

//

Programmgeneratorsystem LOGO.

Der folgende Text ist ein Programm LOGO Zufallsgenerator. Der entstandene Programmierungstext wird in der Zone des Textplakatklebers angeschlagen, der im Arbeitsfenster angesiedelt ist. Um diesen Text wiederzugewinnen muß man klicken dreimal auf dem Plakatkleber und den Text LOGO auswählen, der in produziert wurde es, der mit dem Textmarker mit der Maus streicht an, dann es, im Briefbeschwerer zu kopieren. Man muß noch nur mehr das erzeugende Programm löschen und den Text im Textverarbeitung kleben. Dieser Text illustriert ebenfalls eine Kombination nützlicher Charaktere für den Anschlag der eckige Klammer und der Anführungszeichen in einer Liste. So:

^^ als ein öffnender eckige Klammer [angeschlagen wird.
"" als ein schließender eckige Klammer] angeschlagen wird.
|| (Alt Gr gerades + 6), als Anführungszeichen " angeschlagen wird.

//

löschttext

```
setze "Befehllinien [ setze vorwärts Stiftab fi startezufall Stifthoch op Mitte rückwärts wiederhole re prüfe li ]
setze "ZahlFunktionen [ Kap zufallzahl pi Mausx Mausy xco yco ]
setze "ListeFunktionen [ Grenzenwert? Ort lage ]
setze "BoolischeFunktionen [ falsch aufbildschirm? sichtbar? wahr ]
setze "Farben [ weiss grün gelb rot orange linde pink sarcelle braun rosa violett geld grau schwarz blau olivefarbe ]
setze "Programm [ Bild löschttext beschleunige freimacht Stiftab ]
setze "FarbenSachenListe []
setze "ZählenSachenListe []
setze "eckige_Klammer 0
setze "alte_befehlen [ Stiftab]
```

```
lerne Befehl_hinzugeben :c
prüfe (Länge :alte_befehlen) = 3
wennwahr setze "alte_befehlen oe :alte_befehlen
setze "alte_befehlen Satz :alte_befehlen :c
Ende
```

```
lerne Name_Sache
startezufall 1 5 setze "l zufallzahl
startezufall 1 26 setze "n "
wiederhole :l [ setze "n Wort :n Zeichen (96 + zufallzahl) ]
rückgabe Wort "|| :n // || ist " //
Ende
```

```
lerne_Befehl
setze "c Element :Befehllinien zufallzahl
solange Element? :c :alte_befehlen setze "c Element :Befehllinien zufallzahl
wenn :c <> "op Befehl_hinzugeben :c
rückgabe :c
Ende
```

```
lerne Farbe
startezufall 1 16
rückgabe Element :Farben zufallzahl
Ende
```

```
lerne Schleife_Index
startezufall 2 360
rückgabe zufallzahl
Ende
```

```
lerne_Länge
startezufall 1 15
rückgabe zufallzahl
Ende
```

```
lerne Wert
startezufall -1000 1000
rückgabe zufallzahl
Ende
```

```
lerne ZahlSachsuche
startezufall 1 Länge :ZählenSachenListe
rückgabe Element :ZählenSachenListe zufallzahl
Ende
```

```
lerne FarbeSachsuche
startezufall 1 Länge :FarbenSachenListe
rückgabe Element :FarbenSachenListe zufallzahl
Ende
```

```
lerne ZahlFunktionensuche
startezufall 1 7
rückgabe Element :ListeFunktionen zufallzahl
Ende
```

```
lerne BoolischeFunktionensuche
startezufall 1 4
rückgabe Element :BoolischeFunktionen zufallzahl
Ende
```

```
lerne etwaszuweisung
setze "Name Wort" || "a
solange ( Element? :Name :ZählenSachenListe ) oder
( Element? :Name :FarbenSachenListe ) setze "Name Name_Sache

prüfe (Länge :ZählenSachenListe ) <= ( Länge :FarbenSachenListe )
wennwahr [
    setze "Programm Satz :Programm Satz Satz "setze :Name Wert
    setze "ZählenSachenListe Satz :ZählenSachenListe :Name
]
wennfalsch [
    setze "Programm Satz :Programm Satz Satz "setze :Name Farbe
    setze "FarbenSachenListe Satz :FarbenSachenListe :Name
]
Ende
```

```
lerne etwaszuweisen&reichnen
setze "Programm Satz :Programm Satz "setze" ZahlSachesuche
startezufall 7 9 setze "Fall zufallzahl
wenn (:Fall = 8 ) und ((Länge :ZählenSachenListe)=0)
    solange (:Fall = 8) setze "Fall zufallzahl
wenn :Fall = 7 setze "Programm Satz Satz Satz :Programm "(" Wert ")"
wenn :Fall = 8 setze "Programm Satz :Programm Satz ":" oe oe ZahlSachesuche
wenn :Fall = 9 setze "Programm Satz :Programm ZahlFunktionensuche
    startezufall 1 4
    setze "Fall zufallzahl
    wenn :Fall = 1 setze "Programm Satz :Programm "+"
    wenn :Fall = 2 setze "Programm Satz :Programm "-"
    wenn :Fall = 3 setze "Programm Satz :Programm ""
    wenn :Fall = 4 setze "Programm Satz :Programm "/"
startezufall 5 9
setze "Fall zufallzahl
wenn (:Fall = 8) und ((Länge :ZählenSachenListe)= 0) solange (:Fall = 8) setze "Fall zufallzahl
```

```
wenn :Fall = 5 [
    startezufall 0 1 setze "Komplex zufallzahl
    wenn :Komplex = 1 [ // Komplexer berechenbarer Ausdruck.//
        setze "Programm Satz :Programm "("
        setze "Fall zufallzahl
        prüfe :Fall = 1
        wennwahr setze "Programm Satz Satz Satz :Programm "(" Wert ")"
        wennfalsch setze "Programm Satz :Programm Satz ":" oe oe ZahlSachesuche
        startezufall 1 4
        setze "Fall zufallzahl
        wenn :Fall = 1 setze "Programm Satz :Programm "+"
        wenn :Fall = 2 setze "Programm Satz :Programm "-"
        wenn :Fall = 3 setze "Programm Satz :Programm ""
        wenn :Fall = 4 setze "Programm Satz :Programm "/"
    ]
    setze "Programm Satz :Programm "cos"
    startezufall 0 1
    setze "Fall zufallzahl
    prüfe :Fall = 0
    wennwahr setze "Programm Satz Satz Satz :Programm "(" Wert ")"
    wennfalsch setze "Programm Satz :Programm Satz ":" oe oe ZahlSachesuche
    wenn :Komplex = 1 setze "Programm Satz :Programm ")"
]
wenn :Fall = 6 [
    startezufall 0 1 setze "Komplex zufallzahl
    wenn :Komplex = 1 [ // Komplexer berechenbarer Ausdruck. //
        setze "Programm Satz :Programm "("
        setze "Fall zufallzahl
        prüfe :Fall = 1
        wennwahr setze "Programm Satz Satz Satz :Programm "(" Wert ")"
        wennfalsch setze "Programm Satz :Programm Satz ":" oe oe ZahlSachesuche
        startezufall 1 4
```

```

setze "Fall zufallzahl
wenn :Fall = 1 setze "Programm Satz :Programm "+"
wenn :Fall = 2 setze "Programm Satz :Programm "-
wenn :Fall = 3 setze "Programm Satz :Programm ""
wenn :Fall = 4 setze "Programm Satz :Programm "/"
]
setze "Programm Satz :Programm "sin"
startezufall 0 1
setze "Fall zufallzahl
prüfe :Fall = 0
wennwahr setze "Programm Satz Satz Satz :Programm "(" Wert ")"
wennfalsch setze "Programm Satz :Programm Satz ":" oe oe ZahlSachsuche
wenn :Komplex = 1 setze "Programm Satz :Programm ")"
]
wenn :Fall = 7 setze "Programm Satz Satz Satz :Programm "(" Wert ")"
wenn :Fall = 8 setze "Programm Satz :Programm Satz ":" oe oe ZahlSachsuche
wenn :Fall = 9 setze "Programm Satz :Programm ZahlFunktionensuche
Ende

startezufall 1 10
wiederhole 5 etwaszuweisung

lerne Aufbau
startezufall 2 13
setze "Befehl_Befehl
wenn :Befehl = "setze [ etwaszuweisung Befehl_hinzugeben "setze ]

wenn :Befehl = "fi [
    wenn (Element? (:alte_befehlen) "fi) [
        startezufall 0 1 Befehl_hinzugeben "fi
        setze "Fall zufallzahl
        wenn (:Fall = 1) und ((Länge :FarbenSachenListe) = 0) setze "Fall 0
        prüfe :Fall = 0 // Ist das Argument ein Farbe (unmittelbarer Wert)?//
        wennwahr [ setze "Programm Satz :Programm Satz "fi Farbe ]
        wennfalsch [ // Das Argument ist ein Sache. //
            setze "Programm Satz :Programm Satz "fi Satz ":" oe oe FarbeSachsuche
        ]
    ]
]

wenn :Befehl = "wiederhole [
    startezufall 0 2
    setze "Fall zufallzahl Befehl_hinzugeben "
    wenn (:Fall = 1) und ((Länge :ZählenSachenListe) = 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert? //
    wennwahr setze "Programm Satz :Programm Satz Satz "wiederhole abs Wert ""^ [ //
    wennfalsch [
        prüfe :Fall = 1
        wennwahr setze "Programm Satz :Programm Satz Satz "wiederhole Satz ":" oe oe ZahlSachsuche ""^
        wennfalsch setze "Programm Satz :Programm [ wiederhole zufallzahl ]
    ]
    setze "eckige_Klammer :eckige_Klammer + 1
    startezufall 1 3 setze "Fall zufallzahl
    wiederhole :Fall Aufbau
    prüfe (lz :Programm) = ""^
    wennwahr setze "Programm ol ol ol :Programm
    wennfalsch setze "Programm Satz :Programm ""^
    setze "eckige_Klammer :eckige_Klammer - 1
    Befehl_hinzugeben "
]

wenn :Befehl = "prüfe [
    startezufall 0 4
    setze "Fall zufallzahl
    wenn (:Fall = 1) und ((Länge :ZählenSachenListe) = 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert?//
    wennwahr setze "Programm Satz :Programm Satz "prüfe Wert
    wennfalsch [
        prüfe :Fall = 1
        wennwahr setze "Programm Satz :Programm Satz "prüfe Satz ":" oe oe ZahlSachsuche
        wennfalsch [
            prüfe :Fall = 2
            wennwahr setze "Programm Satz :Programm Satz "prüfe "zufallzahl
            wennfalsch [
                prüfe :Fall = 3
                wennwahr setze "Programm Satz :Programm Satz "prüfe ZahlFunktionensuche
                wennfalsch setze "Programm Satz :Programm Satz "prüfe BoolischeFunktionensuche
            ]
        ]
    ]
]

```

```
]
]
]
```

startezufall 1 6

```
setze "Fall zufallzahl
wenn :Fall = 1 setze "Programm Satz :Programm "<
wenn :Fall = 2 setze "Programm Satz :Programm ">
wenn :Fall = 3 setze "Programm Satz :Programm "<=
wenn :Fall = 4 setze "Programm Satz :Programm ">=
wenn :Fall = 5 setze "Programm Satz :Programm "="
wenn :Fall = 6 setze "Programm Satz :Programm "<>
```

startezufall 0 4

```
setze "Fall zufallzahl Befehl_hinzugeben "prüfe
wenn ( :Fall = 1 ) und ((Länge :ZählenSachenListe)= 0) setze "Fall 0
prüfe :Fall = 0 //Ist das Argument ein unmittelbarer Wert?//
wennwahr setze "Programm Satz :Programm Wert
wennfalsch [
  prüfe :Fall = 1
  wennwahr setze "Programm Satz :Programm Satz ": oe oe ZahlSachsuche
  wennfalsch [
    prüfe :Fall = 2
    wennwahr setze "Programm Satz :Programm zufallzahl
    wennfalsch [
      prüfe :Fall = 3
      wennwahr setze "Programm Satz :Programm ZahlFunktionnsuche
      wennfalsch setze "Programm Satz :Programm BoolischeFunktionnsuche
    ]
  ]
]
```

```
setze "Programm Satz :Programm Satz "wennwahr" ""^
startezufall 1 3 setze "h zufallzahl
wiederhole :h Aufbau
prüfe (lz :Programm) = ""^
wennwahr setze "Programm ol ol :Programm
wennfalsch setze "Programm Satz :Programm ""
Befehl_hinzugeben "prüfe
setze "Programm Satz :Programm Satz "wennfalsch" ""^
startezufall 1 3 setze "h zufallzahl
wiederhole :h Aufbau
prüfe (lz :Programm) = ""^
wennwahr setze "Programm ol ol :Programm
wennfalsch setze "Programm Satz :Programm ""
Befehl_hinzugeben "prüfe
]
```

```
wenn :Befehl = "vorwärts" [
  wenn nicht (Element? :alte_befehlen "vorwärts" ) [
    startezufall 0 2
    setze "Fall zufallzahl Befehl_hinzugeben "vorwärts
    wenn ( :Fall = 1 ) und ( (Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert?//
    wennwahr setze "Programm Satz :Programm Satz "vorwärts_Länge
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz "vorwärts Satz ": oe oe ZahlSachsuche
      wennfalsch setze "Programm Satz :Programm Satz "vorwärts "zufallzahl
    ]
  ]
]
```

```
wenn :Befehl = "rückwärts" [
  wenn nicht (Element? :alte_befehlen "rückwärts" ) [
    startezufall 0 2
    setze "Fall zufallzahl Befehl_hinzugeben "rückwärts
    wenn ( :Fall = 1 ) und ( (Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert? //
    wennwahr setze "Programm Satz :Programm Satz "rückwärts_Länge
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz "rückwärts Satz ": oe oe ZahlSachsuche
      wennfalsch setze "Programm Satz :Programm Satz "rückwärts "zufallzahl
    ]
  ]
]
```

```

wenn :Befehl = "li [
  wenn nicht (Element? :alte_befehlen "li ) [
    startezufall 0 3
    setze "Fall zufallzahl Befehl_hinzugeben "li
    wenn ( :Fall = 1 ) und ((Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert?//
    wennwahr setze "Programm Satz :Programm Satz "li _Länge
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz "li Satz ": oe oe ZahlSachesuche
      wennfalsch [
        prüfe :Fall = 2
        wennwahr setze "Programm Satz :Programm Satz "li "zufallzahl
        wennfalsch setze "Programm Satz :Programm Satz "li "Kompass
      ]
    ]
  ]
]

wenn :Befehl = "re [
  wenn nicht (Element? :alte_befehlen "re ) [
    startezufall 0 3
    setze "Fall zufallzahl Befehl_hinzugeben "re
    wenn ( :Fall = 1 ) und ( (Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert?//
    wennwahr setze "Programm Satz :Programm Satz "re _Länge
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz "re Satz ": oe oe ZahlSachesuche
      wennfalsch [
        prüfe :Fall = 2
        wennwahr setze "Programm Satz :Programm Satz "re "zufallzahl
        wennfalsch setze "Programm Satz :Programm Satz "re "Kompass
      ]
    ]
  ]
]

wenn :Befehl = "startezufall [
  wenn nicht (Element? :alte_befehlen "startezufall ) [
    Befehl_hinzugeben "startezufall
    startezufall 0 2
    setze "Fall zufallzahl
    wenn ( :Fall = 1 ) und ((Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 //Ist das Argument ein unmittelbarer Wert?//
    wennwahr setze "Programm Satz :Programm Satz "startezufall Schleife_Index
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz "startezufall Satz ": oe oe ZahlSachesuche
      wennfalsch setze "Programm Satz :Programm ZahlFunktionensuche
    ]
    startezufall 0 2
    setze "Fall zufallzahl
    wenn ( :Fall = 1 ) und ( (Länge :ZählenSachenListe)= 0) setze "Fall 0
    prüfe :Fall = 0 // Ist das Argument ein unmittelbarer Wert? //
    wennwahr setze "Programm Satz :Programm Schleife_Index
    wennfalsch [
      prüfe :Fall = 1
      wennwahr setze "Programm Satz :Programm Satz ": oe oe ZahlSachesuche
      wennfalsch setze "Programm Satz :Programm ZahlFunktionensuche
    ]
  ]
]

wenn :Befehl = "op etwaszuweisen&reichnen
wenn ( (:Befehl = "Stifthoch) oder
  (:Befehl = "Stiftab) oder
  (:Befehl = "Mitte)) [
  wenn ( nicht (Element? :alte_befehlen "Stifthoch ) und
    nicht ( Element? :alte_befehlen "Stiftab) und
    nicht ( Element? :alte_befehlen "Mitte)) [
    setze "Programm Satz :Programm :Befehl
    Befehl_hinzugeben :Befehl
  ]
]

rückkehr
Ende

```

wiederhole 5 Aufbau

schreiblinie [Um das durch diesen Algorithmus geschaffene Programm LOGO, zurückzukommen über das Arbeitfenster und klicken dreimal über den Textplakatkleber zu setzen, um seinen Inhalt im Briefbeschwerer (mit dem Textmarker anstreichen) zu kopieren. Bleibt, danach diesen Inhalt im zuvor geleerten Textverarbeitung Programmgeneratorsystem LOGO zu kleben.]

schreiblinie "schreiblinie"
schreiblinie :Programm