

TerminalArena

0.1

Erzeugt von Doxygen 1.8.20



<b>1 Verzeichnis der Namensbereiche</b>	<b>1</b>
1.1 Pakete	1
<b>2 Hierarchie-Verzeichnis</b>	<b>3</b>
2.1 Klassenhierarchie	3
<b>3 Klassen-Verzeichnis</b>	<b>5</b>
3.1 Auflistung der Klassen	5
<b>4 Dokumentation der Namensbereiche</b>	<b>7</b>
4.1 TerminalArena-Namensbereichsreferenz	7
4.2 TerminalArena.Classes-Namensbereichsreferenz	7
4.3 TerminalArena.Helpers-Namensbereichsreferenz	8
4.4 TerminalArena.Interfaces-Namensbereichsreferenz	8
<b>5 Klassen-Dokumentation</b>	<b>9</b>
5.1 TerminalArena.Classes.Action Klassenreferenz	9
5.1.1 Ausführliche Beschreibung	9
5.1.2 Beschreibung der Konstruktoren und Destruktoren	9
5.1.2.1 Action()	9
5.1.3 Dokumentation der Property's	10
5.1.3.1 Length	10
5.1.3.2 Name	10
5.1.3.3 What	10
5.2 TerminalArena.Classes.Arena Klassenreferenz	10
5.2.1 Ausführliche Beschreibung	11
5.2.2 Beschreibung der Konstruktoren und Destruktoren	11
5.2.2.1 Arena()	11
5.2.3 Dokumentation der Elementfunktionen	11
5.2.3.1 StartFight()	11
5.2.4 Dokumentation der Property's	11
5.2.4.1 Ui	12
5.3 TerminalArena.Classes.Coordinate Klassenreferenz	12
5.3.1 Ausführliche Beschreibung	12
5.3.2 Beschreibung der Konstruktoren und Destruktoren	12
5.3.2.1 Coordinate() [1/2]	12
5.3.2.2 Coordinate() [2/2]	13
5.3.3 Dokumentation der Elementfunktionen	13
5.3.3.1 NextLine()	13
5.3.3.2 Reset()	13
5.3.4 Dokumentation der Property's	13
5.3.4.1 X	14
5.3.4.2 Y	14
5.4 TerminalArena.Helpers.Dice Klassenreferenz	14

5.4.1 Ausführliche Beschreibung	14
5.4.2 Beschreibung der Konstruktoren und Destruktoren	14
5.4.2.1 Dice() [1/2]	14
5.4.2.2 Dice() [2/2]	15
5.4.3 Dokumentation der Elementfunktionen	15
5.4.3.1 GetSides()	15
5.4.3.2 Roll()	15
5.4.3.3 ToString()	16
5.5 TerminalArena.Classes.Fight Klassenreferenz	16
5.5.1 Ausführliche Beschreibung	16
5.5.2 Beschreibung der Konstruktoren und Destruktoren	16
5.5.2.1 Fight()	16
5.5.3 Dokumentation der Elementfunktionen	17
5.5.3.1 DoFight()	17
5.6 TerminalArena.Classes.Fighter Klassenreferenz	17
5.6.1 Ausführliche Beschreibung	19
5.6.2 Beschreibung der Konstruktoren und Destruktoren	19
5.6.2.1 Fighter()	19
5.6.3 Dokumentation der Elementfunktionen	19
5.6.3.1 Attack()	19
5.6.3.2 BarGraphic()	20
5.6.3.3 Defend()	20
5.6.3.4 DoAction()	20
5.6.3.5 HealthBar()	21
5.6.3.6 IsAlive()	21
5.6.3.7 ToString()	21
5.6.4 Dokumentation der Property's	22
5.6.4.1 Actions	22
5.6.4.2 Damage	22
5.6.4.3 Defense	22
5.6.4.4 Health	22
5.6.4.5 Message	22
5.6.4.6 Name	23
5.7 TerminalArena.Classes.HealthPotion Klassenreferenz	23
5.7.1 Ausführliche Beschreibung	24
5.7.2 Beschreibung der Konstruktoren und Destruktoren	24
5.7.2.1 HealthPotion() [1/2]	24
5.7.2.2 HealthPotion() [2/2]	24
5.7.3 Dokumentation der Property's	25
5.7.3.1 HpBonus	25
5.7.3.2 Name	25
5.7.3.3 Value	25

---

5.8 TerminalArena.Classes.Helper Klassenreferenz . . . . .	25
5.8.1 Ausführliche Beschreibung . . . . .	26
5.8.2 Dokumentation der Elementfunktionen . . . . .	26
5.8.2.1 GetRandomName() . . . . .	26
5.9 TerminalArena.Interfaces.IEntity Schnittstellenreferenz . . . . .	26
5.9.1 Ausführliche Beschreibung . . . . .	27
5.9.2 Dokumentation der Elementfunktionen . . . . .	27
5.9.2.1 Attack() . . . . .	27
5.9.2.2 Defend() . . . . .	28
5.9.3 Dokumentation der Property's . . . . .	28
5.9.3.1 Damage . . . . .	28
5.9.3.2 Defense . . . . .	28
5.9.3.3 Health . . . . .	28
5.9.3.4 Message . . . . .	29
5.9.3.5 Name . . . . .	29
5.10 TerminalArena.Interfaces.IItem Schnittstellenreferenz . . . . .	29
5.10.1 Ausführliche Beschreibung . . . . .	30
5.10.2 Dokumentation der Property's . . . . .	30
5.10.2.1 Name . . . . .	30
5.10.2.2 Value . . . . .	30
5.11 TerminalArena.Classes.Inventory Klassenreferenz . . . . .	30
5.11.1 Ausführliche Beschreibung . . . . .	31
5.11.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	31
5.11.2.1 Inventory() . . . . .	31
5.11.3 Dokumentation der Elementfunktionen . . . . .	31
5.11.3.1 Contains() . . . . .	31
5.11.4 Dokumentation der Datenelemente . . . . .	32
5.11.4.1 Capacity . . . . .	32
5.11.5 Dokumentation der Property's . . . . .	32
5.11.5.1 Items . . . . .	32
5.12 TerminalArena.Classes.LargeHealthPotion Klassenreferenz . . . . .	33
5.12.1 Ausführliche Beschreibung . . . . .	34
5.12.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	34
5.12.2.1 LargeHealthPotion() . . . . .	34
5.13 TerminalArena.Classes.MediumHealthPotion Klassenreferenz . . . . .	34
5.13.1 Ausführliche Beschreibung . . . . .	35
5.13.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	35
5.13.2.1 MediumHealthPotion() . . . . .	35
5.14 TerminalArena.Program Klassenreferenz . . . . .	36
5.14.1 Ausführliche Beschreibung . . . . .	36
5.15 TerminalArena.Classes.SmallHealthPotion Klassenreferenz . . . . .	36
5.15.1 Ausführliche Beschreibung . . . . .	37

5.15.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	37
5.15.2.1 SmallHealthPotion() . . . . .	37
5.16 TerminalArena.Classes.UserInterface Klassenreferenz . . . . .	37
5.16.1 Ausführliche Beschreibung . . . . .	38
5.16.2 Beschreibung der Konstruktoren und Destruktoren . . . . .	38
5.16.2.1 UserInterface() . . . . .	38
5.16.3 Dokumentation der Elementfunktionen . . . . .	38
5.16.3.1 ClearActionArea() . . . . .	39
5.16.3.2 ClearContentArea() . . . . .	39
5.16.3.3 DrawActionBorder() . . . . .	39
5.16.3.4 DrawTitle() . . . . .	39
5.16.3.5 Print() . . . . .	39
5.16.3.6 PrintActions() . . . . .	40
5.16.3.7 PrintInfo() . . . . .	40
5.16.3.8 PrintLine() . . . . .	40
5.16.3.9 PrintMenu() . . . . .	41
<b>Index</b>	<b>43</b>

# Kapitel 1

## Verzeichnis der Namensbereiche

### 1.1 Pakete

Hier folgen die Pakete mit einer Kurzbeschreibung (wenn verfügbar):

<a href="#">TerminalArena</a>	7
<a href="#">TerminalArena.Classes</a>	7
<a href="#">TerminalArena.Helpers</a>	8
<a href="#">TerminalArena.Interfaces</a>	8





## Kapitel 2

# Hierarchie-Verzeichnis

### 2.1 Klassenhierarchie

Die Liste der Ableitungen ist -mit Einschränkungen- alphabetisch sortiert:

TerminalArena.Classes.Action . . . . .	9
TerminalArena.Classes.Arena . . . . .	10
TerminalArena.Classes.Coordinate . . . . .	12
TerminalArena.Helpers.Dice . . . . .	14
TerminalArena.Classes.Fight . . . . .	16
TerminalArena.Classes.Helper . . . . .	25
TerminalArena.Interfaces.IEntity . . . . .	26
TerminalArena.Classes.Fighter . . . . .	17
TerminalArena.Interfaces.IItem . . . . .	29
TerminalArena.Classes.HealthPotion . . . . .	23
TerminalArena.Classes.LargeHealthPotion . . . . .	33
TerminalArena.Classes.MediumHealthPotion . . . . .	34
TerminalArena.Classes.SmallHealthPotion . . . . .	36
TerminalArena.Classes.Inventory . . . . .	30
TerminalArena.Program . . . . .	36
TerminalArena.Classes.UserInterface . . . . .	37



## Kapitel 3

# Klassen-Verzeichnis

### 3.1 Auflistung der Klassen

Hier folgt die Aufzählung aller Klassen, Strukturen, Varianten und Schnittstellen mit einer Kurzbeschreibung:

<a href="#">TerminalArena.Classes.Action</a>	
Eine ausführbare Aktion . . . . .	9
<a href="#">TerminalArena.Classes.Arena</a>	
Repräsentiert die <a href="#">Arena</a> für einen Kampf . . . . .	10
<a href="#">TerminalArena.Classes.Coordinate</a>	
Klasse, um eine Koordinate in der Konsole darzustellen . . . . .	12
<a href="#">TerminalArena.Helpers.Dice</a>	
Repräsentiert einen Würfel mit konfigurierbarer Seitenanzahl . . . . .	14
<a href="#">TerminalArena.Classes.Fight</a>	
Diese Klasse repräsentiert einen Kampf . . . . .	16
<a href="#">TerminalArena.Classes.Fighter</a>	
Die Implementierung des IEntity-Interfaces. Hier wird ein Kämpfer erstellt . . . . .	17
<a href="#">TerminalArena.Classes.HealthPotion</a>	
Abstrakte Klasse für Heiltränke . . . . .	23
<a href="#">TerminalArena.Classes.Helper</a>	
Hilfsklasse. Primär für die Namensgenerierung der Kämpfer . . . . .	25
<a href="#">TerminalArena.Interfaces.IEntity</a>	
Interface für die Kämpferklasse. Eigenschaften und Methoden, die für alle Kämpfer identisch sein sollen . . . . .	26
<a href="#">TerminalArena.Interfaces.IItem</a>	
Beschreibt einen Gegenstand . . . . .	29
<a href="#">TerminalArena.Classes.Inventory</a>	
Repräsentiert das Inventar eines Kämpfers . . . . .	30
<a href="#">TerminalArena.Classes.LargeHealthPotion</a>	
Ein mittlerer Heiltrank, der 50 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse . . . . .	33
<a href="#">TerminalArena.Classes.MediumHealthPotion</a>	
Ein mittlerer Heiltrank, der 20 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse . . . . .	34
<a href="#">TerminalArena.Program</a>	
. . . . .	36
<a href="#">TerminalArena.Classes.SmallHealthPotion</a>	
Ein kleiner Heiltrank, der 5 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse . . . . .	36
<a href="#">TerminalArena.Classes.UserInterface</a>	
Klasse für das User Interface. Hier sind alle Methoden enthalten, die das Interface darstellen . . . . .	37



## Kapitel 4

# Dokumentation der Namensbereiche

### 4.1 TerminalArena-Namensbereichsreferenz

#### Klassen

- class [Program](#)

### 4.2 TerminalArena.Classes-Namensbereichsreferenz

#### Klassen

- class [Action](#)  
*Eine ausführbare Aktion.*
- class [Arena](#)  
*Repräsentiert die [Arena](#) für einen Kampf.*
- class [Coordinate](#)  
*Klasse, um eine Koordinate in der Konsole darzustellen.*
- class [Fight](#)  
*Diese Klasse repräsentiert einen Kampf.*
- class [Fighter](#)  
*Die Implementierung des IEntity-Interfaces. Hier wird ein Kämpfer erstellt.*
- class [HealthPotion](#)  
*Abstrakte Klasse für Heiltränke.*
- class [Helper](#)  
*Hilfsklasse. Primär für die Namensgenerierung der Kämpfer.*
- class [Inventory](#)  
*Repräsentiert das Inventar eines Kämpfers.*
- class [LargeHealthPotion](#)  
*Ein mittlerer Heiltrank, der 50 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.*
- class [MediumHealthPotion](#)  
*Ein mittlerer Heiltrank, der 20 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.*
- class [SmallHealthPotion](#)  
*Ein kleiner Heiltrank, der 5 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.*
- class [UserInterface](#)  
*Klasse für das User Interface. Hier sind alle Methoden enthalten, die das Interface darstellen.*

## 4.3 TerminalArena.Helpers-Namensbereichsreferenz

### Klassen

- class [Dice](#)  
*Repräsentiert einen Würfel mit konfigurierbarer Seitenanzahl.*

## 4.4 TerminalArena.Interfaces-Namensbereichsreferenz

### Klassen

- interface [IEntity](#)  
*Interface für die Kämpferklasse. Eigenschaften und Methoden, die für alle Kämpfer identisch sein sollen.*
- interface [IItem](#)  
*Beschreibt einen Gegenstand.*

## Kapitel 5

# Klassen-Dokumentation

### 5.1 TerminalArena.Classes.Action Klassenreferenz

Eine ausführbare Aktion.

#### Öffentliche Methoden

- [Action](#) (string name, string what)  
*Konstruktor einer Aktion*

#### Propertys

- string [Name](#) [get, set]  
*Der Name der Aktion. Dieser wird im Interface angezeigt*
- int [Length](#) [get]  
*Die Länge des Aktionsnamens*
- string [What](#) [get]  
*Eine Zeichenfolge, die die Aktion identifiziert*

#### 5.1.1 Ausführliche Beschreibung

Eine ausführbare Aktion.

Definiert in Zeile 7 der Datei Action.cs.

#### 5.1.2 Beschreibung der Konstruktoren und Destruktoren

##### 5.1.2.1 Action()

```
TerminalArena.Classes.Action.Action (
    string name,
    string what )
```

Konstruktor einer Aktion

**Parameter**

<i>name</i>	Der Name der Aktion
<i>what</i>	Der Identifikator

Definiert in Zeile 27 der Datei Action.cs.

### 5.1.3 Dokumentation der Propertys

#### 5.1.3.1 Length

```
int TerminalArena.Classes.Action.Length [get]
```

Die Länge des Aktionsnamens

Definiert in Zeile 16 der Datei Action.cs.

#### 5.1.3.2 Name

```
string TerminalArena.Classes.Action.Name [get], [set]
```

Der Name der Aktion. Dieser wird im Interface angezeigt

Definiert in Zeile 12 der Datei Action.cs.

#### 5.1.3.3 What

```
string TerminalArena.Classes.Action.What [get]
```

Eine Zeichenfolge, die die Aktion identifiziert

Definiert in Zeile 20 der Datei Action.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Action.cs

## 5.2 TerminalArena.Classes.Arena Klassenreferenz

Repräsentiert die [Arena](#) für einen Kampf.



## Öffentliche Methoden

- [Arena](#) ()  
*Erstellt die [Arena](#).*
- void [StartFight](#) ()  
*Beginnt den Kampf gegen alle Gegner in der Liste*

## Propertys

- [UserInterface Ui](#) [get, set]  
*Das Benutzerinterface*

### 5.2.1 Ausführliche Beschreibung

Repräsentiert die [Arena](#) für einen Kampf.

Definiert in Zeile 10 der Datei Arena.cs.

### 5.2.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.2.2.1 Arena()

```
TerminalArena.Classes.Arena.Arena ( )
```

Erstellt die [Arena](#).

Definiert in Zeile 24 der Datei Arena.cs.

### 5.2.3 Dokumentation der Elementfunktionen

#### 5.2.3.1 StartFight()

```
void TerminalArena.Classes.Arena.StartFight ( )
```

Beginnt den Kampf gegen alle Gegner in der Liste

Definiert in Zeile 33 der Datei Arena.cs.

### 5.2.4 Dokumentation der Propertys

#### 5.2.4.1 Ui

`UserInterface` TerminalArena.Classes.Arena.Ui [get], [set]

Das Benutzerinterface

Definiert in Zeile 15 der Datei Arena.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Arena.cs

### 5.3 TerminalArena.Classes.Coordinate Klassenreferenz

Klasse, um eine Koordinate in der Konsole darzustellen.

#### Öffentliche Methoden

- `Coordinate` (int x, int y)  
*Erstellt eine X/Y-Koordinate mit den angegebenen Werten.*
- `Coordinate` ()  
*Erstellt eine Koordinate aus der momentanen Cursorposition.*
- void `Reset` ()  
*Setzt den Cursor auf die X/Y-Koordinate zurück.*
- void `NextLine` ()  
*Setzt die Koordinate eine Zeile herunter.*

#### Propertys

- int `X` [get, set]  
*Position von Links*
- int `Y` [get, set]  
*Position von Oben*

#### 5.3.1 Ausführliche Beschreibung

Klasse, um eine Koordinate in der Konsole darzustellen.

Definiert in Zeile 8 der Datei Coordinate.cs.

#### 5.3.2 Beschreibung der Konstruktoren und Destruktoren

##### 5.3.2.1 `Coordinate()` [1/2]

```
TerminalArena.Classes.Coordinate.Coordinate (
    int x,
    int y )
```

Erstellt eine X/Y-Koordinate mit den angegebenen Werten.

**Parameter**

<i>x</i>	X (oder left)
<i>y</i>	Y (oder top)

Definiert in Zeile 25 der Datei Coordinate.cs.

**5.3.2.2 Coordinate() [2/2]**

```
TerminalArena.Classes.Coordinate.Coordinate ( )
```

Erstellt eine Koordinate aus der momentanen Cursorposition.

Definiert in Zeile 34 der Datei Coordinate.cs.

**5.3.3 Dokumentation der Elementfunktionen****5.3.3.1 NextLine()**

```
void TerminalArena.Classes.Coordinate.NextLine ( )
```

Setzt die Koordinate eine Zeile herunter.

Definiert in Zeile 51 der Datei Coordinate.cs.

**5.3.3.2 Reset()**

```
void TerminalArena.Classes.Coordinate.Reset ( )
```

Setzt den Cursor auf die X/Y-Koordinate zurück.

Definiert in Zeile 43 der Datei Coordinate.cs.

**5.3.4 Dokumentation der Propertys**

#### 5.3.4.1 X

```
int TerminalArena.Classes.Coordinate.X [get], [set]
```

Position von Links

Definiert in Zeile 13 der Datei Coordinate.cs.

#### 5.3.4.2 Y

```
int TerminalArena.Classes.Coordinate.Y [get], [set]
```

Position von Oben

Definiert in Zeile 18 der Datei Coordinate.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Coordinate.cs

## 5.4 TerminalArena.Helpers.Dice Klassenreferenz

Repräsentiert einen Würfel mit konfigurierbarer Seitenanzahl.

### Öffentliche Methoden

- [Dice](#) (int sides)  
*Erstellt einen Würfel mit der gewünschten Seitenzahl.*
- [Dice](#) ()  
*Erstellt einen Würfel mit 6 Seiten.*
- int [Roll](#) ()  
*Würfelt mit dem Würfel.*
- int [GetSides](#) ()  
*Gibt die Anzahl der Seiten zurück.*
- override string [ToString](#) ()  
*Gibt eine Beschreibung des "Dice"-Objekts als Zeichenfolge zurück.*

### 5.4.1 Ausführliche Beschreibung

Repräsentiert einen Würfel mit konfigurierbarer Seitenanzahl.

Definiert in Zeile 8 der Datei Dice.cs.

### 5.4.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.4.2.1 Dice() [1/2]

```
TerminalArena.Helpers.Dice.Dice (  
    int sides )
```

Erstellt einen Würfel mit der gewünschten Seitenzahl.

**Parameter**

<i>sides</i>	Seiten des Würfels
--------------	--------------------

Definiert in Zeile 17 der Datei Dice.cs.

**5.4.2.2 Dice() [2/2]**

```
TerminalArena.Helpers.Dice.Dice ( )
```

Erstellt einen Würfel mit 6 Seiten.

Definiert in Zeile 26 der Datei Dice.cs.

**5.4.3 Dokumentation der Elementfunktionen****5.4.3.1 GetSides()**

```
int TerminalArena.Helpers.Dice.GetSides ( )
```

Gibt die Anzahl der Seiten zurück.

**Rückgabe**

int

Definiert in Zeile 45 der Datei Dice.cs.

**5.4.3.2 Roll()**

```
int TerminalArena.Helpers.Dice.Roll ( )
```

Würfelt mit dem Würfel.

**Rückgabe**

int

Definiert in Zeile 36 der Datei Dice.cs.

### 5.4.3.3 ToString()

```
override string TerminalArena.Helpers.Dice.ToString ( )
```

Gibt eine Beschreibung des "Dice"-Objekts als Zeichenfolge zurück.

#### Rückgabe

string

Definiert in Zeile 54 der Datei Dice.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Dice.cs

## 5.5 TerminalArena.Classes.Fight Klassenreferenz

Diese Klasse repräsentiert einen Kampf.

### Öffentliche Methoden

- [Fight](#) ([Fighter](#) enemy, [Fighter](#) player, [UserInterface](#) ui)  
*Initialisiert einen Kampf zwischen zwei Kämpfern.*
- void [DoFight](#) ()  
*Führt den Kampf aus, bis einer der Kämpfer stirbt.*

### 5.5.1 Ausführliche Beschreibung

Diese Klasse repräsentiert einen Kampf.

Definiert in Zeile 11 der Datei Fight.cs.

### 5.5.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.5.2.1 Fight()

```
TerminalArena.Classes.Fight.Fight (
    Fighter enemy,
    Fighter player,
    UserInterface ui )
```

Initialisiert einen Kampf zwischen zwei Kämpfern.

## Parameter

<i>enemy</i>	Der Gegner
<i>player</i>	Der Spieler
<i>ui</i>	Eine Instanz des Interface

Definiert in Zeile 24 der Datei Fight.cs.

### 5.5.3 Dokumentation der Elementfunktionen

#### 5.5.3.1 DoFight()

```
void TerminalArena.Classes.Fight.DoFight ( )
```

Führt den Kampf aus, bis einer der Kämpfer stirbt.

Definiert in Zeile 48 der Datei Fight.cs.

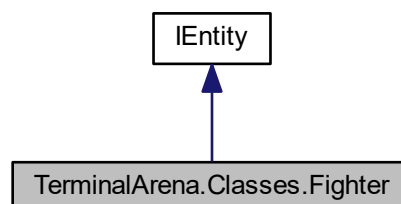
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Fight.cs

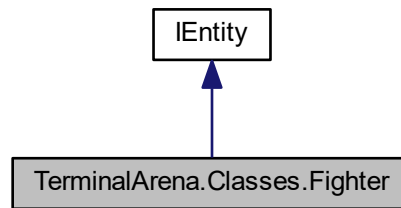
## 5.6 TerminalArena.Classes.Fighter Klassenreferenz

Die Implementierung des IEntity-Interfaces. Hier wird ein Kämpfer erstellt.

Klassendiagramm für TerminalArena.Classes.Fighter:



Zusammengehörigkeiten von TerminalArena.Classes.Fighter:



## Öffentliche Methoden

- `Fighter` (string name, int health, int damage, int defense)  
*Erstellt einen Kämpfer mit den angegebenen Attributen*
- virtual void `Attack` (`IEntity` enemy)  
*Greift den spezifizierten Gegner an*
- void `Defend` (int damage)  
*Versucht, sich gegen einen Angriff zu verteidigen*
- bool `IsAlive` ()  
*Gibt den Zustand des Kämpfers zurück*
- string `HealthBar` ()  
*Öffentliche Methode zur Darstellung einer Gesundheitsleiste*
- override string `ToString` ()  
*Liefert die Repräsentation des Fighter-Objekts als String*
- void `DoAction` (string action)  
*Führt die gewünschte Aktion aus und aktualisiert die Aktionen*

## Geschützte Methoden

- string `BarGraphic` (int current, int maximum)  
*Erstellt aus gegenwärtiger und maximaler Resource eine Leiste Dient der grafischen Anzeige der Resource*

## Property

- int `Health` [get, set]  
*Die Gesundheit des Kämpfers*
- int `Damage` [get, set]  
*Der Schaden, den der Kämpfer verursachen kann*
- int `Defense` [get, set]  
*Die Verteidigung des Kämpfers*
- string `Name` [get, set]  
*Der Name des Kämpfers*
- string `Message` [get, set]  
*Eine Nachricht über ein Ereignis*
- List< `Action` > `Actions` [get, set]  
*Eine Liste von Aktionen*



### 5.6.1 Ausführliche Beschreibung

Die Implementierung des IEntity-Interfaces. Hier wird ein Kämpfer erstellt.

Definiert in Zeile 14 der Datei Fighter.cs.

### 5.6.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.6.2.1 Fighter()

```
TerminalArena.Classes.Fighter.Fighter (
    string name,
    int health,
    int damage,
    int defense )
```

Erstellt einen Kämpfer mit den angegebenen Attributen

Parameter

<i>name</i>	Der Name des Kämpfers
<i>health</i>	Die Gesundheit des Kämpfers
<i>damage</i>	Der Schaden des Kämpfers
<i>defense</i>	Die Verteidigung des Kämpfers

Definiert in Zeile 74 der Datei Fighter.cs.

### 5.6.3 Dokumentation der Elementfunktionen

#### 5.6.3.1 Attack()

```
virtual void TerminalArena.Classes.Fighter.Attack (
    IEntity enemy ) [virtual]
```

Greift den spezifizierten Gegner an

Parameter

<i>enemy</i>	Der anzugreifende Gegner
--------------	--------------------------

Implementiert [TerminalArena.Interfaces.IEntity](#).

Definiert in Zeile 93 der Datei Fighter.cs.

### 5.6.3.2 BarGraphic()

```
string TerminalArena.Classes.Fighter.BarGraphic (
    int current,
    int maximum ) [protected]
```

Erstellt aus gegenwärtiger und maximaler Resource eine Leiste Dient der grafischen Anzeige der Resource

#### Parameter

<i>current</i>	Gegenwärtige Resource
<i>maximum</i>	Maximale Resource

#### Rückgabe

Eine Zeichenfolge, die die Resource repräsentiert

Definiert in Zeile 142 der Datei Fighter.cs.

### 5.6.3.3 Defend()

```
void TerminalArena.Classes.Fighter.Defend (
    int damage )
```

Versucht, sich gegen einen Angriff zu verteidigen

#### Parameter

<i>damage</i>	Zu verursachender Schaden
---------------	---------------------------

Implementiert [TerminalArena.Interfaces.IEntity](#).

Definiert in Zeile 104 der Datei Fighter.cs.

### 5.6.3.4 DoAction()

```
void TerminalArena.Classes.Fighter.DoAction (
    string action )
```

Führt die gewünschte Aktion aus und aktualisiert die Aktionen

**Parameter**

<i>action</i>	Die gewünschte Aktion
---------------	-----------------------

Definiert in Zeile 183 der Datei Fighter.cs.

**5.6.3.5 HealthBar()**

```
string TerminalArena.Classes.Fighter.HealthBar ( )
```

Öffentliche Methode zur Darstellung einer Gesundheitsleiste

**Rückgabe**

Eine Zeichenfolge, die die Gesundheit repräsentiert

Definiert in Zeile 164 der Datei Fighter.cs.

**5.6.3.6 IsAlive()**

```
bool TerminalArena.Classes.Fighter.IsAlive ( )
```

Gibt den Zustand des Kämpfers zurück

**Rückgabe**

Lebendig ja/nein

Definiert in Zeile 130 der Datei Fighter.cs.

**5.6.3.7 ToString()**

```
override string TerminalArena.Classes.Fighter.ToString ( )
```

Liefert die Repräsentation des Fighter-Objekts als String

**Rückgabe**

(string) Fighter-Objekt

Definiert in Zeile 174 der Datei Fighter.cs.

## 5.6.4 Dokumentation der Property's

### 5.6.4.1 Actions

```
List<Action> TerminalArena.Classes.Fighter.Actions [get], [set]
```

Eine Liste von Aktionen

Definiert in Zeile 61 der Datei Fighter.cs.

### 5.6.4.2 Damage

```
int TerminalArena.Classes.Fighter.Damage [get], [set]
```

Der Schaden, den der Kämpfer verursachen kann

Definiert in Zeile 26 der Datei Fighter.cs.

### 5.6.4.3 Defense

```
int TerminalArena.Classes.Fighter.Defense [get], [set]
```

Die Verteidigung des Kämpfers

Definiert in Zeile 31 der Datei Fighter.cs.

### 5.6.4.4 Health

```
int TerminalArena.Classes.Fighter.Health [get], [set]
```

Die Gesundheit des Kämpfers

Definiert in Zeile 21 der Datei Fighter.cs.

### 5.6.4.5 Message

```
string TerminalArena.Classes.Fighter.Message [get], [set]
```

Eine Nachricht über ein Ereignis

Definiert in Zeile 46 der Datei Fighter.cs.

#### 5.6.4.6 Name

```
string TerminalArena.Classes.Fighter.Name [get], [set]
```

Der Name des Kämpfers

Definiert in Zeile 41 der Datei Fighter.cs.

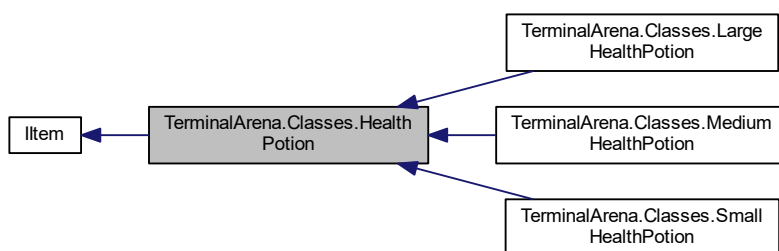
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Fighter.cs

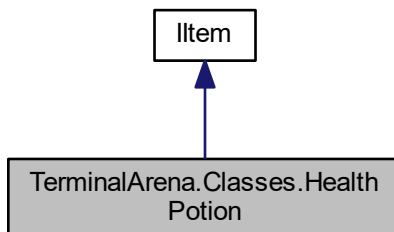
## 5.7 TerminalArena.Classes.HealthPotion Klassenreferenz

Abstrakte Klasse für Heiltränke.

Klassendiagramm für TerminalArena.Classes.HealthPotion:



Zusammengehörigkeiten von TerminalArena.Classes.HealthPotion:



## Geschützte Methoden

- [HealthPotion](#) (int hpBonus)  
*Erstellt einen Heiltrank mit der angegebenen Stärke.*
- [HealthPotion](#) ()  
*Erstellt einen Heiltrank mit der Stärke 10.*

## Property

- string [Name](#) [get, set]  
*Der Name des Heiltranks*
- int [Value](#) [get, set]  
*Der Wert des Heiltranks*
- int [HpBonus](#) [get, set]  
*Die Anzahl an Lebenspunkten, die der Heiltrank wiederherstellt.*

### 5.7.1 Ausführliche Beschreibung

Abstrakte Klasse für Heiltränke.

Definiert in Zeile 8 der Datei HealthPotion.cs.

### 5.7.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.7.2.1 HealthPotion() [1/2]

```
TerminalArena.Classes.HealthPotion.HealthPotion (
    int hpBonus ) [protected]
```

Erstellt einen Heiltrank mit der angegebenen Stärke.

Parameter

<i>hpBonus</i>	Die Stärke des Tranks
----------------	-----------------------

Definiert in Zeile 27 der Datei HealthPotion.cs.

#### 5.7.2.2 HealthPotion() [2/2]

```
TerminalArena.Classes.HealthPotion.HealthPotion ( ) [protected]
```

Erstellt einen Heiltrank mit der Stärke 10.

Definiert in Zeile 34 der Datei HealthPotion.cs.

## 5.7.3 Dokumentation der Property's

### 5.7.3.1 HpBonus

```
int TerminalArena.Classes.HealthPotion.HpBonus [get], [set]
```

Die Anzahl an Lebenspunkten, die der Heiltrank wiederherstellt.

Definiert in Zeile 21 der Datei HealthPotion.cs.

### 5.7.3.2 Name

```
string TerminalArena.Classes.HealthPotion.Name [get], [set]
```

Der Name des Heiltranks

Definiert in Zeile 13 der Datei HealthPotion.cs.

### 5.7.3.3 Value

```
int TerminalArena.Classes.HealthPotion.Value [get], [set]
```

Der Wert des Heiltranks

Definiert in Zeile 17 der Datei HealthPotion.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- HealthPotion.cs

## 5.8 TerminalArena.Classes.Helper Klassenreferenz

Hilfsklasse. Primär für die Namensgenerierung der Kämpfer.

### Öffentliche, statische Methoden

- static string [GetRandomName](#) ()  
*Gibt einen zufälligen Namen aus der names.json-Datei zurück.*

### 5.8.1 Ausführliche Beschreibung

Hilfsklasse. Primär für die Namensgenerierung der Kämpfer.

Definiert in Zeile 11 der Datei Helper.cs.

### 5.8.2 Dokumentation der Elementfunktionen

#### 5.8.2.1 GetRandomName()

```
static string TerminalArena.Classes.Helper.GetRandomName ( ) [static]
```

Gibt einen zufälligen Namen aus der names.json-Datei zurück.

#### Rückgabe

string

Definiert in Zeile 17 der Datei Helper.cs.

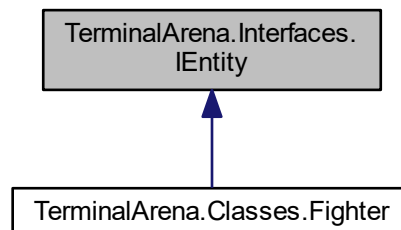
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Helper.cs

## 5.9 TerminalArena.Interfaces.IEntity Schnittstellenreferenz

Interface für die Kämpferklasse. Eigenschaften und Methoden, die für alle Kämpfer identisch sein sollen.

Klassendiagramm für TerminalArena.Interfaces.IEntity:





## Öffentliche Methoden

- void [Attack](#) (IEntity enemy)  
*Greift ein Ziel an. Der Schaden und der Verteidigungswert des Gegners werden berechnet.*
- void [Defend](#) (int damage)  
*Verteidigung gegen einen Angriff. Hier wird entschieden, ob und wie viel Schaden verursacht wird.*

## Property

- int [Health](#) [get, set]  
*Gesundheit des Kämpfers*
- int [Damage](#) [get, set]  
*Schadenswert des Kämpfers*
- int [Defense](#) [get, set]  
*Verteidigungswert des Kämpfers*
- string [Name](#) [get, set]  
*Der Name des Kämpfers*
- string [Message](#) [get, set]  
*Eine Nachricht über Ereignisse*

### 5.9.1 Ausführliche Beschreibung

Interface für die Kämpferklasse. Eigenschaften und Methoden, die für alle Kämpfer identisch sein sollen.

Definiert in Zeile 7 der Datei IEntity.cs.

### 5.9.2 Dokumentation der Elementfunktionen

#### 5.9.2.1 Attack()

```
void TerminalArena.Interfaces.IEntity.Attack (  
    IEntity enemy )
```

Greift ein Ziel an. Der Schaden und der Verteidigungswert des Gegners werden berechnet.

#### Parameter

<i>enemy</i>	Der anzugreifende Gegner
--------------	--------------------------

Implementiert in [TerminalArena.Classes.Fighter](#).

### 5.9.2.2 Defend()

```
void TerminalArena.Interfaces.IEntity.Defend (
    int damage )
```

Verteidigung gegen einen Angriff. Hier wird entschieden, ob und wie viel Schaden verursacht wird.

#### Parameter

<i>damage</i>	Der Schaden, der verursacht werden soll
---------------	---

Implementiert in [TerminalArena.Classes.Fighter](#).

## 5.9.3 Dokumentation der Propertys

### 5.9.3.1 Damage

```
int TerminalArena.Interfaces.IEntity.Damage [get], [set]
```

Schadenswert des Kämpfers

Definiert in Zeile 17 der Datei IEntity.cs.

### 5.9.3.2 Defense

```
int TerminalArena.Interfaces.IEntity.Defense [get], [set]
```

Verteidigungswert des Kämpfers

Definiert in Zeile 22 der Datei IEntity.cs.

### 5.9.3.3 Health

```
int TerminalArena.Interfaces.IEntity.Health [get], [set]
```

Gesundheit des Kämpfers

Definiert in Zeile 12 der Datei IEntity.cs.

#### 5.9.3.4 Message

```
string TerminalArena.Interfaces.IEntity.Message [get], [set]
```

Eine Nachricht über Ereignisse

Definiert in Zeile 32 der Datei IEntity.cs.

#### 5.9.3.5 Name

```
string TerminalArena.Interfaces.IEntity.Name [get], [set]
```

Der Name des Kämpfers

Definiert in Zeile 27 der Datei IEntity.cs.

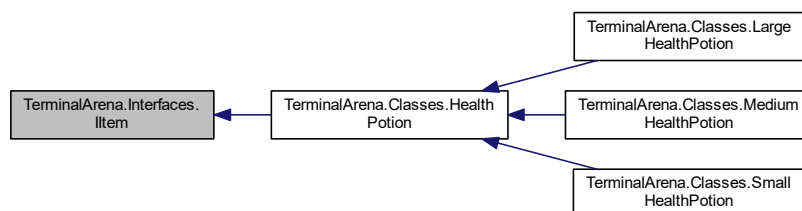
Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- IEntity.cs

## 5.10 TerminalArena.Interfaces.IItem Schnittstellenreferenz

Beschreibt einen Gegenstand.

Klassendiagramm für TerminalArena.Interfaces.IItem:



### Propertys

- string **Name** [get, set]  
*Der Name des Gegenstandes*
- int **Value** [get, set]  
*Der Wert des Gegenstandes*

### 5.10.1 Ausführliche Beschreibung

Beschreibt einen Gegenstand.

Definiert in Zeile 6 der Datei IItem.cs.

### 5.10.2 Dokumentation der Property's

#### 5.10.2.1 Name

```
string TerminalArena.Interfaces.IItem.Name [get], [set]
```

Der Name des Gegenstandes

Definiert in Zeile 11 der Datei IItem.cs.

#### 5.10.2.2 Value

```
int TerminalArena.Interfaces.IItem.Value [get], [set]
```

Der Wert des Gegenstandes

Definiert in Zeile 16 der Datei IItem.cs.

Die Dokumentation für diese Schnittstelle wurde erzeugt aufgrund der Datei:

- IItem.cs

## 5.11 TerminalArena.Classes.Inventory Klassenreferenz

Repräsentiert das Inventar eines Kämpfers.

### Öffentliche Methoden

- **Inventory** (int capacity=10)  
*Initialisiert ein Inventar mit einer Größe von 10 (falls nicht anders angegeben)*
- bool **Contains** (Type itemType)  
*Prüft, ob im Inventar ein Gegenstand des gesuchten Typs existiert, und gibt das Ergebnis zurück.*

### Öffentliche Attribute

- int **Capacity** => **Items**?.Count ?? 0  
*Die Kapazität des Inventars.*

## Propertys

- List< [Item](#) > [Items](#) [get]  
*Eine Liste von Gegenständen.*

### 5.11.1 Ausführliche Beschreibung

Repräsentiert das Inventar eines Kämpfers.

Definiert in Zeile 10 der Datei Inventory.cs.

### 5.11.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.11.2.1 Inventory()

```
TerminalArena.Classes.Inventory.Inventory (
    int capacity = 10 )
```

Initialisiert ein Inventar mit einer Größe von 10 (falls nicht anders angegeben)

Parameter

<i>capacity</i>	Die Kapazität des Inventars
-----------------	-----------------------------

Definiert in Zeile 26 der Datei Inventory.cs.

### 5.11.3 Dokumentation der Elementfunktionen

#### 5.11.3.1 Contains()

```
bool TerminalArena.Classes.Inventory.Contains (
    Type itemType )
```

Prüft, ob im Inventar ein Gegenstand des gesuchten Typs existiert, und gibt das Ergebnis zurück.

Parameter

<i>itemType</i>	Der Typ des Gegenstands
-----------------	-------------------------

**Rückgabe**

Wahr oder Falsch

Definiert in Zeile 36 der Datei Inventory.cs.

## 5.11.4 Dokumentation der Datenelemente

### 5.11.4.1 Capacity

```
int TerminalArena.Classes.Inventory.Capacity => Items?.Count ?? 0
```

Die Kapazität des Inventars.

Definiert in Zeile 20 der Datei Inventory.cs.

## 5.11.5 Dokumentation der Property's

### 5.11.5.1 Items

```
List<IItem> TerminalArena.Classes.Inventory.Items [get]
```

Eine Liste von Gegenständen.

Definiert in Zeile 15 der Datei Inventory.cs.

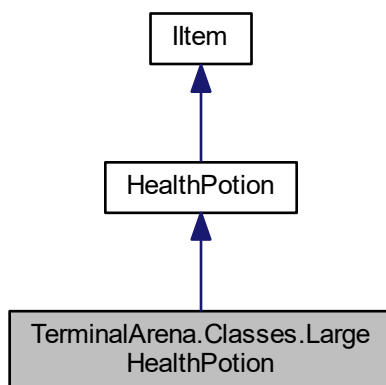
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Inventory.cs

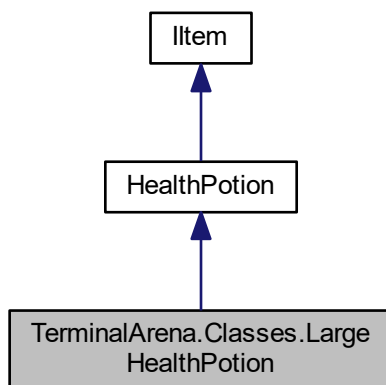
## 5.12 TerminalArena.Classes.LargeHealthPotion Klassenreferenz

Ein mittlerer Heiltrank, der 50 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Klassendiagramm für TerminalArena.Classes.LargeHealthPotion:



Zusammengehörigkeiten von TerminalArena.Classes.LargeHealthPotion:



### Öffentliche Methoden

- [LargeHealthPotion \(\)](#)

*Erstellt einen Heiltrank mit der Stärke 50*

## Weitere Geerbte Elemente

### 5.12.1 Ausführliche Beschreibung

Ein mittlerer Heiltrank, der 50 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Siehe auch

[HealthPotion](#)

Definiert in Zeile 8 der Datei LargeHealthPotion.cs.

### 5.12.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.12.2.1 LargeHealthPotion()

```
TerminalArena.Classes.LargeHealthPotion.LargeHealthPotion ( )
```

Erstellt einen Heiltrank mit der Stärke 50

Definiert in Zeile 13 der Datei LargeHealthPotion.cs.

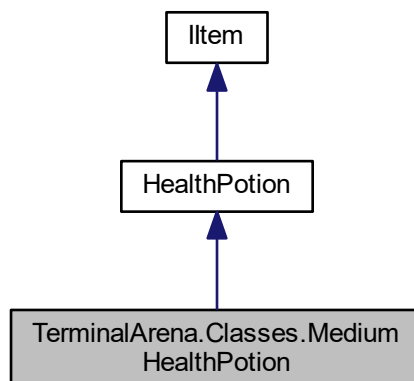
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- LargeHealthPotion.cs

## 5.13 TerminalArena.Classes.MediumHealthPotion Klassenreferenz

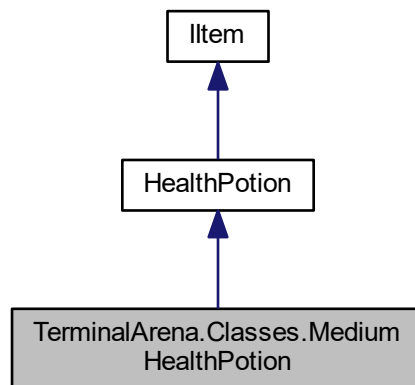
Ein mittlerer Heiltrank, der 20 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Klassendiagramm für TerminalArena.Classes.MediumHealthPotion:





Zusammengehörigkeiten von TerminalArena.Classes.MediumHealthPotion:



## Öffentliche Methoden

- [MediumHealthPotion\(\)](#)  
*Erstellt einen Heiltrank mit der Stärke 20*

## Weitere Geerbte Elemente

### 5.13.1 Ausführliche Beschreibung

Ein mittlerer Heiltrank, der 20 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Siehe auch

[HealthPotion](#)

Definiert in Zeile 8 der Datei MediumHealthPotion.cs.

### 5.13.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.13.2.1 MediumHealthPotion()

```
TerminalArena.Classes.MediumHealthPotion.MediumHealthPotion ( )
```

Erstellt einen Heiltrank mit der Stärke 20

Definiert in Zeile 13 der Datei MediumHealthPotion.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- MediumHealthPotion.cs

## 5.14 TerminalArena.Program Klassenreferenz

### 5.14.1 Ausführliche Beschreibung

Definiert in Zeile 9 der Datei Program.cs.

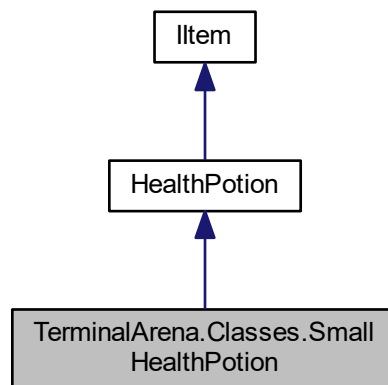
Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- Program.cs

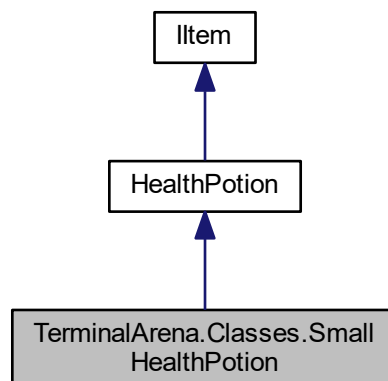
## 5.15 TerminalArena.Classes.SmallHealthPotion Klassenreferenz

Ein kleiner Heiltrank, der 5 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Klassendiagramm für TerminalArena.Classes.SmallHealthPotion:



Zusammengehörigkeiten von TerminalArena.Classes.SmallHealthPotion:



## Öffentliche Methoden

- [SmallHealthPotion](#) ()

*Erstellt einen Heiltrank mit der Stärke 5*

## Weitere Geerbte Elemente

### 5.15.1 Ausführliche Beschreibung

Ein kleiner Heiltrank, der 5 LP wiederherstellen kann. Abgeleitet von der Heiltrank-Klasse.

Siehe auch

[HealthPotion](#)

Definiert in Zeile 8 der Datei SmallHealthPotion.cs.

### 5.15.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.15.2.1 SmallHealthPotion()

```
TerminalArena.Classes.SmallHealthPotion.SmallHealthPotion ( )
```

Erstellt einen Heiltrank mit der Stärke 5

Definiert in Zeile 13 der Datei SmallHealthPotion.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- SmallHealthPotion.cs

## 5.16 TerminalArena.Classes.UserInterface Klassenreferenz

Klasse für das User Interface. Hier sind alle Methoden enthalten, die das Interface darstellen.

## Öffentliche Methoden

- `UIInterface` (string title)  
*Löscht die Inhalte der Konsole, legt globale Eigenschaften fest und zeichnet Rahmen und Titel.*
- void `DrawTitle` (string title)  
*Schreibt den Titel des Fensters.*
- int `PrintMenu` (string title, params string[] options)  
*Zeigt ein Menü mit den gegebenen Optionen an. Gibt den Index der gewählten Option zurück.*
- string `PrintActions` (List< `Action` > actions)  
*Gibt eine Liste von Aktionen als Menü aus.*
- void `DrawActionBorder` ()  
*Zeichnet den Rand des Aktionsfeldes*
- void `ClearActionArea` ()  
*Löscht den Inhalt des Aktionsbereiches*
- void `PrintInfo` ()  
*Zeigt eine Information über das Programm an.*
- void `ClearContentArea` ()  
*Löscht den Inhaltsteil des Fensters (alles unterhalb des Titels und innerhalb der Rahmen).*
- void `Print` (string message)  
*Gibt eine Zeichenfolge im Inhaltsbereich aus, ohne in die nächste Zeile zu springen (Wie Console.WriteLine).*
- void `PrintLine` (string message=null)  
*Gibt eine Zeichenfolge im Inhaltsbereich aus und springt in die nächste Zeile (wie Console.WriteLine). Ohne Zeichenfolge wird der Cursor nur in die nächste Zeile gesetzt.*

### 5.16.1 Ausführliche Beschreibung

Klasse für das User Interface. Hier sind alle Methoden enthalten, die das Interface darstellen.

Definiert in Zeile 10 der Datei `UserInterface.cs`.

### 5.16.2 Beschreibung der Konstruktoren und Destruktoren

#### 5.16.2.1 `UIInterface()`

```
TerminalArena.Classes.UIInterface.UIInterface (
    string title )
```

Löscht die Inhalte der Konsole, legt globale Eigenschaften fest und zeichnet Rahmen und Titel.

Definiert in Zeile 37 der Datei `UserInterface.cs`.

### 5.16.3 Dokumentation der Elementfunktionen

#### 5.16.3.1 ClearActionArea()

```
void TerminalArena.Classes.UserInterface.ClearActionArea ( )
```

Löscht den Inhalt des Aktionsbereiches

Definiert in Zeile 309 der Datei UserInterface.cs.

#### 5.16.3.2 ClearContentArea()

```
void TerminalArena.Classes.UserInterface.ClearContentArea ( )
```

Löscht den Inhaltsteil des Fensters (alles unterhalb des Titels und innerhalb der Rahmen).

Definiert in Zeile 344 der Datei UserInterface.cs.

#### 5.16.3.3 DrawActionBorder()

```
void TerminalArena.Classes.UserInterface.DrawActionBorder ( )
```

Zeichnet den Rand des Aktionsfeldes

Definiert in Zeile 282 der Datei UserInterface.cs.

#### 5.16.3.4 DrawTitle()

```
void TerminalArena.Classes.UserInterface.DrawTitle (
    string title )
```

Schreibt den Titel des Fensters.

##### Parameter

<i>title</i>	Der Titel
--------------	-----------

Definiert in Zeile 106 der Datei UserInterface.cs.

#### 5.16.3.5 Print()

```
void TerminalArena.Classes.UserInterface.Print (
    string message )
```

Gibt eine Zeichenfolge im Inhaltsbereich aus, ohne in die nächste Zeile zu springen (Wie Console.Write).

## Parameter

<i>message</i>	Zeichenfolge
----------------	--------------

Definiert in Zeile 371 der Datei `UserInterface.cs`.

### 5.16.3.6 PrintActions()

```
string TerminalArena.Classes.UserInterface.PrintActions (
    List< Action > actions )
```

Gibt eine Liste von Aktionen als Menü aus.

## Parameter

<i>actions</i>	Eine Liste von Aktionen
----------------	-------------------------

## Rückgabe

Die gewählte Aktion

Definiert in Zeile 204 der Datei `UserInterface.cs`.

### 5.16.3.7 PrintInfo()

```
void TerminalArena.Classes.UserInterface.PrintInfo ( )
```

Zeigt eine Information über das Programm an.

Definiert in Zeile 327 der Datei `UserInterface.cs`.

### 5.16.3.8 PrintLine()

```
void TerminalArena.Classes.UserInterface.PrintLine (
    string message = null )
```

Gibt eine Zeichenfolge im Inhaltsbereich aus und springt in die nächste Zeile (wie `Console.WriteLine`). Ohne Zeichenfolge wird der Cursor nur in die nächste Zeile gesetzt.

## Parameter

<i>message</i>	Zeichenfolge
----------------	--------------

Definiert in Zeile 384 der Datei UserInterface.cs.

#### 5.16.3.9 PrintMenu()

```
int TerminalArena.Classes.UserInterface.PrintMenu (
    string title,
    params string[] options )
```

Zeigt ein Menü mit den gegebenen Optionen an. Gibt den Index der gewählten Option zurück.

##### Parameter

<i>title</i>	Der Titel des Menüs
<i>options</i>	Eine Liste von Optionen

##### Rückgabe

Index der gewählten Option

Definiert in Zeile 135 der Datei UserInterface.cs.

Die Dokumentation für diese Klasse wurde erzeugt aufgrund der Datei:

- UserInterface.cs





# Index

- Action
  - TerminalArena.Classes.Action, [9](#)
- Actions
  - TerminalArena.Classes.Fighter, [22](#)
- Arena
  - TerminalArena.Classes.Arena, [11](#)
- Attack
  - TerminalArena.Classes.Fighter, [19](#)
  - TerminalArena.Interfaces.IEntity, [27](#)
- BarGraphic
  - TerminalArena.Classes.Fighter, [20](#)
- Capacity
  - TerminalArena.Classes.Inventory, [32](#)
- ClearActionArea
  - TerminalArena.Classes.UserInterface, [38](#)
- ClearContentArea
  - TerminalArena.Classes.UserInterface, [39](#)
- Contains
  - TerminalArena.Classes.Inventory, [31](#)
- Coordinate
  - TerminalArena.Classes.Coordinate, [12](#), [13](#)
- Damage
  - TerminalArena.Classes.Fighter, [22](#)
  - TerminalArena.Interfaces.IEntity, [28](#)
- Defend
  - TerminalArena.Classes.Fighter, [20](#)
  - TerminalArena.Interfaces.IEntity, [27](#)
- Defense
  - TerminalArena.Classes.Fighter, [22](#)
  - TerminalArena.Interfaces.IEntity, [28](#)
- Dice
  - TerminalArena.Helpers.Dice, [14](#), [15](#)
- DoAction
  - TerminalArena.Classes.Fighter, [20](#)
- DoFight
  - TerminalArena.Classes.Fight, [17](#)
- DrawActionBorder
  - TerminalArena.Classes.UserInterface, [39](#)
- DrawTitle
  - TerminalArena.Classes.UserInterface, [39](#)
- Fight
  - TerminalArena.Classes.Fight, [16](#)
- Fighter
  - TerminalArena.Classes.Fighter, [19](#)
- GetRandomName
  - TerminalArena.Classes.Helper, [26](#)
- GetSides
  - TerminalArena.Helpers.Dice, [15](#)
- Health
  - TerminalArena.Classes.Fighter, [22](#)
  - TerminalArena.Interfaces.IEntity, [28](#)
- HealthBar
  - TerminalArena.Classes.Fighter, [21](#)
- HealthPotion
  - TerminalArena.Classes.HealthPotion, [24](#)
- HpBonus
  - TerminalArena.Classes.HealthPotion, [25](#)
- Inventory
  - TerminalArena.Classes.Inventory, [31](#)
- IsAlive
  - TerminalArena.Classes.Fighter, [21](#)
- Items
  - TerminalArena.Classes.Inventory, [32](#)
- LargeHealthPotion
  - TerminalArena.Classes.LargeHealthPotion, [34](#)
- Length
  - TerminalArena.Classes.Action, [10](#)
- MediumHealthPotion
  - TerminalArena.Classes.MediumHealthPotion, [35](#)
- Message
  - TerminalArena.Classes.Fighter, [22](#)
  - TerminalArena.Interfaces.IEntity, [28](#)
- Name
  - TerminalArena.Classes.Action, [10](#)
  - TerminalArena.Classes.Fighter, [22](#)
  - TerminalArena.Classes.HealthPotion, [25](#)
  - TerminalArena.Interfaces.IEntity, [29](#)
  - TerminalArena.Interfaces.Item, [30](#)
- NextLine
  - TerminalArena.Classes.Coordinate, [13](#)
- Print
  - TerminalArena.Classes.UserInterface, [39](#)
- PrintActions
  - TerminalArena.Classes.UserInterface, [40](#)
- PrintInfo
  - TerminalArena.Classes.UserInterface, [40](#)
- PrintLine
  - TerminalArena.Classes.UserInterface, [40](#)
- PrintMenu
  - TerminalArena.Classes.UserInterface, [41](#)

- Reset
  - TerminalArena.Classes.Coordinate, 13
- Roll
  - TerminalArena.Helpers.Dice, 15
- SmallHealthPotion
  - TerminalArena.Classes.SmallHealthPotion, 37
- StartFight
  - TerminalArena.Classes.Arena, 11
- TerminalArena, 7
- TerminalArena.Classes, 7
- TerminalArena.Classes.Action, 9
  - Action, 9
  - Length, 10
  - Name, 10
  - What, 10
- TerminalArena.Classes.Arena, 10
  - Arena, 11
  - StartFight, 11
  - Ui, 11
- TerminalArena.Classes.Coordinate, 12
  - Coordinate, 12, 13
  - NextLine, 13
  - Reset, 13
  - X, 13
  - Y, 14
- TerminalArena.Classes.Fight, 16
  - DoFight, 17
  - Fight, 16
- TerminalArena.Classes.Fighter, 17
  - Actions, 22
  - Attack, 19
  - BarGraphic, 20
  - Damage, 22
  - Defend, 20
  - Defense, 22
  - DoAction, 20
  - Fighter, 19
  - Health, 22
  - HealthBar, 21
  - IsAlive, 21
  - Message, 22
  - Name, 22
  - ToString, 21
- TerminalArena.Classes.HealthPotion, 23
  - HealthPotion, 24
  - HpBonus, 25
  - Name, 25
  - Value, 25
- TerminalArena.Classes.Helper, 25
  - GetRandomName, 26
- TerminalArena.Classes.Inventory, 30
  - Capacity, 32
  - Contains, 31
  - Inventory, 31
  - Items, 32
- TerminalArena.Classes.LargeHealthPotion, 33
  - LargeHealthPotion, 34
- TerminalArena.Classes.MediumHealthPotion, 34
  - MediumHealthPotion, 35
- TerminalArena.Classes.SmallHealthPotion, 36
  - SmallHealthPotion, 37
- TerminalArena.Classes.UserInterface, 37
  - ClearActionArea, 38
  - ClearContentArea, 39
  - DrawActionBorder, 39
  - DrawTitle, 39
  - Print, 39
  - PrintActions, 40
  - PrintInfo, 40
  - PrintLine, 40
  - PrintMenu, 41
  - UserInterface, 38
- TerminalArena.Helpers, 8
- TerminalArena.Helpers.Dice, 14
  - Dice, 14, 15
  - GetSides, 15
  - Roll, 15
  - ToString, 15
- TerminalArena.Interfaces, 8
- TerminalArena.Interfaces.IEntity, 26
  - Attack, 27
  - Damage, 28
  - Defend, 27
  - Defense, 28
  - Health, 28
  - Message, 28
  - Name, 29
- TerminalArena.Interfaces.IItem, 29
  - Name, 30
  - Value, 30
- TerminalArena.Program, 36
- ToString
  - TerminalArena.Classes.Fighter, 21
  - TerminalArena.Helpers.Dice, 15
- Ui
  - TerminalArena.Classes.Arena, 11
- UserInterface
  - TerminalArena.Classes.UserInterface, 38
- Value
  - TerminalArena.Classes.HealthPotion, 25
  - TerminalArena.Interfaces.IItem, 30
- What
  - TerminalArena.Classes.Action, 10
- X
  - TerminalArena.Classes.Coordinate, 13
- Y
  - TerminalArena.Classes.Coordinate, 14