren Trainingsrunden für das vierköpfige deutsche Team qualifizieren, das an der Internationalen Informatik-Olympiade 2016 in Russland teilnimmt.

Informatik-Workshops etc.

Informatik-Workshops exklusiv für TeilnehmerInnen werden in Baden-Württemberg, vom Hasso-Plattner-Institut, von Hochschulen wie der RWTH Aachen, der TU Dortmund, der TU Braunschweig und der LMU München (gemeinsam mit der QAware GmbH), von der Firma INFORM sowie vom Max-Planck-Institut für Informatik (2. Runde) veranstaltet. Bei einigen von Fraunhofer-Instituten veranstalteten „Talent Schools“ gibt es reservierte BwInf-Plätze. Die Firma Google lädt ausgewählte Teilnehmerinnen zum „Girls@Google Day“ ein.

Ausgewählte Endrundenteilnehmer werden im September 2015 vom Bundesministerium für Bildung und Forschung zum „Tag der Talente“ eingeladen.

Eine Einsendung zur zweiten Runde kann in vielen Bundesländern als besondere Lernleistung in die Abiturwertung eingebracht werden.

Preise für BwInf-Schulen

Für eine substantielle Beteiligung am Wettbewerb werden Schulpreise vergeben: An mindestens 3 vollwertigen Einsendungen (also mit je mindestens 3 bearbeiteten Aufgaben) zur 1. Runde – oder an 2 vollwertigen Einsendungen und 2 weiteren Einsendungen in der Juniorliga – müssen mindestens 10 Schülerinnen und Schüler einer Schule, darunter bei gemischten Schulen mindestens 2 Jungen und mindestens 2 Mädchen, beteiligt sein. **Neu:** Mindestens eine der gewerteten Einsendungen muss in Hauptliga oder Juniorliga mit einem ersten oder zweiten Preis ausgezeichnet werden.

Schulen, die diese Bedingungen erfüllen, werden als „BwInf-Schule 2014/2015“ ausgezeichnet: sie erhalten ein entsprechendes Zertifikat, ein Label zur Nutzung auf der Schul-Website und einen Gutschein im Wert von **300 Euro** für Bücher oder andere für den Informatikunterricht benötigte Dinge.

Die Regeln

Teilnahmeberechtigt

... sind Jugendliche, die nach dem 24.11.1992 geboren wurden. Sie dürfen jedoch zum 1.9.2014 noch nicht ihre (informatikbezogene) Ausbildung abgeschlossen oder eine Berufstätigkeit begonnen haben. Personen, die im Wintersemester 2014/15 an einer Hochschule studieren, sind ausgeschlossen, falls sie nicht gleichzeitig noch die Schule besuchen. Jugendliche, die nicht deutsche Staatsangehörige sind, müssen wenigstens vom 1.9. bis 24.11.2014 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Junioraufgaben dürfen von Jugendlichen bearbeitet werden, die jünger als 17 Jahre alt sind (geboren nach dem 24.11.1997) und, falls sie die Schule besuchen, noch nicht in der Qualifikationsphase der Sekundarstufe 2 sind. Ein Team darf Junioraufgaben bearbeiten, wenn mindestens ein Mitglied des Teams die genannten Bedingungen erfüllt.

Weiterkommen

An der zweiten Runde dürfen jene teilnehmen, die allein oder mit ihrem Team wenigstens drei Aufgaben der ersten Runde weitgehend richtig gelöst haben. Für die dritte Runde qualifizieren sich die besten ca. 30 Teilnehmer der zweiten Runde. In der Juniorliga gibt es voraussichtlich noch keine zweite Runde.

Einsendungen

... enthalten Bearbeitungen zu mindestens einer Aufgabe und werden von Einzelpersonen oder Gruppen (Teams) abgegeben. Eine Einsendung besteht für jede bearbeitete Aufgabe aus **Dokumentation** und (bei Aufgaben mit Programmierauftrag) Implementierung. Die Dokumentation enthält eine Beschreibung der Lösungsidee und Beispiele, welche die Korrektheit der Lösung belegen. Ist ein Programm gefordert, sollen außerdem die Umsetzung der Lösungsidee in das Programm erläutert und die wichtigsten Teile des Quelltextes hinzugefügt werden. Achtung: eine gute Dokumentation muss nicht lang sein! Die **Implementierung** umfasst das (möglichst eigenständig lauffähige) Programm selbst und den kompletten Quelltext des Programms.

Die **Einsendung** wird über das Online-Anmeldesystem als Dateiarchiv im ZIP-Format abgegeben. Dieses Archiv muss zu jeder bearbeiteten Aufgabe auf oberster Ebene enthalten:

- > die Dokumentation: ein PDF-Dokument;
- > die Implementierung: einen Ordner mit Programm- und Quelltextdateien.

Anmeldung

Die Anmeldung ist bis zum Einsendeschluss möglich, und zwar online über: pms.bwinf.de
Wettbewerbsteilnehmer können sich dort eigenständig registrieren, zum Wettbewerb anmelden und ggf. Teams bilden. Die Anmeldung zum Wettbewerb und das Bilden von Teams kann auch **von Lehrkräften vorgenommen** werden.

Einsendeschluss: 24.11.2014

Verspätete Einsendungen können nicht berücksichtigt werden. Der Rechtsweg ist ausgeschlossen. Die Einsendungen werden nicht zurückgegeben. Der Veranstalter erhält das Recht, die Beiträge in geeigneter Form zu veröffentlichen.

Beispiellösung: Songwriter

Hinweis:

Der Aufgabentext wird hier nur der Vollständigkeit halber abgedruckt. Die Dokumentation zu einer Aufgabenbearbeitung muss und soll den Aufgabentext nicht enthalten.

Das Duo „Fake that“ ist der aktuelle Stern am Pop-Himmel. Sie bringen einen Hit nach dem anderen heraus, aber nach einiger Zeit dämmert es selbst ihren größten Fans, dass die Texte ihrer Silben-Songs (so nennt das Duo seinen Stil) sich alle sehr ähneln. Das letzte Album hieß wohl nicht umsonst „It's Always The Same“.



Nellie, ein mittlerweile etwas gelangweilter Fan der beiden, hat festgestellt, wie die Texte der Silben-Songs funktionieren:

- > Eine Silbe wird aus einem Konsonanten und einem Vokal gebildet. Beispiele: 'do', 'nu', 'la'.
- > Eine Zeile besteht aus einer ungeraden Anzahl von Wiederholungen einer Grundsilbe, wobei derjenigen in der Mitte ein 'p di' angehängt wird. Beispiele: 'sup di', 'da dap di da', 'ne ne nep di ne ne'.
- > Eine Strophe besteht aus mindestens zwei Zeilen. Die Menge der Konsonanten und Vokale, aus denen die Grundsilben einer Strophe gebildet werden, ist immer sehr klein, z. B. {s, u, a}. Die Anzahl der Silbenwiederholungen ist für alle Zeilen einer Strophe gleich (Nellie spricht deshalb von der Silbenzahl einer Strophe). Am Ende einer Strophe steht gelegentlich ein markiger Call wie 'yeah!', 'yo man', 'fake that!' oder ähnlich pseudo-cooles Zeug.
- > Ein Song besteht aus mindestens zwei Strophen. Die Zeilenzahlen der Strophen eines Songs folgen einem Muster, z. B. „immer 3 Zeilen“, „abwechselnd 4 und 6 Zeilen“ usw. Auch die Silbenzahlen der Strophen folgen ähnlichen Mustern.

Nellie zweifelt nun an der Kreativität von „Fake that“. Solche Songtexte kann man sich bestimmt vom Computer schreiben lassen!

Juniaraufgabe

Schreibe einen „Songwriter“, also ein Programm, das einen nach Nellies Regeln aufgebauten Text eines Silben-Songs erzeugen kann. Stelle dabei so weit wie möglich sicher, dass die von deinem Programm erzeugten Texte sich voneinander unterscheiden. Gib in der Dokumentation mindestens drei unterschiedliche Songtexte an, die dein Programm erzeugt hat.

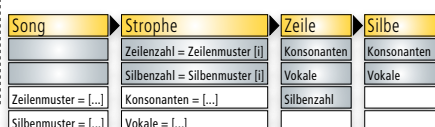
Lösungsidee

Die Lösung soll so funktionieren wie die Einheiten eines Silben-Songs beschrieben sind. Folgende, aufeinander aufbauende Einheiten gibt es: Silbe, Zeile, Strophe und Song. Für jede Song-Einheit soll ein eigener Programmteil entwickelt werden, der diese nach Vorschrift erzeugt. Dabei bauen auch die Programmteile aufeinander auf: 'Song' verwendet 'Strophe', 'Strophe' verwendet 'Zeile' und 'Zeile' verwendet 'Silbe'.

Die Programmteile sind nicht unabhängig voneinander:

- > Die Anzahl der Zeilen einer Strophe und auch die Anzahl der Silben in den Zeilen der Strophe sollen je einem Muster folgen. Wenn ein Muster eine Folge von Zahlen ist (eine Zahl pro Strophe), kann 'Song' jeder 'Strophe' die passende Zahl übergeben.
- > Für eine Strophe müssen die Mengen von Konsonanten und Vokalen bestimmt werden, aus denen die Silben der Zeilen bestehen müssen. 'Strophe' muss die beiden Buchstabenmengen an 'Zeile' übergeben (und 'Zeile' wiederum an 'Silbe').

Das folgende Bild stellt die Zusammenarbeit der verschiedenen Programmteile dar.



Für eine Strophe muss zusätzlich noch festgelegt werden, ob sie einen „Call“ hat. Wenn ja, wird einer ausgewählt und als zusätzliche Zeile hinten an die Strophe angefügt.

Schließlich muss noch beachtet werden, dass „die von deinem Programm erzeugten Texte sich voneinander unterscheiden“. Dazu kann bei der Bestimmung der Muster, der Konsonanten und Vokale einer Strophe, der Auswahl des Calls, der Auswahl von Konsonant und Vokal für eine Silbe usw. der Zufall eingesetzt werden.

Umsetzung

Die Lösungsidee wird in ein Programm in der Sprache Python umgesetzt. Für jede Song-Einheit wird eine Funktion geschrieben, jeweils mit den nötigen Parametern:

```
Song: def song()
Strophe: def strophe(anzahl_zeilen, anzahl_silben)
Zeile: def zeile(anzahl_silben, konsonanten, vokale)
Silbe: def silbe(konsonanten, vokale)
```

Außerdem wird eine Funktion benötigt, die die Muster (also die Zahlenfolgen) erzeugt. Die Funktion `muster(laenge, grundmenge)` liefert eine Folge von Elementen einer Grundmenge (z. B. der möglichen Zeilenanzahlen), wobei die Folge eine bestimmte Länge haben muss (z. B. die Anzahl der Strophen eines Songs). Die Folge ist zufällig von einer dieser drei Arten: (1) zwei sich abwechselnde Elemente, (2) alle Elemente gleich und (3) die Elemente der Grundmenge (nötigenfalls wiederholt, wenn die Grundmenge kürzer ist als die gewünschte Folgenlänge).

Die Zufallseffekte werden durch verschiedene Funktionen des Python-Moduls `random` erzielt:

```
random.choice(folge): wählt zufällig ein Element der Folge.
random.sample(folge, anzahl): wählt zufällig eine Anzahl von Elementen der Folge.
random.randint(int1, int2): wählt zufällig eine Integer-Zahl aus der Menge {int1, ..., int2}.
```

Der Song wird in der Funktion `strophe` zeilenweise ausgegeben.

Beispiele

Durch einen Aufruf der Funktion `song` wird in der Python-Shell ein Songtext ausgegeben. Aus Platzgründen werden nur zwei Beispiele gezeigt.

```
>>> song()
da da dap di da da
bi bi bip di bi bi
```

```
bu bu bup di bu bu
do do dop di do do
Bwlnf rocks!
```

Dieser Song besteht aus zwei Strophen. Das Strophenmuster ist „immer zwei Zeilen“, das Silbenmuster ist „immer fünf Silben“. In Strophe 1 kommen die Konsonanten d und b sowie die Vokale a und i vor, in Strophe 2 b und d bzw. u und o. Strophe 2 hat einen Call.

```
>>> song()
li lip di li
li lip di li

le le lep di le le
sa sa sap di sa sa
se se sep di se se
sa sa sap di sa sa

gi gi gi gip di gi gi gi
bo bo bo bop di bo bo bo
yeah!
```

Dieser Song hat das Zeilenmuster [2,4,2] (zwei Zahlen abwechselnd) und das Silbenmuster [3,5,7] (die möglichen Silbenzahlen nacheinander). Nur die letzte Strophe hat einen Call.

Quelltext

```
import random
```

```
vokale = ['a','e','i','o','u']
# nur geeignete Konsonanten - Geschmackssache
konsonanten = ['b','d','g','l','n','s']
# mögliche Calls - auch 'leere' Calls
calls = ["yeah!", "fake that!", "Bwlnf rocks!", "", "", ""]
# mögliche Silbenanzahlen in den Zeilen - müssen ungerade sein
silbenzahlen = [3,5,7]
zeilenzahlen = [2,3,4,5] # mögliche Zeilenanzahlen in den Strophen
strophenzahlen = [2,3,4,5] # mögliche Strophenzahlen für einen Song
```

```
def silbe(konsonanten, vokale):
    """Bildet eine Silbe aus einem der Konsonanten und einem der Vokale."""
    return random.choice(konsonanten) + random.choice(vokale)
```

```
def zeile(konsonanten, vokale, anzahl_silben):
    """Bildet eine Zeile mit der gegebenen Anzahl von Silben."""
    # Die Silbe wird für die ganze Zeile einmal bestimmt.
    zeilensilbe = silbe(konsonanten, vokale)
    halbezeile = (anzahl_silben // 2) * (zeilensilbe + ' ')
    return halbezeile + zeilensilbe + 'p di ' + halbezeile
```

```
def call():
    """Sucht aus der Menge der Calls einen (eventuell leeren) aus."""
    auswahl = random.randint(0, len(calls)-1)
    return calls[auswahl]
```

```
def strophe(anzahl_zeilen, anzahl_silben):
    """Bildet eine Strophe mit den gegebenen Anzahlen für Zeilen und Silben."""
    anzahl_konsonanten = random.randint(2,3) # zwei oder drei Konsonanten
    auswahl_konsonanten = random.sample(konsonanten, anzahl_konsonanten)
    anzahl_vokale = random.randint(2,3) # zwei oder drei Vokale
    auswahl_vokale = random.sample(vokale, anzahl_vokale)
    for i in range(anzahl_zeilen):
        print(zeile(anzahl_silben, auswahl_konsonanten, auswahl_vokale))
    stropfen_call = call()
    # ein nicht-leerer Call wird an die Strophe angehängt
    if stropfen_call != "":
        print(stropfen_call)
    print("") # eine leere Zeile am Ende der Strophe
```

```
def muster(laenge, grundfolge):
    """Wählt eine Folge (der gegebenen Laenge) von Elementen der gegebenen Grundfolge auf drei verschiedene Arten aus."""
    folge = []
    auswahl = random.randint(1,3)
    if auswahl == 1: # Art 1: abwechselnd zwei verschiedene Elemente
        elementpaar = random.sample(grundfolge, 2) # zwei Elemente auswählen
        for i in range(laenge):
            folge.append(elementpaar[i%2]) # i%1 ergibt abwechselnd 0 und 1
    elif auswahl == 2: # Art 2: immer das gleiche Element
        element = random.choice(grundfolge) # Element auswählen
        for i in range(laenge):
            folge.append(element)
    else: # auswahl == 3; Art 3: Grundmenge der Reihe nach
        for i in range(laenge):
            folge.append(grundfolge[i%len(grundfolge)])
    return folge
```

```
def song():
    """Bildet einen vollständigen Song."""
    stropfenzahl = random.choice(strophenzahlen)
    zeilenzahlmuster = muster(stropfenzahl, zeilenzahlen)
    silbenzahlmuster = muster(stropfenzahl, silbenzahlen)
    for i in range(stropfenzahl):
        strophe(zeilenzahlmuster[i], silbenzahlmuster[i])
```