

# ANHANG

# B

## Greenfoot API



Die Greenfoot API besteht aus sieben Klassen:

<b>Actor</b>	<b>Actor</b> -Methoden stehen allen Unterklassen von <b>Actor</b> zur Verfügung.	<b>MouseInfo</b>	Liefert Informationen über den Zustand der Maus und das letzte Mausereignis.
<b>World</b>	<b>World</b> -Methoden stehen der Welt zur Verfügung.	<b>GreenfootImage</b>	Wird benötigt für die Darstellung von und Arbeit mit Bildern.
<b>Greenfoot</b>	Wird verwendet, um mit der Greenfoot-Umgebung selbst zu kommunizieren.	<b>UserInfo</b>	Zur Sicherung von Benutzerdaten auf einem Server (nur, wenn auf der Greenfoot-Webseite ausgeführt)
<b>GreenfootSound</b>	Für die Sound-Wiedergabe		

Die hier gezeigte API gehört zu Greenfoot-Version 2.4.0. Wenn du eine neuere Version von Greenfoot verwendest, schaue bitte in die Online-Dokumentation.

### Methoden der Klasse World

**World**(int worldWidth, int worldHeight, int cellSize)

Erzeugt eine neue Welt.

**World**(int worldWidth, int worldHeight, int cellSize, boolean bounded)

Erzeugt eine neue Welt (möglicherweise grenzenlos).

void **act**()

Eine Methode zum Agieren in der Welt. Wird einmal für jeden **act**-Schritt aufgerufen.

void **addObject**(Actor object, int x, int y)

Fügt einen Akteur in die Welt ein.

GreenfootImage **getBackground**()

Liefert das Hintergrundbild der Welt zurück.

int **getCellSize**()

Liefert die Größe einer Zelle (in Pixel) zurück.

Color **getColorAt**(int x, int y)

Liefert die Farbe in der Mitte der Zelle zurück.

int **getHeight**()

Liefert die Höhe der Welt (als Anzahl von Zellen) zurück.

List **getObjects**(Class cls)

Ermittelt alle Objekte in der Welt.

List **getObjectsAt**(int x, int y, Class cls)

Liefert alle Objekte in einer gegebenen Zelle zurück.

int **getWidth**()

Liefert die Breite der Welt (als Anzahl von Zellen) zurück.

int **numberOfObjects**()

Ermittelt die aktuelle Anzahl der Akteure in der Welt.

void **removeObject**(Actor object)

Entfernt ein Objekt aus der Welt.

void **removeObjects**(Collection objects)

Entfernt eine Liste von Objekten aus der Welt.

## Methoden der Klasse World (Fortsetzung)

<b>void repaint()</b>	Zeichnet die Welt neu.
<b>void setActOrder(Class... classes)</b>	Legt fest, in welcher Reihenfolge die Objekte in der Welt agieren.
<b>void setBackground(GreenfootImage image)</b>	Richtet das übergebene Bild als Hintergrundbild für die Welt ein.
<b>void setBackground(String filename)</b>	Richtet das Bild aus der angegebenen Bilddatei als Hintergrundbild für die Welt ein.
<b>void setPaintOrder(Class... classes)</b>	Legt fest, in welcher Reihenfolge die Objekte in der Welt gezeichnet werden.
<b>void showText(String text, int x, int y)</b>	Zeigt Text zentriert an der angegebenen Position in der Welt an.
<b>void started()</b>	Wird vom Greenfoot-System aufgerufen, wenn die Ausführung begonnen hat.
<b>void stopped()</b>	Wird vom Greenfoot-System aufgerufen, wenn die Ausführung beendet wurde.

## Methoden der Klasse Actor

<b>Actor()</b>	Erzeugt einen Akteur.
<b>void act()</b>	Die <b>act</b> -Methode wird von dem Greenfoot Framework aufgerufen, um Objekten die Möglichkeit zu geben, eine Aktion auszuführen.
<b>protected void addToWorld(World world)</b>	Diese Methode wird von dem Greenfoot-System aufgerufen, nachdem das Objekt in die Welt eingefügt wurde.
<b>GreenfootImage getImage()</b>	Liefert das Bild zurück, das diesen Akteur repräsentiert.
<b>protected List getIntersectingObjects(Class cls)</b>	Liefert alle Objekte zurück, die sich mit diesem Objekt überschneiden.
<b>protected List getNeighbours(int distance, boolean diagonal, Class cls)</b>	Liefert die Nachbarn dieses Objekts innerhalb eines bestimmten Abstands zurück.
<b>protected List getObjectsAtOffset(int dx, int dy, Class cls)</b>	Liefert alle Objekte zurück, die sich mit einer gegebenen Position überschneiden (relativ zur Position dieses Objekts).
<b>protected List getObjectsInRange(int r, Class cls)</b>	Liefert alle Objekte im Umkreis <b>r</b> um dieses Objekt zurück.
<b>protected Actor getOneIntersectingObject(Class cls)</b>	Liefert ein Objekt zurück, das sich mit diesem Objekt überschneidet.
<b>protected Actor getOneObjectAtOffset(int dx, int dy, Class cls)</b>	Liefert ein Objekt zurück, das sich in der angegebenen Zelle befindet (relativ zur Position dieses Objekts).
<b>int getRotation()</b>	Liefert die aktuelle Drehung dieses Objekts zurück.
<b>World getWorld()</b>	Liefert die Welt zurück, in der dieses Objekt lebt.
<b>int getX()</b>	Liefert die x-Koordinate von der aktuellen Position des Objekts zurück.

## Methoden der Klasse Actor

<code>int getY()</code>	Liefert die <i>y</i> -Koordinate von der aktuellen Position des Objekts zurück.
<code>protected boolean intersects(Actor other)</code>	Prüft, ob dieses Objekt sich mit einem anderen gegebenen Objekt überschneidet.
<code>boolean isAtEdge()</code>	Stellt fest, ob der Akteur den Rand der Welt erreicht hat.
<code>protected boolean isTouching(Class cls)</code>	Prüft, ob dieser Akteur irgendein anderes Objekt der angegebenen Klasse berührt.
<code>void move(int distance)</code>	Verschiebt diesen Akteur um die angegebene Distanz in Blickrichtung.
<code>protected void removeTouching(Class cls)</code>	Entfernt ein Objekt der angegebenen Klasse, das der Akteur aktuell berührt (falls es eines gibt).
<code>void setImage(GreenfootImage image)</code>	Setzt das Bild dieses Objekts auf das angegebene Bild.
<code>void setImage(String filename)</code>	Setzt ein Bild dieses Objekts aus einer Bilddatei.
<code>void setLocation(int x, int y)</code>	Weist diesem Objekt eine neue Position zu.
<code>void setRotation(int rotation)</code>	Setzt die Rotation für dieses Objekt.
<code>void turn(int amount)</code>	Dreht diesen Akteur um die angegebene Gradzahl.
<code>void turnTowards(int x, int y)</code>	Dreht diesen Akteur in die Blickrichtung der angegebenen Position.

## Methoden der Klasse Greenfoot

<code>Greenfoot()</code>	Konstruktor
<code>static void delay(int time)</code>	Verzögert die Ausführung um eine Anzahl von Zeitschritten. Die Dauer eines Zeitschritts wird durch den Geschwindigkeitsregler definiert.
<code>static String getKey()</code>	Liefert die zuletzt gedrückte Taste, seit die Methode das letzte Mal aufgerufen wurde.
<code>static int getMicLevel()</code>	Liefert den Eingangspiegel des Mikrofons.
<code>static MouseInfo getMouseInfo()</code>	Liefert ein <b>MouseInfo</b> -Objekt mit Informationen über den Zustand der Maus zurück.
<code>static int getRandomNumber(int limit)</code>	Liefert eine Zufallszahl zwischen 0 (einschließlich) und <b>limit</b> (ausschließlich).
<code>static boolean isKeyDown(String keyName)</code>	Prüft, ob eine angegebene Taste gerade gedrückt wird.
<code>static boolean mouseClicked(Object obj)</code>	<b>True</b> , wenn mit der Maus auf das angegebene Objekt geklickt wurde.
<code>static boolean mouseDragEnded(Object obj)</code>	<b>True</b> , wenn das Ziehen mit der Maus beendet wurde.
<code>static boolean mouseDragged(Object obj)</code>	<b>True</b> , wenn die Maus auf das angegebene Objekt gezogen wurde.
<code>static boolean mouseMoved(Object obj)</code>	<b>True</b> , wenn die Maus auf das angegebene Objekt bewegt wurde.

## Methoden der Klasse Greenfoot

<code>static boolean mousePressed(Object obj)</code>	<b>True</b> , wenn die Maus auf dem angegebenen Objekt gedrückt wurde.
<code>static void playSound(String soundFile)</code>	Spielt einen Sound aus einer Datei ab.
<code>static void setSpeed(int speed)</code>	Setzt die Ausführungsgeschwindigkeit der Simulation.
<code>static void setWorld(World world)</code>	Setzt die auszuführende Welt auf die angegebene Welt.
<code>static void start()</code>	Startet die Simulation oder fährt mit einer angehaltenen Simulation fort.
<code>static void stop()</code>	Stoppt die Simulation.

## Methoden der Klasse MouseInfo

<code>Actor getActor()</code>	Liefert den Akteur (falls vorhanden), auf den sich die aktuelle Mausektion bezieht.
<code>int getButton()</code>	Die Nummer der gedrückten oder geklickten Maus-taste (falls vorhanden).
<code>int getClickCount()</code>	Die Anzahl der Mausklicks dieses Mauseignisses.
<code>int getX()</code>	Die aktuelle x-Position des Mausursors.
<code>int getY()</code>	Die aktuelle y-Position des Mausursors.
<code>String toString()</code>	Liefert eine String-Repräsentation dieser Mause-reignis-Info zurück.

## Methoden der Klasse GreenfootImage

<code>GreenfootImage(GreenfootImage image)</code>	Erzeugt ein <b>GreenfootImage</b> -Objekt aus einem anderen <b>GreenfootImage</b> -Objekt.
<code>GreenfootImage(int width, int height)</code>	Erzeugt ein leeres (transparentes) Bild der angegebenen Größe.
<code>GreenfootImage(String filename)</code>	Erzeugt ein Bild aus einer Bilddatei.
<code>GreenfootImage(String string, int size, Color foreground, Color background)</code>	Erzeugt ein Bild mit der angegebenen Zeichenkette als Text in der angegebenen Schriftgröße, mit den angegebenen Farben für Vorder- und Hintergrund.
<code>GreenfootImage(String string, int size, Color foreground, Color background, Color outline)</code>	Erzeugt ein Bild mit der angegebenen Zeichenkette als Text in der angegebenen Schriftgröße, mit den angegebenen Farben für Vorder- und Hintergrund und Kontur.
<code>void clear()</code>	Löscht das Bild.
<code>void drawImage(GreenfootImage image, int x, int y)</code>	Zeichnet das angegebene Bild in dieses Bild.
<code>void drawLine(int x1, int y1, int x2, int y2)</code>	Zeichnet eine Linie vom Punkt (x1, y1) zu Punkt (x2, y2) in der aktuellen Zeichenfarbe.
<code>void drawOval(int x, int y, int width, int height)</code>	Zeichnet in der aktuellen Zeichenfarbe ein Oval, das von dem angegebenen Rechteck begrenzt wird.

## Methoden der Klasse GreenfootImage

<code>void drawPolygon(int[] xPoints, int[] yPoints, int nPoints)</code>	Zeichnet ein geschlossenes Polygon, das durch Felder von <i>x</i> - und <i>y</i> -Koordinaten definiert wird.
<code>void drawRect(int x, int y, int width, int height)</code>	Zeichnet den Umriss des angegebenen Rechtecks.
<code>void drawShape(Shape shape)</code>	Zeichnet eine Form direkt auf das Bild.
<code>void drawstring(String string, int x, int y)</code>	Zeichnet den durch <b>string</b> vorgegebenen Text in der aktuellen Schriftart und Farbe.
<code>void fill()</code>	Füllt das gesamte Bild mit der aktuellen Zeichenfarbe.
<code>void fillOval(int x, int y, int width, int height)</code>	Füllt ein Oval, das von dem angegebenen Rechteck begrenzt wird, mit der aktuellen Zeichenfarbe.
<code>void fillPolygon(int[] xPoints, int[] yPoints, int nPoints)</code>	Füllt ein geschlossenes Polygon, das durch Felder von <i>x</i> - und <i>y</i> -Koordinaten definiert wird, mit der aktuellen Zeichenfarbe.
<code>void fillRect(int x, int y, int width, int height)</code>	Füllt das angegebene Rechteck.
<code>void fillShape(Shape shape)</code>	Zeichnet eine ausgefüllte Form direkt auf das Bild.
<code>BufferedImage getAwtImage()</code>	Liefert das <b>BufferedImage</b> zurück, das hinter dem <b>GreenfootImage</b> steht.
<code>Color getColor()</code>	Liefert die aktuelle Zeichenfarbe zurück.
<code>Color getColorAt(int x, int y)</code>	Liefert die Farbe eines bestimmten Pixels zurück.
<code>Font getFont()</code>	Liefert die aktuelle Schriftart zurück.
<code>int getHeight()</code>	Liefert die Höhe des Bildes zurück.
<code>int getTransparency()</code>	Liefert die Transparenz des Bildes (im Bereich 0–255) zurück.
<code>int getWidth()</code>	Liefert die Breite des Bildes zurück.
<code>void mirrorHorizontally()</code>	Spiegelt das Bild horizontal (um die <i>x</i> -Achse).
<code>void mirrorVertically()</code>	Spiegelt das Bild vertikal (um die <i>y</i> -Achse).
<code>void rotate(int degrees)</code>	Dreht das Bild um seinen Mittelpunkt.
<code>void scale(int width, int height)</code>	Skaliert das Bild auf eine neue Größe.
<code>void setColor(Color color)</code>	Setzt die aktuelle Zeichenfarbe.
<code>void setColorAt(int x, int y, Color color)</code>	Setzt die Farbe eines bestimmten Pixels auf die gegebene Farbe.
<code>void setFont(Font f)</code>	Setzt die aktuelle Schriftart.
<code>void setTransparency(int t)</code>	Setzt die Transparenz des Bildes (im Bereich 0–255).
<code>String toString()</code>	Liefert eine String-Repräsentation dieses Bildes zurück.

## Methoden der Klasse GreenfootSound

<code>GreenfootSound(String filename)</code>	Erzeugt einen neuen Sound aus der angegebenen Datei.
<code>int getVolume()</code>	Liefert die aktuelle Lautstärke des Sounds, ein Wert zwischen 0 (aus) und 100 (maximale Lautstärke).
<code>boolean isPlaying()</code>	<b>True</b> , wenn der Sound aktuell abgespielt wird.

Methoden der Klasse `GreenfootSound`

<code>void pause()</code>	Pausiert den Sound, falls dieser aktuell abgespielt wird.
<code>void play()</code>	Startet das Abspielen des Sounds.
<code>void playLoop()</code>	Spielt diesen Sound wiederholt in einer Schleife ab.
<code>int setVolume(int level)</code>	Setzt die aktuelle Lautstärke des Sounds auf einen Wert zwischen 0 (aus) und 100 (maximale Lautstärke).
<code>void stop()</code>	Stoppt den Sound, falls dieser aktuell abgespielt wird.
<code>String toString()</code>	Liefert eine String-Repräsentation dieses Sounds zurück, der den Namen der Datei enthält und angibt, ob der Sound aktuell abgespielt wird oder nicht.

Methoden und Felder der Klasse `UserInfo`

<code>static int NUM_INTS</code>	Anzahl der <code>int</code> -Zahlen, die gespeichert werden können.
<code>static int NUM_STRINGS</code>	Anzahl der Strings, die gespeichert werden können.
<code>static int STRING_LENGTH_LIMIT</code>	Maximale Anzahl der Zeichen, die in jedem String gespeichert werden können.
<code>int getInt(int index)</code>	Liefert den Wert von <code>int</code> an der angegebenen Position (0 bis einschließlich <code>NUM_INTS - 1</code> ).
<code>static UserInfo getMyInfo()</code>	Liefert die Daten, die für den aktuellen Benutzer gespeichert sind.
<code>static List getNearby(int maxAmount)</code>	Liefert eine sortierte Liste mit <code>UserInfo</code> -Einträgen dieses Szenarios zurück, die sich in der Nähe des aktuellen Benutzers befinden.
<code>int getRank()</code>	Liefert die Gesamtwertung des Benutzers für dieses Szenario.
<code>int getScore()</code>	Liefert den Punktestand des Benutzers.
<code>String getString(int index)</code>	Liefert den Wert der Zeichenkette an der angegebenen Position (0 bis einschließlich <code>NUM_STRINGS - 1</code> ).
<code>static List getTop(int maxAmount)</code>	Liefert eine (von oben nach unten) sortierte Liste mit <code>UserInfo</code> -Einträgen dieses Szenarios.
<code>GreenfootImage getUserImage()</code>	Liefert ein Bild des Benutzers zurück.
<code>String getUsername()</code>	Liefert den Benutzernamen des Benutzers, zu dem dieser Speicher gehört.
<code>static boolean isStorageAvailable()</code>	Gibt an, ob der Speicher verfügbar ist.
<code>void setInt(int index, int value)</code>	Setzt den Wert des Eintrags an der angegebenen Position (0 bis einschließlich <code>NUM_INTS - 1</code> ).
<code>void setScore(int score)</code>	Setzt den Punktestand des Benutzers.
<code>void setString(int index, String value)</code>	Setzt den Wert des Strings an der angegebenen Position (0 bis einschließlich <code>NUM_STRINGS - 1</code> ).