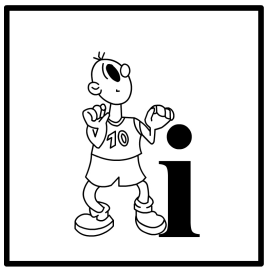
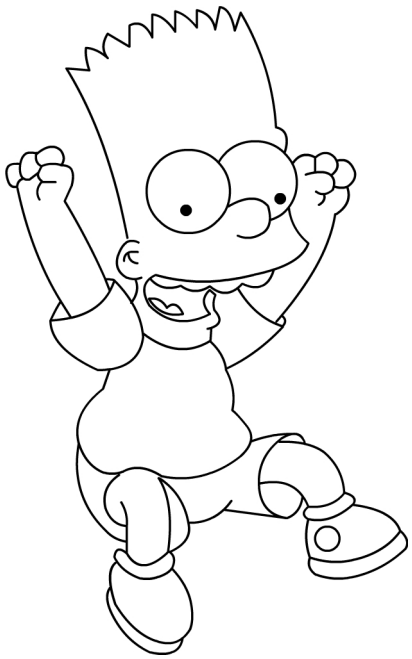
Idee, Zweck, Systembeschreibung:



In der Schule benötigen die *Personen* einen *Geheimnachrichten-Generator.* Hierfür soll es möglich sein, auf einer *Benutzeroberfläche (Hauptfenster)* nach der Eingabe der *Nachricht* (im Eingabebereich) eine *geheime Nachricht erstellen (verschlüsseln)* und auf der Benutzeroberfläche (Hauptfenster) *anzeigen*. Beide Schritte erfolgen durch das Anklicken einer Schaltfläche *Nachricht erstellen (verschlüsseln).* Eine weitere Schaltfläche *Eingaben löschen* soll dafür sorgen, dass alle gemachten Eingaben *gelöscht* werden. Die *Person* kann auch geheime *Nachrichten lesen (entschlüsseln),* indem Sie die dafür vorgesehene Schaltfläche anklickt. Die entschlüsselte *Nachricht* wird daraufhin im Ausgabebereich *angezeigt*. Die Ver- und Entschlüsselung (sind Methoden) der *Nachricht* soll einem bestimmten Schema (ROT13-Algoritmus oder Caesar-Chiffre) folgen. Eine Erklärung dazu folgt im Anschluss an die vorgegebene System-Architektur.

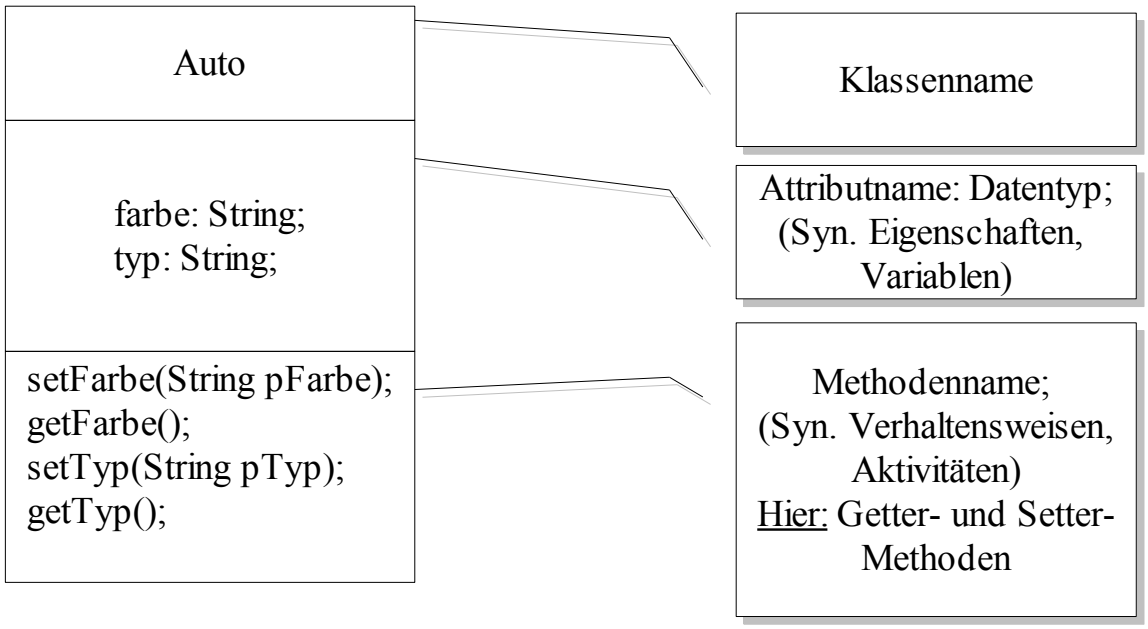
Anwendungsfälle:

		
Bart Simpson ist *Schüler des Wirtschaftsgymnasiums der KSW.* Er gibt die *Nachricht* als *Eingabe* „Hallo Franz ich bin im Computerraum“ ein und erhält die *Ausgabe* „Unyyb Senam vpu ova vz Pbzchgreenhz“.	*Red Barklay* ist *Schüler der Berufsschule.* Er gibt als *Nachricht* „Hallo Moe endlich ist Wochenende“ ein und erhält das *Ausgabe* „Unyyb Zbr raqyvpu vfg Jbpuraraqr“.	*Moe Szyslak* ist *Schüler des Berufskollegs der KSW.* Er gibt die *Nachricht* „Unyyb Zbr raqyvpu vfg Jbpuraraqr“ ein und erhält die *Ausgabe* „Hallo Moe endlich ist Wochenende“.

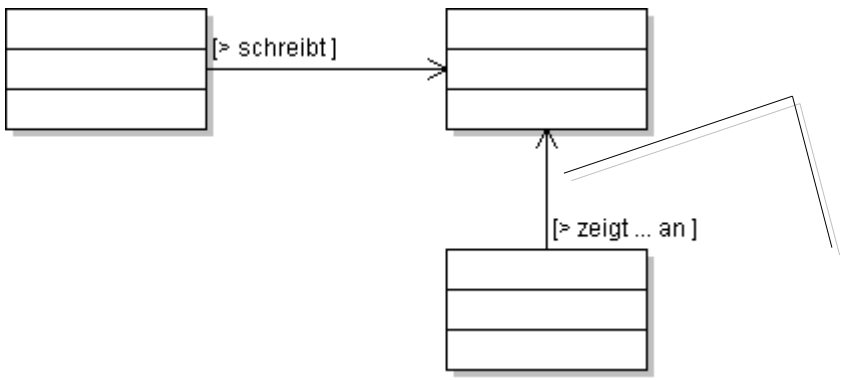
[Grafiken: The Simpsons TM, erstellt von CJ]

Merke: Sogenannte Getter- und Setter-Methoden[1] (Syn. Aktivität, Verhaltensweise) existieren für jede Eigenschaft. Z.B. Die Methoden setFarbe(String pFarbe) und getFarbe() für das Attribut (syn. Variable, Eigenschaft) String farbe der Klasse Auto. Diese Methoden dienen dazu, Eigenschaftswerte einzelner Objekte zu modifizieren (bearbeiten, ändern) bzw. erstmalig zu initialisieren (Wert setzen). Es sind quasi Teilhandlungen auf unterster Ebene (Hinweis: kleinschrittig denken).

Zur Erinnerung die UML-Notation einer Klasse:



Vorschlag für die System Architektur des Geheimnachrichten-Generators:



Hinweis: Ohne diese Verbinder (Pfeile) können die Klassen nicht miteinander kommunizieren.

1 „get“ steht für „holen“, „set“ steht für „setzen“

[Grafiken: The Simpsons TM, erstellt von CJ]

Schema für die Ver- bzw. Entschlüsselung (ROT13-Algorithmus):

Der ROT13-Algoritms (auch Caesar-Chiffre oder Verschiebechiffre) ist ein Verfahren das symmetrisch verschlüsselt. Wenn jemand den Schlüssel kennt, kann er die Nachricht ver- und entschlüsseln.

> **%%Merke:** Das Verschieben entspricht also dem ver- und entschlüsseln! Überprüfen Sie diese Aussage!**%%**

ROT13-Algorithmus: `public void verschieben(){...}`

```
//Buchstaben verschieben
public void verschieben() {
  String sIn = this.getMessageInput();
  String sOut = new String();
      for (int i = 0; i < sIn.length(); i++) {
          char c = sIn.charAt(i);
          if      (c >= 'a' && c <= 'm') c+=13;
          else if (c >= 'n' && c <= 'z') c-=13;
          else if (c >= 'A' && c <= 'M') c+=13;
          else if (c >= 'N' && c <= 'Z') c-=13;
          sOut = sOut + c;
      }
      this.setMessageOutput(sOut);
}
```

Linksverschiebung

Rechtsverschiebung

Erklärung:
Die eingegebene Nachricht wird in sIn vorübergehend gespeichert. Die auszugebende Ausgabe noch leer, aber das lokale Attribut sOut existiert bereits (ist deklariert und initialisiert).

For-Schleife: ist eine Kontrollstruktur die es u.a. erlaubt einen String (Zeichenkette) von Anfang (i = 0) bis hinten (i = 12) zu durchlaufen.

`sIn.charAt()`-Methode: Holt ein Zeichen an der Stelle *i* (*Index*) aus der Zeichenkette *sIn* raus und speichert es vorübergehend in einem Attribut *c* vom Typ *char* (char steht für character und enthält ein *einzelnes Zeichen*).

[Grafiken: The Simpsons TM, erstellt von CJ]

Kontrollstrukturen: Loops mit der FOR-SCHLEIFE

Mit einer *For-Schleife* wird in unserem Fall eine Zeichenkette (die Nachricht, *sIn*) Zeichen für Zeichen durchlaufen. Das *i* steht für Index und bezeichnet die *Stelle in der Zeichenkette*.

Wir lesen diese Schleife wie am folgenden Beispiel erläutert:

```
for(int i = 0; i< sIn.length();i++ ) {
    System.out.println("mache irgendwas...");
}
```

Für den Anfang ist der Zähler 0 *(int i = 0)*. Dann durchlaufen wir die Zeichenkette solange wir nicht am Ende angekommen sind *(solange i < als die Länge der Zeichenkette)* und zählen dann den Zähler um eins hoch *(rücken also um eine Stelle in der Zeichenkette weiter mit i++)*.

Kontrollstrukturen: Fallunterscheidung mit ELSEIF

Bisher haben wir immer nur zwei Fälle unterschieden (erfolgreich oder nicht erfolgreich). Im Falle der Bestimmung der Verschiebung unterscheiden wir X-Fälle (mehrere). Hierfür bietet sich eine neue Kontrollstruktur an, die ELSEIF-Anweisung. Ein Beispiel soll die Verwendung verdeutlichen:

Gewinnkategorie	**Min**	**Max**
hoch	20000	> 20000
mittel	10000	19999
niedrig	9999	0

```
public void wahlDerGewinnkategorie(int pGewinn){
int mGe = pGewinn;
    if(mGe >= 20000){
        this.setGewinnKategorie("hoch");
        System.out.println("Gewinnkategorie: hoch");
    }else if(mGe >= 10000 && mGe <= 19999){
        this.setGewinnKategorie("mittel");
        System.out.println("Gewinnkategorie: mittel");
    }else{
        this.setGewinnKategorie("niedrig");
        System.out.println("Gewinnkategorie: niedrig");
    }
}
```

[Grafiken: The Simpsons TM, erstellt von CJ]

Stellen Sie sich das *Alphabet* so vor: Eine durchnummerierte Liste...

Großbuchstaben:

Index	0	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Eingabe	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Kleinbuchstaben:

Index	0	1	2	3	4	5	6	7	8	9	0	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Eingabe	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Beispiel 1: `(c >= 'a' && c <= 'm') c+=13`

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Eingabe	a	b	c	d	e	f	g	h	i	j	k	l	m
Ausgabe	n	o	p	q	r	s	t	u	v	w	x	y	z

Man verschiebt im ersten Fall (IF-Zweig) einfach die Ausgabe um 13 Stellen im Alphabet nach rechts (`c+=13`).

Beispiel 2: `(c >= 'n' && c <= 'z') c-=13`

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Eingabe	n	o	p	q	r	s	t	u	v	w	x	y	z
Ausgabe	a	b	c	d	e	f	g	h	i	j	k	l	m

Man verschiebt im zweiten Fall (erstes ELSEIF) einfach die Ausgabe um 13 Stellen im Alphabet nach links (`c-=13`).

Für Großbuchstaben gilt das gleiche!

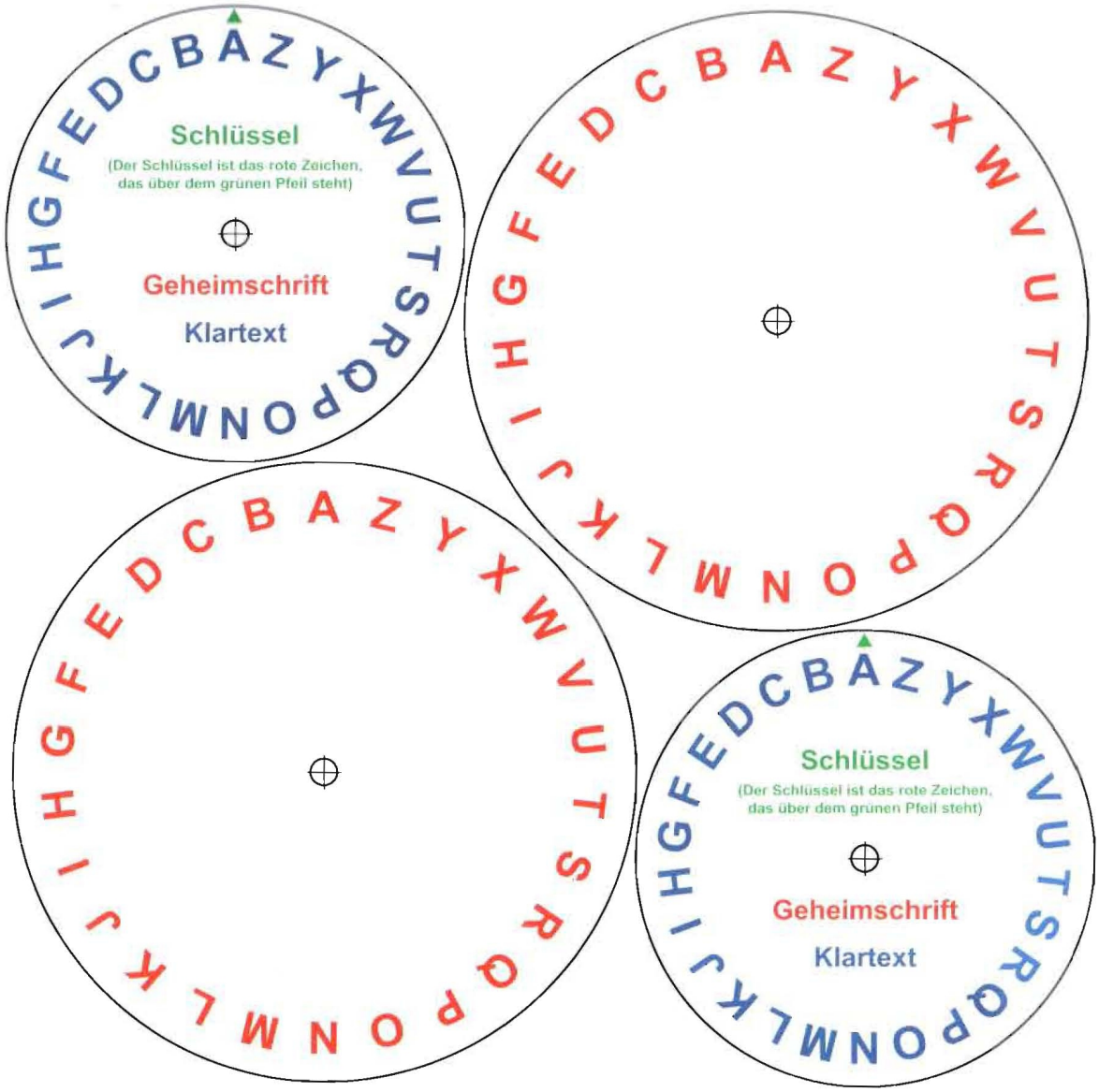
So könnte die grafische Benutzeroberfläche aussehen:

[Grafiken: The Simpsons TM, erstellt von CJ]

Vorlage: Basteln Sie zum besseren Verständnis die Cäsar-Scheibe

Testen Sie die Scheibe an verschiedenen Beispielen, sodass Sie ihren Kollegen bei der Projektpräsentation problemlos zeigen können, wie der ROT13-Algorithmus (Cäsar-Chiffre) funktioniert.

Im allerbesten Fall basteln Sie jedem Kollegen eine eigene Scheibe ;)!



[Quelle: Gallenbacher, Jens, „Abenteuer Informatik“, Spektrum Verlag, 2008]

[Grafiken: The Simpsons TM, erstellt von CJ]