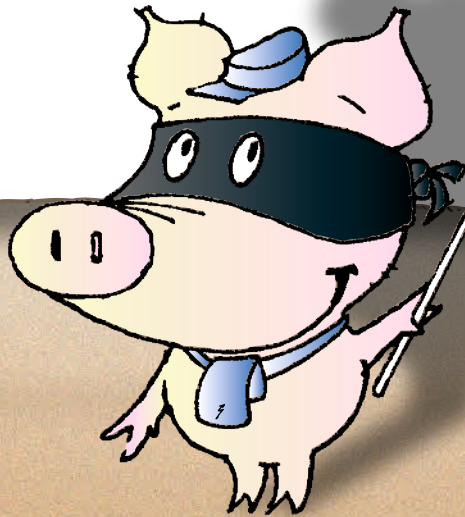
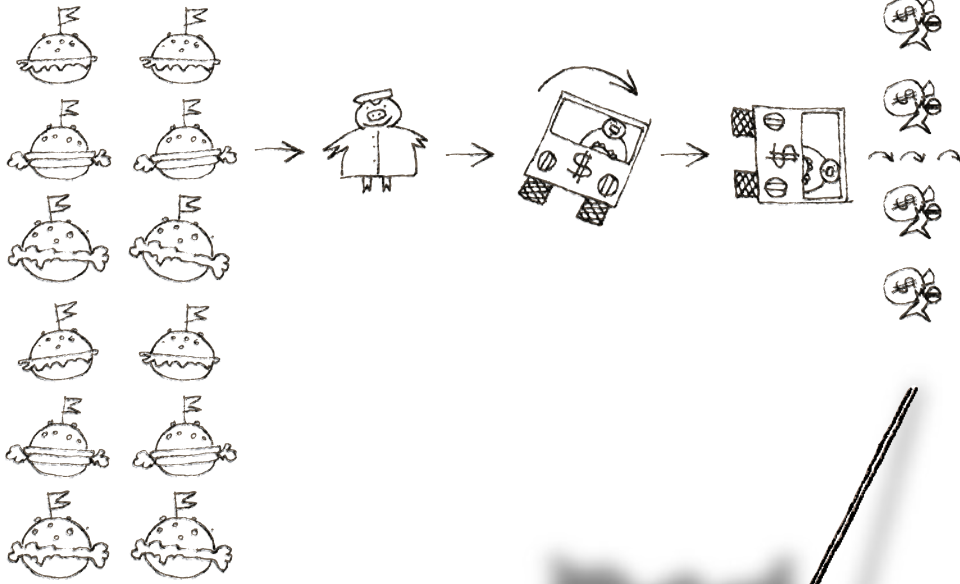
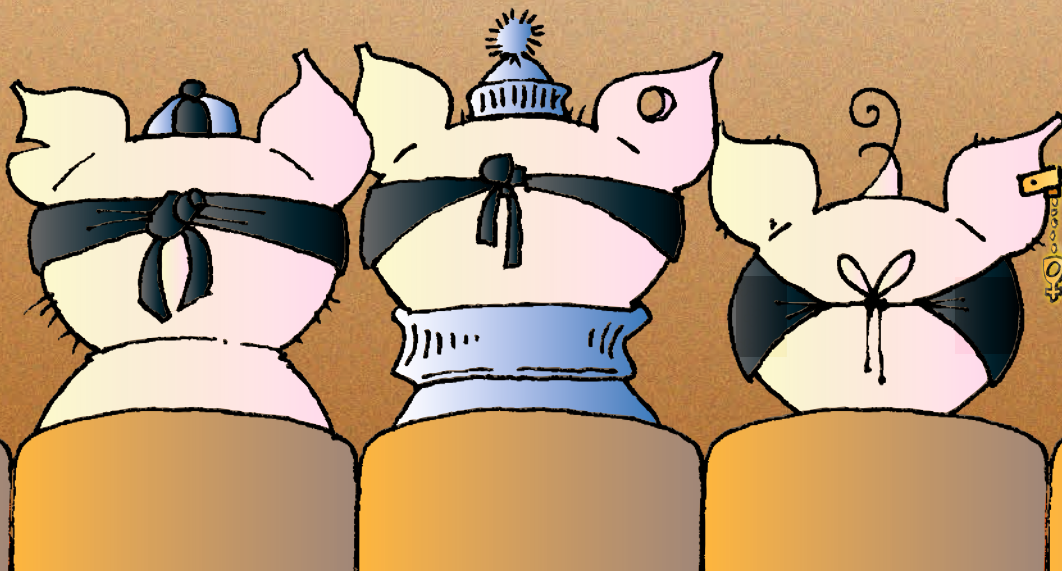


1.            2.            3.            4.



# 31. BUNDESWETTBEWERB INFORMATIK





Liebe Teilnehmerinnen und Teilnehmer am Bundeswettbewerb Informatik,

Informatik ist allgegenwärtig. Sie prägt unseren Alltag in nahezu jeder Hinsicht: Gleich, ob wir telefonieren, Auto fahren, die Mikrowelle oder die Spülmaschine bedienen – ständig hilft sie uns, denn in all diesen Geräten sorgen informationstechnische Systeme für eine intelligente Steuerung.

Die Beschäftigung mit der faszinierenden Welt der Informatik bietet Schülerinnen und Schülern mehrere Chancen: Sie können sich schon früh mit technologischen Zusammenhängen vertraut machen, und sie können eigene Talente entdecken. Eine berufliche Tätigkeit in der Informatik erschöpft sich nicht in der Entwicklung von Computerprogrammen. Sie erlaubt vielmehr einen aktiven Beitrag zum technologischen und wirtschaftlichen Fortschritt.

Der Bundeswettbewerb Informatik und der Jugendwettbewerb „Informatik-Biber“ geben Schülerinnen und Schülern solche wichtigen Einblicke in die Welt der Bits und Bytes. Sie wecken bei jungen Menschen Neugier, Interesse und Begeisterung für das Fach Informatik und tragen außerdem dazu bei, die Bedeutung dieses Faches einer breiteren Öffentlichkeit zu vermitteln.

Über 150.000 Jugendliche haben im vergangenen Jahr durch ihre Teilnahme am Informatik-Biber bewiesen, dass sie keinerlei Berührungsängste mit der Informatik haben. Jene, die durch den Informatik-Biber Lust auf mehr bekommen haben, können ihre Interessen im Rahmen des Bundeswettbewerbs Informatik vertiefen. Beim Lösen der Aufgaben wünsche ich allen Teilnehmerinnen und Teilnehmern gute Ideen und viel Erfolg.

Prof. Dr. Annette Schavan  
Bundesministerin für Bildung und Forschung

## Gesellschaft für Informatik e.V. (GI)

Die Gesellschaft für Informatik e.V. (GI) ist mit rund 22.000 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](http://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 4000 Mitarbeiter in bundesweit 19 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](http://www.innovisions.de). Weitere Informationen über den Fraunhofer IuK-Verbund gibt es auf [www.iuk.fraunhofer.de](http://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](http://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 erfährt der Bundeswettbewerb Informatik und damit die Initiative BWINF (Bundesweit Informatiknachwuchs fördern) 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 31. 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. Er gehört zu den bundesweiten Schülerwettbewerben, die von den Kultusministerien der Länder empfohlen werden. Gefördert wird er vom Bundesministerium für Bildung und Forschung und steht unter der Schirmherrschaft des Bundespräsidenten. Der BwInf ist Kern von „Bundesweit Informatiknachwuchs fördern“ (BWINF), einer gemeinsamen Initiative von GI, Fraunhofer-Verbund IuK-Technologie und Max-Planck-Institut für Informatik. Die Gestaltung des Wettbewerbs und die Auswahl der Sieger obliegen dem Beirat; Vorsitzende: Prof. Dr. Nicole Schweikard, Universität Frankfurt. Die Auswahl und Entwicklung von Aufgaben und die Festlegung von Bewertungsverfahren übernimmt der Aufgabenausschuss; Vorsitzender: Prof. Dr. Peter Rossmann, RWTH Aachen. Die Geschäftsstelle des Wettbewerbs 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 analysiert und bearbeitet im Team zwei Informatik-Probleme.

## Juniorliga

Um die Teilnahme jüngerer Schülerinnen und Schüler am BwInf zu fördern, gibt es in der ersten Runde die Juniorliga. Für Schüler, die nicht älter als 16 Jahre sind, werden zwei leichtere Aufgaben gestellt, die Junioraufgaben. Stammt eine Einsendung von Schülern, die alle die Altersgrenze für Junioraufgaben erfüllen, und sind darin nur Junioraufgaben bearbeitet, nimmt sie in der Juniorliga teil. Die Juniorliga wird getrennt bewertet, Preise werden separat vergeben.

# 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 2013 in Kaiserslautern ausgerichtet wird, vom Fraunhofer IESE und der TU Kaiserslautern.

Die dort ermittelten Bundessieger werden in der Regel ohne weiteres Aufnahmeverfahren in die Studienstiftung des deutschen Volkes aufgenommen. Zusätzlich sind für den Bundessieg, 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 2014 in Taiwan teilnimmt.

## Informatik-Workshops etc.

Informatik-Workshops exklusiv für TeilnehmerInnen werden in Baden-Württemberg, vom Hasso-Plattner-Institut, von Hochschulen wie der RWTH Aachen oder der TU Dortmund und dem 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“ und einige der Allerbesten zu einem Treffen mit Informatik-Talenten aus ganz Europa ein.

Ausgewählte Endrundenteilnehmer werden im Herbst 2013 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 – wobei eine vollwertige Einsendung durch zwei Einsendungen in der Juniorliga ersetzt werden kann – müssen mindestens 10 Schülerinnen und Schüler einer Schule, darunter bei gemischten Schulen mindestens 2 Jungen und mindestens 2 Mädchen, beteiligt sein. Schulen, die diese Bedingung erfüllen, werden als „BwInf-Schule 2012/2013“ 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 3.12.1990 geboren wurden. Sie dürfen jedoch zum 1.9.2012 noch nicht ihre (informatikbezogene) Ausbildung abgeschlossen oder eine Berufstätigkeit begonnen haben. Personen, die im Wintersemester 2012/13 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 3.12.2012 ihren Wohnsitz in Deutschland haben oder eine staatlich anerkannte deutsche Schule im Ausland besuchen.

Junioraufgaben dürfen von bis zu 16-Jährigen bearbeitet werden (geboren nach dem 3.12.1995) bzw. von Gruppen mit mindestens einem solchen Mitglied.

## Weiterkommen

An der zweiten Runde dürfen jene teilnehmen, die allein oder mit ihrer Gruppe 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 noch keine zweite Runde.

## Einsendungen

... enthalten Bearbeitungen zu mindestens einer Aufgabe und werden von Einzelpersonen oder Gruppen abgegeben. Eine Einsendung besteht aus einem schriftlichen und einem elektronischen Teil. Zu jeder bearbeiteten Aufgabe enthält der schriftliche Teil (Dokumentation) eine Beschreibung der Lösungsidee und Beispiele, die die Korrektheit der Lösung belegen. Ist ein Programm gefordert, sollen außerdem die Umsetzung der Lösungsidee in das Programm erläutern und die wichtigsten Teile des Quelltextes hinzugefügt werden. Die Dokumentation muss ausgedruckt abgegeben werden. Achtung: eine gute Dokumentation muss nicht lang sein! Der elektronische Teil enthält für Aufgaben, in denen ein Programm gefordert ist, das lauffähige Programm selbst und den kompletten Quelltext des Programms. Er muss, zusammen mit einer elektronischen Fassung der Dokumentation, auf CD/DVD abgegeben werden.

## Anmeldung

Die Anmeldung ist bis zum Einsendeschluss möglich, und zwar online unter

[www.bundeswettbewerb-informatik.de/anmeldung](http://www.bundeswettbewerb-informatik.de/anmeldung)

In diesem Jahr ist zum ersten Mal eine **Lehreranmeldung** möglich. Bitte erkundigen Sie sich unter [bwinf@bwinf.de](mailto:bwinf@bwinf.de) nach den Modalitäten.

## Einsendung an:

BWINF, Wachsbleiche 7, 53111 Bonn

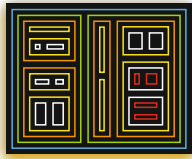
## Einsendeschluss:

3.12.2012 (Datum des Poststempels)

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: Rechteckschoner

Als eine ältere Mitschülerin von den letzten Zweitrundenaufgaben erzählte, hatte Moni die Idee zu einem eigenen Bildschirmschoner: Gezeichnet werden soll ein Rechteck, das zwei Rechtecke enthält, die jeweils wieder zwei Rechtecke enthalten usw. Damit das nett aussieht, soll bei der Rechteckschachtelung der Zufall eine sinnvolle Rolle spielen, und ein enthaltenes Rechteck soll sich nicht nur in der Größe von dem es unmittelbar enthaltenden Rechteck unterscheiden – etwa so:



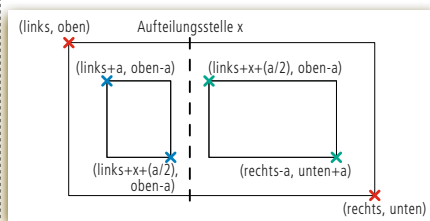
Nach dem Aufbau soll das fertige Rechteckschachtelbild immer wieder so aktualisiert werden, dass mal das eine, mal das andere Rechteck (evtl. mit allen enthaltenen Rechtecken) sein Aussehen verändert.

## Aufgabe

Entwickle einen solchen „Rechteckschoner“. Demonstriere seine Funktion an mindestens drei verschiedenen Rechteckschachtelbildern mit jeweils zwei Veränderungen.

## Lösungsidee

Um ein einzelnes Rechteck zeichnen zu können, müssen seine Position und seine Farbe festgelegt werden. Die Farbe wird zufällig aus einer festen Liste gewählt. Die Position eines Rechtecks wird durch zwei diagonal gegenüberliegende Eckpunkte festgelegt. Beim äußersten Rechteck werden die Eckpunkte so bestimmt, dass sie alle den gleichen Abstand zu den Bildschirmrändern haben. Um zwei Rechtecke in das äußere zu zeichnen, wird dessen Fläche zufällig aufgeteilt. Die Eckpunkte der inneren Rechtecke werden so bestimmt, dass sie den gleichen Abstand zueinander haben wie zum äußeren Rechteck. Die folgende Abbildung zeigt, wie die Eckpunkte der inneren Rechtecke aus den Eckpunkten des äußeren Rechtecks (links, oben) und (rechts, unten) sowie der „Aufteilungsstelle“  $x$  berechnet werden. Das äußere Rechteck wird senkrecht aufgeteilt, wenn es breiter ist als hoch (wie in der Abbildung); andernfalls wird es waagrecht aufgeteilt, und die Eckpunkte der inneren Rechtecke werden entsprechend berechnet.



Damit das Rechteckbild gleichmäßig gefüllt wird, wird eine Liste der noch zu zeichnenden Rechtecke geführt. Diese Liste wird nach und nach abgearbeitet. Beim Füllen eines Rechtecks werden die beiden inneren Rechtecke hinten an die Liste angehängt. So wird erreicht, dass die Rechtecke Stufe für Stufe gezeichnet werden: erst das äußere Rechteck, dann dessen zwei innere Rechtecke, dann deren vier innere Rechtecke usw. Da ein Rechteck nur dann gefüllt wird, wenn es noch genügend Platz für zwei innere Rechtecke hat, endet dieser Prozess, wenn alle zu zeichnenden Rechtecke die Mindestgröße unterschreiten.

Alle gezeichneten Rechtecke werden in einer Liste abgespeichert. Ist das Bild einmal fertig, wird aus dieser Liste zufällig ein Rechteck ausgewählt. Es wird dann mit seinem Inhalt zunächst komplett gelöscht und übermalt und dann mit neuen inneren Rechtecken neu gezeichnet – auf die gleiche Art und Weise wie ursprünglich das äußerste Rechteck.

## Umsetzung

Die Lösungsidee wird in der Sprache Python umgesetzt, und zwar mit Hilfe des Moduls „turtle“, das viele Funktionen zum Zeichnen von Figuren bereitstellt.

Alle Eigenschaften und wichtigen Funktionen eines Rechtecks werden in einer Klasse **Rechteck** zusammengefasst. Eigenschaften sind im Wesentlichen die Koordinaten der Eckpunkte (**links**, **oben**, **rechts** und **unten**) und die Farbe **farbe**. Als Funktionen (oder besser: Methoden) eines Rechtecks werden definiert:

**zeichne** : nutzt Funktionen des Moduls turtle, um anhand der Eckpunkte ein Rechteck der gesetzten Farbe zu zeichnen.

**fuelle** : bestimmt, ob das Rechteck groß genug ist, um zwei innere Rechtecke zu erhalten. Wenn ja, wird das Rechteck aufgeteilt und es werden zwei innere Rechtecke erzeugt.

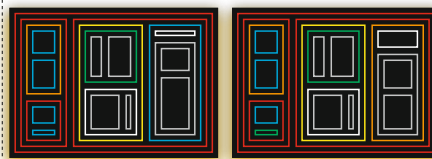
**loesche** : entfernt das Rechteck aus der Liste aller Rechtecke und löscht rekursiv seine Inhaltsrechtecke; jedes gelöschte Rechteck wird abschließend in Hintergrundfarbe neu gezeichnet und damit auch aus der Anzeige gelöscht.

Alle Rechtecke (also alle Objekte der Klasse **Rechteck**) werden in der globalen Liste **alle\_rechtecke** gespeichert.

Die zentrale Funktion zum Zeichnen eines Rechtecks ist **zeichne\_rechtecke**; sie benutzt die lokale Liste **aktuelle\_rechtecke**, um – wie in der Lösungsidee beschrieben – die Rechtecke stufenweise zu erzeugen und zu zeichnen.

## Beispiel

Aus Platzgründen zeigen wir nur ein Rechteckschachtelbild mit einer Veränderung. Die Bilder sind 400 mal 300 Pixel groß. Die Veränderungen vom linken zum rechten Bild sind: Das innerste Rechteck links unten wurde gelöscht und in anderer Farbe neu gezeichnet; außerdem wurde im rechten Drittel des Bildes das blaue Rechteck gelöscht und mit neuen inneren Rechtecken neu gezeichnet.



## Quelltext

```
# Modulimport
import turtle
import random
```

```
# Hier kann die Größe des Fensters an den Bildschirm angepasst werden.
SCREEN_WIDTH = 400
SCREEN_HEIGHT = 300
```

```
SPEED = 5 # Zeichengeschwindigkeit
```

```
# Farben für die Rechtecke
COLORS = ["red", "white", "yellow", "green", "blue",
           "orange", "brown", "pink", "gray", "violet"]
BGCOLOR = "black" # Hintergrundfarbe
```

```
# Rechteck-Klasse mit Eigenschaften und Operationen
class Rechteck:
```

```
    def __init__(self, links, oben, rechts, unten):
        """Die Eigenschaften des Rechtecks werden initialisiert."""
        self.links=links
        self.oben=oben
        self.rechts=rechts
        self.unten=unten
        self.hoehe=oben-unten # nicht nötig, aber nützlich
        self.breite=rechts-links # nicht nötig, aber nützlich
        self.farbe=self.waehle_farbe()
        self.inhalt=[] # Liste für die beiden direkt enthaltenen Rechtecke
```

```
    def waehle_farbe(self):
        """Die Linienfarbe wird zufällig ausgewählt."""
        random.shuffle(COLORS)
        stift.color(COLORS[0])
```

```
    def zeichne(self):
        """Das Rechteck wird mit Turtle-Kommandos gezeichnet."""
        stift.up()
        stift.goto(self.links, self.oben)
        stift.down()
        stift.fd(self.breite)
        stift.right(90)
        stift.fd(self.hoehe)
        stift.right(90)
        stift.fd(self.breite)
        stift.right(90)
        stift.fd(self.hoehe)
        stift.right(90)
```

```
def fuelle(self):
```

```
    """Das Rechteck bekommt zwei Rechtecke als Inhalt, wenn es groß
    genug ist. Wenn es breiter als hoch ist, werden die enthaltenen
    Rechtecke nebeneinander, sonst übereinander platziert."""
    if self.breite >= 60 and self.hoehe >= 60:
        if self.breite > self.hoehe:
            x = random.uniform(self.links+30, self.rechts-30)
            r1 = Rechteck(self.links+10, self.oben-10, x-5, self.unten+10)
            r2 = Rechteck(x+5, self.oben-10, self.rechts-10, self.unten+10)
            self.inhalt = [r1, r2]
        else:
            x = random.uniform(self.oben-30, self.unten+30)
            r1 = Rechteck(self.links+10, self.oben-10, self.rechts-10, x+5)
            r2 = Rechteck(self.links+10, x-5, self.rechts-10, self.unten+10)
            self.inhalt = [r1, r2]
    return self.inhalt
```

```
def loesche(self):
    """Das Rechteck wird mit seinem ganzen Inhalt, also rekursiv,
    aus der Liste mit allen Rechtecken gelöscht. Die Rechtecke werden
    von innen nach außen mit der Hintergrundfarbe übermalt."""
    global alle_rechtecke
    alle_rechtecke.remove(self)
    for irechteck in self.inhalt:
        irechteck.loesche()
    stift.color(BGCOLOR)
    self.zeichne()
```

```
def schluss(x,y):
    """Das Turtle-Fenster wird geschlossen."""
    global screen
    screen.bye()
```

```
def setup_turtle():
    """Turtle-(Stift) und Turtle-Fenster werden vorbereitet."""
    global stift, screen
    stift=turtle.Turtle()
    screen=stift.getscreen()
    screen.title("Rechteckschoner")
    screen.bgcolor(BGCOLOR)
    screen.setup(SCREEN_WIDTH, SCREEN_HEIGHT, None, None)
    screen.listen()
    screen.onclick(schluss) # Mit einem Klick die Turtle beenden.
    stift.reset()
    stift.speed(SPEED)
```

```
def zeichne_rechtecke(ausseneck):
    """Ein Rechteck wird mit seinen inneren Rechtecken gezeichnet."""
    aktuelle_rechtecke = []
    aktuelle_rechtecke.append(ausseneck)
    # Alle Rechtecke werden gezeichnet, so lange Platz da ist.
    for r0 in aktuelle_rechtecke:
        for r in r0.fuelle():
            aktuelle_rechtecke.append(r)
            r0.zeichne()
    return aktuelle_rechtecke
```

```
def main():
    global alle_rechtecke
    setup_turtle()
    alle_rechtecke = []
    # Mit dem ersten, äußersten Rechteck geht es los.
    starteck = Rechteck(-SCREEN_WIDTH/2+20, SCREEN_HEIGHT/2-20,
                       SCREEN_WIDTH/2-20, -SCREEN_HEIGHT/2+20)
    # Endlosschleife, in der Rechtecke gezeichnet und gelöscht werden.
    while 1:
        alle_rechtecke.extend(zeichne_rechtecke(starteck))
        # Ein Rechteck zum Löschen auswählen und löschen ...
        wegeck = random.choice(alle_rechtecke)
        wegeck.loesche()
        # ... und mit neuer Farbe zum neuen Starteck machen.
        wegeck.waehle_farbe()
        starteck = wegeck
    return "EVENTLOOP"
```

```
if __name__ == '__main__':
    msg = main()
    print(msg)
    mainloop()
```