
Kerapu Documentation

P.R. Water

feb. 16, 2020

Inhoud:

1	Installatie	3
1.1	Vorbereiding	3
2	Voorbeeld	5
3	Testset	7
4	Licentie	9
5	API	11
5.1	kerapu package	11
	Python-moduleindex	31
	Index	33

Een implementatie van de Grouper in Python.

Kerapu kan eenvoudig geïnstalleerd worden met pip:

```
pip3 install kerapu
```

1.1 Voorbereiding

Alvorens gebruik te kunnen maken van Kerapu moeten de boombestanden en referentietabellen worden geconverteerd en opgeslagen. Ten tijde van schrijven van dit document was de meest recente versie van Grouper Tabellen v20180920, vervang in de onderstaande tekst deze versie voor de meest recente versie.

- Download het bestand Grouper Tabellen v20180920 van https://puc.overheid.nl/nza/doc/PUC_259930_22/.
- Extract de XML-bestanden uit het ZIP-bestand:

```
unzip -x "20190101 GrouperTabellen v20180920.zip"
```

- Converteer de XML-bestanden naar CSV (in het voorbeeld hieronder worden de CSV-bestanden weggeschreven in de folder `var/lib`):

```
kerapu kerapu:shredder "20190101 BoomBestanden v20180920.xml" var/lib/  
kerapu kerapu:shredder "20190101 Referenties v20180920.xml" var/lib/
```


Hieronder een voorbeeld om de zorgproductcode van een subtraject af te leiden.

```
from kerapu.Kerapu import Kerapu
from kerapu.lbz.Subtraject import Subtraject

# Maak een Grouper object en laad boombestanden en referentietabellen.
grouper = Kerapu()
grouper.init_static('var/lib')

# Maak een subtraject object.
subtraject = Subtraject('1',          # Subtrajectnummer
                        '0303',       # Zorgverlenersspecificatiecode
                        '0280',       # Diagnosecode
                        '11',         # Zorgtypecode
                        '000',        # Zorgvraagcode
                        '2012-01-01', # Begindatum subtraject
                        '2000-01-01', # Geboortedatum
                        'M',          # Geslachtscode
                        '01234567')   # AGB-code zorginstelling

# Voeg uitgevoerde zorgactiviteiten aan het subtraject toe.
subtraject.add_zorg_activiteit('038940', 1)
subtraject.add_zorg_activiteit('038941', 1)
subtraject.add_zorg_activiteit('190012', 1)
subtraject.add_zorg_activiteit('190015', 1)

# Bepaal zorgproductgroep en zorgproduct.
zorg_product_code = grouper.bepaal_zorg_product(subtraject)

print('Zorgproductgroepcode: {}'.format(subtraject.zorg_product_groep_code))
print('Zorgproductcode: {}'.format(subtraject.zorg_product_code))
```


HOOFDSTUK 3

Testset

De grouperbestanden bevatten sinds 2019 een testset. Deze testset is terug te vinden in bestand `var/lib/testset.csv` en zijn een onderdeel van de unittesten van Kerapu.

Het commando voor het converteren van de XML-bestanden met test data is:

```
./bin/kerapu kerapu:test-shredder ~/Downloads/20190101\ Testset\ Grouper\ RZ19b\_  
↳v20180920.zip test/var/lib/testset.csv
```


HOOFDSTUK 4

Licentie

Dit project is gelicentieerd onder de MIT-licentie.

5.1 kerapu package

5.1.1 Subpackages

kerapu.application package

Submodules

kerapu.application.KerapuApplication module

kerapu.application.kerapu module

Module contents

kerapu.boom package

Subpackages

kerapu.boom.attribuut package

Submodules

kerapu.boom.attribuut.Attribuut module

```
class kerapu.boom.attribuut.Attribuut.Attribuut (attribuut_id: int,  
                                              boom_parameter_nummer: int)
```

Basisklassen: `object`

Abstract klasse voor attributen.

tel (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type int

kerapu.boom.attribuut.Attribuut12 module

class `kerapu.boom.attribuut.Attribuut12.Attribuut12` (*attribuut_id: int,*
boom_parameter_nummer: int, filter_waarde: str)

Basisklassen: *kerapu.boom.attribuut.Attribuut.Attribuut*

Klasse voor attributen met toetswijze 1 (gelijk) en waarde type 2 (alfanumeriek).

tel (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type int

kerapu.boom.attribuut.Attribuut21 module

class `kerapu.boom.attribuut.Attribuut21.Attribuut21` (*attribuut_id: int,*
boom_parameter_nummer: int, onder_filter_waarde: int,
boven_filter_waarde: int)

Basisklassen: *kerapu.boom.attribuut.Attribuut.Attribuut*

Klasse voor attributen met toetswijze 2 (tussen) en waarde type 1 (numeriek).

tel (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen dat de boomparameter voldoet aan de voorwaarde van dit attribuut.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type int

Module contents

`kerapu.boom.attribuut.maak_attribuut` (*attribuut_id: int, boom_parameter_nummer: int, filter_toets_wijze: int, filter_waarde_type: int, onder_filter_waarde: str, boven_filter_waarde: str*) → `kerapu.boom.attribuut.Attribuut.Attribuut`

Fabriek voor het maken van attributen.

Parameters

- **attribuut_id** (*int*) – Het ID van het attribuut.
- **boom_parameter_nummer** (*int*) – Het ID van de boomparameter van het attribuut.
- **filter_toets_wijze** (*int*) – De filtertoetswijze
- **filter_waarde_type** (*int*) – Het type van de waarde van het attribuut.
- **onder_filter_waarde** (*str*) – De ondergrens.

- **boven_filter_waarde** (*str*) – De bovengrens.

Return type *Attribuut*

kerapu.boom.attribuut_groep_koppeling package

Submodules

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling module

class kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling.**AttribuutGroepKoppeling**

Basisklassen: *object*

Abstract klasse voor attribuutgroepkoppelingen.

test (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → bool
Test of een subtraject voldoet aan een attribuutgroepkoppeling.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type *bool*

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2 module

class kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2.**AttribuutGroepKoppeling2**

Basisklassen: *kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling*,
AttribuutGroepKoppeling

Klasse voor attribuutgroepkoppelingen met filtertoetswijze 2 (tussen onder- en bovengrens).

test (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → bool
Test of een subtraject voldoet aan een attribuutgroepkoppeling.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type *bool*

Module contents

`kerapu.boom.attribuut_groep_koppeling.maak_attribuut_groep_koppeling` (*attribute_groep_id:*

int, attribuut:
kerapu.boom.attribuut.Attribuut
attribuut_toets_wijze:
int, onder_toets_waarde:
int, boven_toets_waarde:
int) →
`kerapu.boom.attribuut_groep_k`

Fabriek voor het maken van attribuutgroepkoppelingen.

Parameters

- `attribute_groep_id` (*int*) – Het ID van de koppeling.
- `attribuut` (*Attribuut*) – Het attribuut van de koppeling.
- `attribuut_toets_wijze` (*int*) – De attribuuttoetswijze.
- `onder_toets_waarde` (*int*) – De ondergrens.
- `boven_toets_waarde` (*int*) – De bovengrens.

Return type *AttribuutGroepKoppeling*

`kerapu.boom.boom_parameter` package

Submodules

`kerapu.boom.boom_parameter.BehandelKlasse` module

class `kerapu.boom.boom_parameter.BehandelKlasse.BehandelKlasse` (*weeg_factor_nummer:*
int)

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter behandelklasse: som van aantal (met en zonder weegfactor).

Boomparameternummers: 351, 451, 551.

tel (*behandel_klasse_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → *int*

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subtraject voorkomt in een behandelklasse.

Parameters

- `behandel_klasse_code` (*str*) – De behandelklassecode waartegen getest moet worden.
- `subtraject` (*Subtraject*) – Het subtraject.

Return type *int*

kerapu.boom.boom_parameter.BoomParameter module**class** `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`Basisklassen: `object`

Abstracte klasse voor boomparameters.

tel (*waarde*, *subtraject*: `kerapu.lbz.Subtraject.Subtraject`) → `int`
Geeft het aantal malen dat de boomparameter voldoet aan een waarde.**Parameters**

- **waarde** (`[int|str]`) – De waarde waartegen getest moet worden.
- **subtraject** (`kerapu.lbz.Subtraject.Subtraject`) – Het subtraject.

Return type `int`**kerapu.boom.boom_parameter.DiagnoseCluster module****class** `kerapu.boom.boom_parameter.DiagnoseCluster.DiagnoseCluster` (*cluster_nummer*:
`int`)Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter diagnosecluster.

Boomparameternummers: 232, 233, 234, 235, 236, 237.

tel (*cluster_code*: `str`, *subtraject*: `kerapu.lbz.Subtraject.Subtraject`) → `int`
Geeft het aantal malen (d.w.z. 0 of 1) dat een subtraject voldoet aan een diagnoseclustercode.**Parameters**

- **cluster_code** (`str`) – De cluster_code waartegen getest moet worden.
- **subtraject** (`Subtraject`) – Het subtraject.

Return type `int`**kerapu.boom.boom_parameter.DiagnoseCode module****class** `kerapu.boom.boom_parameter.DiagnoseCode.DiagnoseCode`Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter diagnosecode.

Boomparameternummer: 230.

tel (*diagnose_attribuut_code*: `str`, *subtraject*: `kerapu.lbz.Subtraject.Subtraject`) → `int`
Geeft het aantal malen (d.w.z. 0 of 1) dat een subtraject voldoet aan een een (specialismecode, diagnosecode) combinatie.**Parameters**

- **diagnose_attribuut_code** (`str`) – De attribuutcode voor (specialismecode, diagnosecode) combinatie.
- **subtraject** (`Subtraject`) – Het subtraject.

Return type `int`

kerapu.boom.boom_parameter.Geslacht module

class kerapu.boom.boom_parameter.Geslacht.**Geslacht**

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter geslacht.

Boomparameternummer: 230.

tel (*geslacht_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat de patient van een subtraject voldoet aan een geslacht.

Parameters

- **geslacht_code** (*str*) – De geslachtscode waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type int

kerapu.boom.boom_parameter.Leeftijd module

class kerapu.boom.boom_parameter.Leeftijd.**Leeftijd**

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter leeftijd.

Boomparameternummer: 100.

tel (*leeg, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft de leeftijd van de patient van een subtraject.

Parameters

- **leeg** (*None*) – Wordt niet gebruikt.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type int

kerapu.boom.boom_parameter.SpecialismeCluster module

class kerapu.boom.boom_parameter.SpecialismeCluster.**SpecialismeCluster** (*cluster_nummer: int*)

Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter specialismecluster.

Boomparameternummers: 210, 202.

tel (*cluster_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van een subtraject voldoet aan een specialismecluster.

Parameters

- **cluster_code** (*str*) – De clustercode waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type int

kerapu.boom.boom_parameter.SpecialismeCode module**class** kerapu.boom.boom_parameter.SpecialismeCode.**SpecialismeCode**Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter specialismecode.

Boomparameternummer: 200.

tel (*specialisme_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van een subtraject voldoet aan een specialismecode.

Parameters

- **specialisme_code** (*str*) – De specialismecode.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type int**kerapu.boom.boom_parameter.ZorgActiviteitCluster module****class** kerapu.boom.boom_parameter.ZorgActiviteitCluster.**ZorgActiviteitCluster** (*cluster_nummer: int, weeg_factor_nummer: int*)Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter zorgactiviteitcluster.

Boomparameternummers: 301..310, 401..410, 401..510.

tel (*cluster_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subtraject voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type int**kerapu.boom.boom_parameter.ZorgActiviteitCode module****class** kerapu.boom.boom_parameter.ZorgActiviteitCode.**ZorgActiviteitCode** (*weeg_factor_nummer: int*)Basisklassen: *kerapu.boom.boom_parameter.BoomParameter.BoomParameter*

Klasse voor boomparameter zorgactiviteit.

Boomparameternummers: 300, 400, 500.

tel (*zorg_activiteit_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) in een subtraject voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.

- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

kerapu.boom.boom_parameter.ZorgInstellingCode module

class `kerapu.boom.boom_parameter.ZorgInstellingCode.ZorgInstellingCode`

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter zorginstelling.

Boomparameternummer: 110.

tel (*zorg_instelling_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat het zorginstelling van een subtraject voldoet een zorginstelling-code.

Parameters

- **zorg_instelling_code** (*str*) – De AGB-code waaraan de zorginstelling moet voldoen.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

kerapu.boom.boom_parameter.ZorgTypeCode module

class `kerapu.boom.boom_parameter.ZorgTypeCode.ZorgTypeCode`

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter zorgtype.

Boomparameternummer: 210.

tel (*zorg_type_attribuut_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat het zorgtype van een subtraject voldoet aan een (specialismecode, zorgtypecode) combinatie.

Parameters

- **zorg_type_attribuut_code** (*str*) – De attribuutcode voor (specialismecode, zorgtypecode) combinatie.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`

kerapu.boom.boom_parameter.ZorgVraagCluster module

class `kerapu.boom.boom_parameter.ZorgVraagCluster.ZorgVraagCluster` (*cluster_nummer: int*)

Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter zorgvraagcluster.

Boomparameternummers: 221, 222.

tel (*cluster_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voorkomt in een zorgvraagcluster.

Parameters

- **cluster_code** (*str*) – De cluster_code waartegen getest moet worden.
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`**kerapu.boom.boom_parameter.ZorgVraagCode module****class** `kerapu.boom.boom_parameter.ZorgVraagCode.ZorgVraagCode`Basisklassen: `kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Klasse voor boomparameter zorgvraagcode.

Boomparameternummers: 220.

tel (*zorg_vraag_attribuut_code: str, subtraject: kerapu.lbz.Subtraject.Subtraject*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voldoet aan een (specialismecode, zorgvraagcode).

Parameters

- **zorg_vraag_attribuut_code** (*str*) – De attribuutcode voor (specialismecode, zorgvraagcode).
- **subtraject** (*Subtraject*) – Het subtraject.

Return type `int`**Module contents**`kerapu.boom.boom_parameter.create_boom_parameter` (*boom_parameter_nummer: int*) →`kerapu.boom.boom_parameter.BoomParameter.BoomParameter`

Een fabriek met hergebruik voor het maken van boomparameters. Het aanroepen van deze functie met hetzelfde boomparameternummer zal tekens het zelfde object opleveren.

Parameters **boom_parameter_nummer** (*int*) – Het nummer van de boomparameter.**Return type** `BoomParameter`**Submodules****kerapu.boom.AttribuutGroep module****class** `kerapu.boom.AttribuutGroep.AttribuutGroep` (*attribute_groep_id: int, aan-**tal_voorwaarden_voor_true:**int,**koppelingen:**List[kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKo*Basisklassen: `object`

Klasse voor attribuutgroep.

test (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → `bool`

Test of een subtraject voldoet aan de voorwaarden van deze attribuutgroep.

Parameters **subtraject** (*Subtraject*) – Het subtraject.**Return type** `bool`

kerapu.boom.BeslisRegel module

class kerapu.boom.BeslisRegel.**BeslisRegel** (*beslist_regel_id: int, attribuut_groep: kerapu.boom.AttribuutGroep.AttribuutGroep, label_true: str, label_false: str, indicatie_aanspraakbeperking: bool*)

Basisklassen: `object`

Klasse voor beslisregels.

klim (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → `str`
Klimt door de beslisboom een geeft het uiteindelijk gevonden label terug.

Parameters `subtraject` (`Subtraject`) – Het subtraject.

Return type `str`

verrijk (*beslist_regel_true, beslist_regel_false*) → `None`
Verrijkt deze beslisregel met beslisregels voor True and False.

Parameters

- **beslist_regel_true** (`kerapu.boom.BeslisRegel.BeslisRegel`) – De beslisregel voor True.
- **beslist_regel_false** (`kerapu.boom.BeslisRegel.BeslisRegel`) – De beslisregel voor False.

kerapu.boom.ZorgProductGroep module

class kerapu.boom.ZorgProductGroep.**ZorgProductGroep** (*zorg_product_groep_code: str*)
Basisklassen: `object`

Klasse voor zorgproductgroepen.

klim (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → `str`
Bepaalt de zorgproductgroepcode van een subtraject.

Parameters `subtraject` (`subtraject`) – Het subtraject.

versie_toevoegen (*versie: kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie*) → `None`
Voegt een versie toe aan de lijst met versies voor dit zorgproductgroep.

Parameters `versie` (`ZorgProductGroepVersie`) – De toe te voegen versie.

kerapu.boom.ZorgProductGroepVersie module

class kerapu.boom.ZorgProductGroepVersie.**ZorgProductGroepVersie** (*zorg_product_groep_code: str, beslis_regel: kerapu.boom.BeslisRegel.BeslisRegel, begin_datum: str, eind_datum: str*)

Basisklassen: `object`

Klasse voor versies zorgproductgroepversies.

is_actueel (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → bool
 Geeft True als deze versie actueel is voor een subtraject, anders False.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type *bool*

klim (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → str
 Klimt door de beslisboom en geeft het uiteindelijk gevonden label terug.

Parameters **subtraject** (*Subtraject*) – Het subtraject.

Return type *str*

Module contents

kerapu.command package

Submodules

kerapu.command.ShredderCommand module

kerapu.command.TestsetShredderCommand module

Module contents

kerapu.lbz package

Submodules

kerapu.lbz.Diagnose module

class `kerapu.lbz.Diagnose.Diagnose` (*specialisme_code: str, diagnose_code: str*)
 Basisklassen: *object*

Klasse voor diagnoses.

diagnose_attribute_aantal (*diagnose_attribute_code: str, datum: str*) → int
 Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voldoet aan een (specialismecode, diagnosecode) op een peildatum.

Parameters

- **diagnose_attribute_code** (*str*) – De attribuutcode voor de (specialismecode, diagnosecode) combinatie.
- **datum** (*str*) – De peildatum.

Return type *int*

diagnose_cluster_aantal (*cluster_code: str, cluster_nummer: int, datum: str*) → int
 Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voorkomt in een diagnosecodecluster op een peildatum.

Parameters

- **cluster_code** (*str*) – De diagnoseclustercode.

- **cluster_nummer** (*int*) – De clusternummer (0..6).
- **datum** (*str*) – De peildatum.

Return type *int*

static init_static (*folder: str*) → None
Initialiseert alle statistische data.

Parameters folder (*str*) – De folder met alle goupertabellen.

kerapu.lbz.Patient module

class kerapu.lbz.Patient.**Patient** (*geboorte_datum: str, geslacht_code: str*)

Basisklassen: *object*

Klasse voor patiëntenn.

geslacht_code

Geeft het geslacht van deze patiënt.

Return type *str*

leeftijd (*datum: str*) → *int*

Geeft de leeftijd van deze patient op een peildatum.

Parameters datum (*str*) – De peildatum.

Return type *int*

static normaliseer_geslacht_code (*geslacht_code: str*) → *str*

Normaliseert een geslachtscode naar 1 (man), 2 (vrouw) of 9 (anders).

Parameters geslacht_code (*str*) – De geslachtscode.

Return type *str*

kerapu.lbz.Specialisme module

class kerapu.lbz.Specialisme.**Specialisme** (*specialisme_code: str*)

Basisklassen: *object*

Klasse voor specialismen.

static init_static (*folder: str*) → None

Initialiseert alle statistische data.

Parameters folder (*str*) – De folder met alle goupertabellen.

specialisme_aantal (*specialisme_code: str, datum: str*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat dit specialisme voldoet aan een attributecode op een gegeven datum . :param str specialisme_code: De attribuutcode waaraan voldaan moet worden. :param str datum: De datum.

Return type *int*

specialisme_cluster_aantal (*cluster_code: str, cluster_nummer: int, datum: str*) → *int*

Geeft het aantal malen (d.w.z. 0 of 1) dat dit specialisme voldoet aan een clustercode op een gegeven datum.

Parameters

- **cluster_code** (*str*) – De clustercode waaraan voldaan moet worden.
- **cluster_nummer** (*int*) – Het clusternummer.
- **datum** (*str*) – De datum.

Return type `int`

kerapu.lbz.Subtraject module

```
class kerapu.lbz.Subtraject.Subtraject (subtraject_nummer: str, specialisme_code: str, diagnose_code: str, zorg_type_code: str, zorg_vraag_code: str, begin_datum: str, geboorte_datum: str, geslacht_code: str, zorg_instelling_code: str)
```

Basisklassen: `object`

Klasse voor subtrajecten.

add_zorg_activiteit (*zorg_activiteit_code: str, aantal: int*) → None
Voegt een zorgactiviteit toe and dit subtraject.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **aantal** (*int*) – Het aantal malen (of eenheden) dat de zorgactiviteit is uitgevoerd.

begin_datum

Geeft de begindatum van dit subtraject.

Return type `str`

leeftijd

Geeft de leeftijd van de patient van dit subtraject.

Return type `int`

subtraject_nummer

Geeft het subtrajectnummer van dit subtraject.

Return type `str`

telling_behandel_klasse (*behandel_klasse_code: str, weeg_factor_nummer: int*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voorkomt in een behandelklasse.

Parameters

- **behandel_klasse_code** (*str*) – De behandelklassecode waartegen getest moet worden.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).

Return type `int`

telling_diagnose_attribuut (*diagnose_attribuut_code: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat de diagnose van dit subtraject voldoet aan een (specialismecode, diagnosecode) combinatie.

Parameters **diagnose_attribuut_code** (*str*) – De attribuutcode voor de (specialismecode, diagnosecode) combinatie.

Return type `int`

telling_diagnose_cluster (*cluster_code: str, cluster_nummer: int*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat in dit subtraject voldoet aan een diagnoseclustercode.

Parameters

- **cluster_code** (*str*) – De cluster_code waartegen getest moet worden.
- **cluster_nummer** (*int*) – Het clusternummer (1..6).

Return type int

telling_geslacht_code (*geslacht_code: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat de patient van dit subtraject voldoet aan een geslacht.

Parameters **geslacht_code** (*str*) – De geslachtscode waartegen getest moet worden.

Return type int

telling_specialisme (*specialisme_code: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van dit subtraject voldoet aan een specialismecode.

Parameters **specialisme_code** (*str*) – De specialismecode.

Return type int

telling_specialisme_cluster (*cluster_code: str, cluster_nummer: int*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat het uitvoerend specialisme van dit subtraject voldoet aan een specialismecluster.

Parameters

- **cluster_code** (*str*) – De clustercode waartegen getest moet worden.
- **cluster_nummer** (*int*) – Het clusternummer (1..2).

Return type int

telling_zorg_activiteit (*zorg_activiteit_code: str, weeg_factor_nummer: int*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).

Return type int

telling_zorg_activiteit_cluster (*cluster_code: str, cluster_nummer: int, weeg_factor_nummer: int*) → int

Geeft het aantal zorgactiviteiten (met inachtneming van weegfactor) dat in dit subtraject voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **cluster_nummer** (*int*) – Het clusternummer (1..10).
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).

Return type int

telling_zorg_instelling (*agb_code: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorginstelling van dit subtraject voldoet aan AGB-code.

Parameters `agb_code` (*str*) – De AGB-code waaraan de zorginstelling moet voldoen.

Return type `int`

telling_zorg_type_attribuut (*zorg_type_attribuut_code: str*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgtype van dit subtraject voldoet aan een (specialismecode, zorgtypecode) combinatie.

Parameters `zorg_type_attribuut_code` (*str*) – De attribuutcode voor de (specialismecode, zorgtypecode) combinatie.

Return type `int`

telling_zorg_vraag_attribuut (*zorg_vraag_attribuut_code: str*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van dit subtraject voldoet aan een (specialismecode, zorgvraagcode) combinatie.

Parameters `zorg_vraag_attribuut_code` (*str*) – De attribuutcode voor de (specialismecode, zorgvraagcode) combinatie.

Return type `int`

telling_zorg_vraag_cluster (*cluster_code: str, cluster_nummer: int*) → `int`

Geeft het aantal malen (d.w.z. 0 of 1) dat de zorgvraag van een subtraject voorkomt in een zorgvraagcluster.

Parameters

- `cluster_code` (*str*) – De cluster_code waartegen getest moet worden.
- `cluster_nummer` (*int*) – Het clusternummer (1..2).

Return type `int`

zorg_product_code

Geeft de zorgproductcode van dit subtraject.

Return type `str|None`

zorg_product_groep_code

Geeft de zorgproductgroepcode van dit subtraject.

Return type `str|None`

kerapu.lbz.ZorgActiviteit module

class `kerapu.lbz.ZorgActiviteit.ZorgActiviteit` (*zorg_activiteit_code: str, aantal: int*)

Basisklassen: `object`

Klasse voor zorgactiviteiten.

behandel_klasse_aantal (*zorg_product_groep_code: str, behandel_klasse_code: str, weeg_factor_nummer: int, datum: str*) → `int`

Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voorkomt in een behandelklasse op een peildatum.

Parameters

- `zorg_product_groep_code` (*str*) – De zorgproductgroepcode van het subtraject van deze zorgactiviteit.
- `behandel_klasse_code` (*str*) – De behandelklasse.
- `weeg_factor_nummer` (*int*) – Het weegfactornummer (0..2).
- `datum` (*str*) – De peildatum.

Return type `int`

static init_static (*folder: str*) → None
Initialiseert alle statistische data.

Parameters **folder** (*str*) – De folder met alle goupertabellen.

zorg_activiteit_aantal (*zorg_activiteit_code: str, weeg_factor_nummer: int, datum: str*) → int
Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voldoet aan een zorgactiviteitcode.

Parameters

- **zorg_activiteit_code** (*str*) – De zorgactiviteitcode.
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).
- **datum** (*str*) – De peildatum.

Return type `int`

zorg_activiteit_cluster_aantal (*cluster_code: str, cluster_nummer: int, weeg_factor_nummer: int, datum: str*) → int
Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorgactiviteit voorkomt in een zorgactiviteitcluster.

Parameters

- **cluster_code** (*str*) – De zorgactiviteitclustercode.
- **cluster_nummer** (*int*) – het cluster nummer (1..10).
- **weeg_factor_nummer** (*int*) – Het weegfactornummer (0..2).
- **datum** (*str*) – De peildatum.

Return type `int`

kerapu.lbz.ZorgInstelling module

class `kerapu.lbz.ZorgInstelling.ZorgInstelling` (*zorg_instelling_code: str*)
Basisklassen: `object`

Klasse voor zorginstellingen.

zorg_instelling_aantal (*zorg_instelling_code: str*) → int
Geeft het aantal malen (met inachtneming van weegfactor) dat deze zorginstelling voldoet aan een zorginstellingcode.

Parameters **zorg_instelling_code** (*str*) – De AGB-code van de zorginstelling.

Return type `int`

kerapu.lbz.ZorgType module

class `kerapu.lbz.ZorgType.ZorgType` (*specialisme_code: str, zorg_type_code: str*)
Basisklassen: `object`

Klasse voor zorgtypen.

static init_static (*folder: str*) → None
Initialiseert alle statistische data.

Parameters `folder` (*str*) – De folder met alle goupertabellen.

zorg_type_attribute_aantal (*zorg_type_attribute_code: str, datum: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voldoet aan een (specialismecode, zorgtypecode) combinatie op een peildatum.

Parameters

- **zorg_type_attribute_code** (*str*) – De attribuutcode voor (specialismecode, diagnosecode) combinatie.
- **datum** (*str*) – De peildatum.

Return type int

zorg_type_cluster_aantal (*cluster_code: str, cluster_nummer: int, datum: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat deze zorgtype voorkomt in een zorgtypecluster op een peildatum.

Parameters

- **cluster_code** (*str*) – De zorgtypeclustercode.
- **cluster_nummer** (*int*) – Het clusternummer (0..2).
- **datum** (*str*) – De peildatum.

Return type int

kerapu.lbz.ZorgVraag module

class `kerapu.lbz.ZorgVraag.ZorgVraag` (*specialisme_code: str, zorg_vraag_code: str*)

Basisklassen: `object`

Klasse voor zorgvragen.

static **init_static** (*folder: str*) → None

Initialiseert alle statistische data.

Parameters `folder` (*str*) – De folder met alle goupertabellen.

zorg_vraag_attribute_aantal (*zorg_vraag_attribuut_code: str, datum: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) data deze diagnose voldoet aan een (specialismecode, zorgvraagcode) combinatie op een peildatum.

Parameters

- **zorg_vraag_attribuut_code** (*str*) – De attribuutcode voor (specialismecode, diagnosecode) combinatie.
- **datum** (*str*) – De peildatum.

Return type int

zorg_vraag_cluster_aantal (*cluster_code: str, cluster_nummer: int, datum: str*) → int

Geeft het aantal malen (d.w.z. 0 of 1) dat deze zorgvraag voorkomt in een zorgvraagcluster op een peildatum.

Parameters

- **cluster_code** (*str*) – De zorgvraagclustercode.
- **cluster_nummer** (*int*) – Het clusternummer (0..2).
- **datum** (*str*) – De peildatum.

Return type `int`

Module contents

kerapu.shredder package

Submodules

kerapu.shredder.BoomBestandenShredder module

kerapu.shredder.ReferentieShredder module

kerapu.shredder.Shredder module

Module contents

kerapu.style package

Submodules

kerapu.style.KerapuStyle module

Module contents

5.1.2 Submodules

5.1.3 kerapu.Kerapu module

class `kerapu.Kerapu.Kerapu`

Basisklassen: `object`

Een implementatie van de grouper in Python.

bepaal_zorg_product (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → `str`

Bepaalt de zorgproduct van een subtraject.

Parameters `subtraject` (`Subtraject`) – Het subtraject waarvoor de zorgproductcode moet worden bepaalt.

Return type `str`

bepaal_zorg_product_groep (*subtraject: kerapu.lbz.Subtraject.Subtraject*) → `str`

Bepaalt de zorgproductgroep van een subtraject.

Parameters `subtraject` (`Subtraject`) – Het subtraject waarvoor de zorgproductgroep moet worden bepaalt.

Return type `str`

init_static (*folder: str*) → `None`

Initialiseert alle statistische data.

Parameters `folder` (`str`) – De folder met alle goupertabellen.

5.1.4 Module contents

kerapu.**clean_bool** (*x: str*) → bool

kerapu.**clean_code** (*code: str, lengte: int*) → str

Schoont een code van voor- en naloop whitespace en voorziet de code van het juiste aantal voorloop nullen.

Parameters

- **code** (*str*) – De code.
- **lengte** (*int*) – De gewenste lengte van de code.

Return type *str*

kerapu.**clean_date** (*x: str*) → str

kerapu.**clean_int** (*x: Any, leeg: Optional[int] = None*) → Optional[int]

kerapu.**clean_str** (*x: Any*) → Optional[str]

k

kerapu, 29
kerapu.application, 11
kerapu.boom, 21
kerapu.boom.attribuut, 12
kerapu.boom.attribuut.Attribuut, 11
kerapu.boom.attribuut.Attribuut12, 12
kerapu.boom.attribuut.Attribuut21, 12
kerapu.boom.attribuut_groep_koppeling, 14
kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling, 13
kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2, 13
kerapu.boom.AttribuutGroep, 19
kerapu.boom.BeslisRegel, 20
kerapu.boom.boom_parameter, 19
kerapu.boom.boom_parameter.BehandelKlasse, 14
kerapu.boom.boom_parameter.BoomParameter, 15
kerapu.boom.boom_parameter.DiagnoseCluster, 15
kerapu.boom.boom_parameter.DiagnoseCode, 15
kerapu.boom.boom_parameter.Geslacht, 16
kerapu.boom.boom_parameter.Leeftijd, 16
kerapu.boom.boom_parameter.SpecialismeCluster, 16
kerapu.boom.boom_parameter.SpecialismeCode, 17
kerapu.boom.boom_parameter.ZorgActiviteitCluster, 17
kerapu.boom.boom_parameter.ZorgActiviteitCode, 17
kerapu.boom.boom_parameter.ZorgInstellingCode, 18
kerapu.boom.boom_parameter.ZorgTypeCode, 18
kerapu.boom.boom_parameter.ZorgVraagCluster, 18
kerapu.boom.boom_parameter.ZorgVraagCode, 19
kerapu.boom.ZorgProductGroep, 20
kerapu.boom.ZorgProductGroepVersie, 20
kerapu.command, 21
kerapu.Kerapu, 28
kerapu.lbz, 28
kerapu.lbz.Diagnose, 21
kerapu.lbz.Patient, 22
kerapu.lbz.Specialisme, 22
kerapu.lbz.Subtraject, 23
kerapu.lbz.ZorgActiviteit, 25
kerapu.lbz.ZorgInstelling, 26
kerapu.lbz.ZorgType, 26
kerapu.lbz.ZorgVraag, 27
kerapu.shredder, 28
kerapu.style, 28

A

add_zorg_activiteit() (methode van *kerapu.lbz.Subtraject.Subtraject*), 23

Attribuut (klasse in *kerapu.boom.attribuut.Attribuut*), 11

Attribuut12 (klasse in *kerapu.boom.attribuut.Attribuut12*), 12

Attribuut21 (klasse in *kerapu.boom.attribuut.Attribuut21*), 12

AttribuutGroep (klasse in *kerapu.boom.AttribuutGroep*), 19

AttribuutGroepKoppeling (klasse in *kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling*), 13

AttribuutGroepKoppeling2 (klasse in *kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling2*), 13

B

begin_datum (attribuut van *kerapu.lbz.Subtraject.Subtraject*), 23

behandel_klasse_aantal() (methode van *kerapu.lbz.ZorgActiviteit.ZorgActiviteit*), 25

BehandelKlasse (klasse in *kerapu.boom.boom_parameter.BehandelKlasse*), 14

bepaal_zorg_product() (methode van *kerapu.Kerapu.Kerapu*), 28

bepaal_zorg_product_groep() (methode van *kerapu.Kerapu.Kerapu*), 28

BeslisRegel (klasse in *kerapu.boom.BeslisRegel*), 20

BoomParameter (klasse in *kerapu.boom.boom_parameter.BoomParameter*), 15

C

clean_bool() (in module *kerapu*), 29

clean_code() (in module *kerapu*), 29

clean_date() (in module *kerapu*), 29

clean_int() (in module *kerapu*), 29

clean_str() (in module *kerapu*), 29

create_boom_parameter() (in module *kerapu.boom.boom_parameter*), 19

D

Diagnose (klasse in *kerapu.lbz.Diagnose*), 21

diagnose_attribute_aantal() (methode van *kerapu.lbz.Diagnose.Diagnose*), 21

diagnose_cluster_aantal() (methode van *kerapu.lbz.Diagnose.Diagnose*), 21

DiagnoseCluster (klasse in *kerapu.boom.boom_parameter.DiagnoseCluster*), 15

DiagnoseCode (klasse in *kerapu.boom.boom_parameter.DiagnoseCode*), 15

G

Geslacht (klasse in *kerapu.boom.boom_parameter.Geslacht*), 16

geslacht_code (attribuut van *kerapu.lbz.Patient.Patient*), 22

I

init_static() (methode van *kerapu.Kerapu.Kerapu*), 28

init_static() (statische methode van *kerapu.lbz.Diagnose.Diagnose*), 22

init_static() (statische methode van *kerapu.lbz.Specialisme.Specialisme*), 22

init_static() (statische methode van *kerapu.lbz.ZorgActiviteit.ZorgActiviteit*), 26

init_static() (statische methode van *kerapu.lbz.ZorgType.ZorgType*), 26

init_static() (statische methode van *kerapu.lbz.ZorgVraag.ZorgVraag*), 27

is_actueel() (methode van kerapu.boom.ZorgProductGroepVersie (module), 20
kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie), 20

K

Kerapu (klasse in kerapu.Kerapu), 28

kerapu (module), 29

kerapu.application (module), 11

kerapu.boom (module), 21

kerapu.boom.attribuut (module), 12

kerapu.boom.attribuut.Attribuut (module), 11

kerapu.boom.attribuut.Attribuut12 (module), 12

kerapu.boom.attribuut.Attribuut21 (module), 12

kerapu.boom.attribuut_groep_koppeling (module), 14

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling (module), 13

kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppeling (module), 13

kerapu.boom.AttribuutGroep (module), 19

kerapu.boom.BeslisRegel (module), 20

kerapu.boom.boom_parameter (module), 19

kerapu.boom.boom_parameter.BehandelKlasseLeeftijd (module), 14

kerapu.boom.boom_parameter.BoomParameterLeeftijd (module), 15

kerapu.boom.boom_parameter.DiagnoseCluster (module), 15

kerapu.boom.boom_parameter.DiagnoseCode (module), 15

kerapu.boom.boom_parameter.Geslacht (module), 16

kerapu.boom.boom_parameter.Leeftijd (module), 16

kerapu.boom.boom_parameter.SpecialismeCluster (module), 16

kerapu.boom.boom_parameter.SpecialismeCode (module), 17

kerapu.boom.boom_parameter.ZorgActiviteitCluster (module), 17

kerapu.boom.boom_parameter.ZorgActiviteitCode (module), 17

kerapu.boom.boom_parameter.ZorgInstellingCode (module), 18

kerapu.boom.boom_parameter.ZorgTypeCode (module), 18

kerapu.boom.boom_parameter.ZorgVraagCluster (module), 18

kerapu.boom.boom_parameter.ZorgVraagCode (module), 19

kerapu.boom.ZorgProductGroep (module), 20

kerapu.command (module), 21

kerapu.Kerapu (module), 28

kerapu.lbz (module), 28

kerapu.lbz.Diagnose (module), 21

kerapu.lbz.Patient (module), 22

kerapu.lbz.Specialisme (module), 22

kerapu.lbz.Subtraject (module), 23

kerapu.lbz.ZorgActiviteit (module), 25

kerapu.lbz.ZorgInstelling (module), 26

kerapu.lbz.ZorgType (module), 26

kerapu.lbz.ZorgVraag (module), 27

kerapu.shredder (module), 28

kerapu.style (module), 28

klim() (methode van kerapu.boom.BeslisRegel.BeslisRegel), 20

klim() (methode van kerapu.boom.ZorgProductGroep.ZorgProductGroep), 20

klim() (methode van kerapu.boom.ZorgProductGroepVersie.ZorgProductGroepVersie), 21

L

leeftijd (attribuut van kerapu.lbz.Subtraject.Subtraject), 23

leeftijd (klasse in kerapu.boom.boom_parameter.Leeftijd), 16

leeftijd() (methode van kerapu.lbz.Patient.Patient), 22

M

maak_attribuut() (in module kerapu.boom.attribuut), 12

maak_attribuut_groep_koppeling() (in module kerapu.boom.attribuut_groep_koppeling), 14

N

normaliseer_geslacht_code() (statische methode van kerapu.lbz.Patient.Patient), 22

P

Patient (klasse in kerapu.lbz.Patient), 22

S

Specialisme (klasse in kerapu.lbz.Specialisme), 22

specialisme_aantal() (methode van kerapu.lbz.Specialisme.Specialisme), 22

specialisme_cluster_aantal() (methode van kerapu.lbz.Specialisme.Specialisme), 22

SpecialismeCluster	(klasse in kerapu.boom.boom_parameter.SpecialismeCluster),	kerapu.boom.boom_parameter.ZorgTypeCode.ZorgTypeCode),
16	tel ()	18
SpecialismeCode	(klasse in kerapu.boom.boom_parameter.SpecialismeCode),	kerapu.boom.boom_parameter.ZorgVraagCluster.ZorgVraagClus
17	tel ()	18
Subtraject	(klasse in kerapu.lbz.Subtraject),	kerapu.boom.boom_parameter.ZorgVraagCode.ZorgVraagCode),
subtraject_nummer	(attribuut van kerapu.lbz.Subtraject.Subtraject),	19
T		
tel ()	(methode van kerapu.boom.attribuut.Attribuut.Attribuut),	telling_behandel_klasse () (methode van kerapu.lbz.Subtraject.Subtraject),
11	11	23
tel ()	(methode van kerapu.boom.attribuut.Attribuut12.Attribuut12),	telling_diagnose_attribuut () (methode van kerapu.lbz.Subtraject.Subtraject),
12	12	23
tel ()	(methode van kerapu.boom.attribuut.Attribuut21.Attribuut21),	telling_diagnose_cluster () (methode van kerapu.lbz.Subtraject.Subtraject),
12	12	24
tel ()	(methode van kerapu.boom.boom_parameter.BehandelKlasse.BehandelKlasse),	telling_geslacht_code () (methode van kerapu.lbz.Subtraject.Subtraject),
14	14	24
tel ()	(methode van kerapu.boom.boom_parameter.BoomParameter.BoomParameter),	telling_specialisme () (methode van kerapu.lbz.Subtraject.Subtraject),
15	15	24
tel ()	(methode van kerapu.boom.boom_parameter.DiagnoseCluster.DiagnoseCluster),	telling_specialisme_cluster () (methode van kerapu.lbz.Subtraject.Subtraject),
15	15	24
tel ()	(methode van kerapu.boom.boom_parameter.DiagnoseCode.DiagnoseCode),	telling_zorg_activiteit () (methode van kerapu.lbz.Subtraject.Subtraject),
15	15	24
tel ()	(methode van kerapu.boom.boom_parameter.Geslacht.Geslacht),	telling_zorg_activiteit_cluster () (methode van kerapu.lbz.Subtraject.Subtraject),
16	16	24
tel ()	(methode van kerapu.boom.boom_parameter.Leeftijd.Leeftijd),	telling_zorg_instelling () (methode van kerapu.lbz.Subtraject.Subtraject),
16	16	24
tel ()	(methode van kerapu.boom.boom_parameter.SpecialismeCluster.SpecialismeCluster),	telling_zorg_instelling_type_attribuut () (methode van kerapu.lbz.Subtraject.Subtraject),
16	16	25
tel ()	(methode van kerapu.boom.boom_parameter.SpecialismeCode.SpecialismeCode),	telling_zorg_vraag_attribuut () (methode van kerapu.lbz.Subtraject.Subtraject),
17	17	25
tel ()	(methode van kerapu.boom.boom_parameter.ZorgActiviteitCluster.ZorgActiviteitCluster),	telling_zorg_vraag_cluster () (methode van kerapu.lbz.Subtraject.Subtraject),
17	17	25
tel ()	(methode van kerapu.boom.ZorgProductGroep.ZorgProductGroep),	test () (methode van kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppelin
17	17	13
tel ()	(methode van kerapu.boom.ZorgInstellingCode.ZorgInstellingCode),	test () (methode van kerapu.boom.attribuut_groep_koppeling.AttribuutGroepKoppelin
18	18	13
tel ()	(methode van kerapu.boom.ZorgInstellingCode.ZorgInstellingCode),	test () (methode van kerapu.boom.AttribuutGroep.AttribuutGroep),
18	18	19
tel ()	(methode van kerapu.boom.ZorgInstellingCode.ZorgInstellingCode),	verrijk () (methode van kerapu.boom.BeslisRegel.BeslisRegel),
18	18	20
tel ()	(methode van kerapu.boom.ZorgInstellingCode.ZorgInstellingCode),	versie_toevoegen () (methode van kerapu.boom.ZorgProductGroep.ZorgProductGroep),
18	18	20
Z		
tel ()	(methode van kerapu.boom.ZorgInstellingCode.ZorgInstellingCode),	_aantal () (methode van kerapu.lbz.ZorgActiviteit.ZorgActiviteit),
18	18	26

`zorg_activiteit_cluster_aantal()` (methode van `kerapu.lbz.ZorgActiviteit.ZorgActiviteit`), 26

`zorg_instelling_aantal()` (methode van `kerapu.lbz.ZorgInstelling.ZorgInstelling`), 26

`zorg_product_code` (attribuut van `kerapu.lbz.Subtraject.Subtraject`), 25

`zorg_product_groep_code` (attribuut van `kerapu.lbz.Subtraject.Subtraject`), 25

`zorg_type_attribute_aantal()` (methode van `kerapu.lbz.ZorgType.ZorgType`), 27

`zorg_type_cluster_aantal()` (methode van `kerapu.lbz.ZorgType.ZorgType`), 27

`zorg_vraag_attribute_aantal()` (methode van `kerapu.lbz.ZorgVraag.ZorgVraag`), 27

`zorg_vraag_cluster_aantal()` (methode van `kerapu.lbz.ZorgVraag.ZorgVraag`), 27

`ZorgActiviteit` (klasse in `kerapu.lbz.ZorgActiviteit`), 25

`ZorgActiviteitCluster` (klasse in `kerapu.boom.boom_parameter.ZorgActiviteitCluster`), 17

`ZorgActiviteitCode` (klasse in `kerapu.boom.boom_parameter.ZorgActiviteitCode`), 17

`ZorgInstelling` (klasse in `kerapu.lbz.ZorgInstelling`), 26

`ZorgInstellingCode` (klasse in `kerapu.boom.boom_parameter.ZorgInstellingCode`), 18

`ZorgProductGroep` (klasse in `kerapu.boom.ZorgProductGroep`), 20

`ZorgProductGroepVersie` (klasse in `kerapu.boom.ZorgProductGroepVersie`), 20

`ZorgType` (klasse in `kerapu.lbz.ZorgType`), 26

`ZorgTypeCode` (klasse in `kerapu.boom.boom_parameter.ZorgTypeCode`), 18

`ZorgVraag` (klasse in `kerapu.lbz.ZorgVraag`), 27

`ZorgVraagCluster` (klasse in `kerapu.boom.boom_parameter.ZorgVraagCluster`), 18

`ZorgVraagCode` (klasse in `kerapu.boom.boom_parameter.ZorgVraagCode`), 19