

PPL Assignment - Question6

API Documentation

April 10, 2017

Contents

Contents	1
1 Module Boys	2
1.1 Variables	2
1.2 Class Boy	2
1.2.1 Methods	2
2 Module Gifts	3
2.1 Variables	3
2.2 Class Gift	3
2.2.1 Methods	3
3 Module Girls	4
3.1 Variables	4
3.2 Class Girl	4
3.2.1 Methods	4
4 Module boyGeek	5
4.1 Variables	5
4.2 Class boyGeek	5
4.2.1 Methods	5
5 Module boyGenerous	6
5.1 Variables	6
5.2 Class boyGenerous	6
5.2.1 Methods	6
6 Module boyMiser	7
6.1 Variables	7
6.2 Class boyMiser	7
6.2.1 Methods	7
7 Module couple	8
7.1 Variables	8
7.2 Class couple	8
7.2.1 Methods	8
8 Module driver	9

8.1	Functions	9
8.2	Variables	9
9	Module droughtPeriod	10
9.1	Variables	10
9.2	Class CoupleBreaker	10
9.2.1	Methods	10
10	Module generator	11
10.1	Variables	11
10.2	Class randomGenerator	11
10.2.1	Methods	11
11	Module giftEssential	12
11.1	Variables	12
11.2	Class giftEssential	12
11.2.1	Methods	12
12	Module giftLuxury	13
12.1	Variables	13
12.2	Class giftLuxury	13
12.2.1	Methods	13
13	Module giftUtility	14
13.1	Variables	14
13.2	Class giftUtility	14
13.2.1	Methods	14
14	Module girlChoosy	15
14.1	Variables	15
14.2	Class girlChoosy	15
14.2.1	Methods	15
15	Module girlDesperate	16
15.1	Variables	16
15.2	Class girlDesperate	16
15.2.1	Methods	16
16	Module girlNormal	17
16.1	Variables	17
16.2	Class girlNormal	17
16.2.1	Methods	17
17	Module q6	18
17.1	Functions	18
18	Module reader	19
18.1	Variables	19
18.2	Class read	19
18.2.1	Methods	19
19	Script script-CouplesNew.txt	20

20 Script script-girls_csv**21**

1 Module Boys

1.1 Variables

Name	Description
--package--	Value: None

1.2 Class Boy

Known Subclasses: boyGeek.boyGeek, boyGenerous.boyGenerous, boyMiser.boyMiser

The parent class for all types of boys

1.2.1 Methods

`__init__(self, boy)`

constructor which initializes attributes such as :- name = name of the boy attractiveness = attractiveness of the boy intelligence = intelligence of the boy budget = budget of the boy minimumAttrReq = minimum attractiveness required in the girl bType = type of the boy status = relationship status

`gifting(self)`

`happinessCalculator(self)`

2 Module Gifts

2.1 Variables

Name	Description
--package--	Value: None

2.2 Class Gift

Known Subclasses: giftEssential.giftEssential, giftLuxury.giftLuxury, giftUtility.giftUtility

Parent class for all types of Gifts

2.2.1 Methods

<code>--init--(self, gift)</code>
constructor with attributes :- giftType = type of the gift price = price of the gift value = value of the gift

3 Module Girls

3.1 Variables

Name	Description
--package--	Value: None

3.2 Class Girl

Known Subclasses: girlChoosy.girlChoosy, girlDesperate.girlDesperate, girlNormal.girlNormal

”Parent class for all types of girls

3.2.1 Methods

`__init__(self, girl)`

constructor which initializes attributes such as :- name = name of the girl attractiveness = attractiveness of the girl intelligence = intelligence of the girl maintainanceCost = maintainance cost of the girl chosingCri = chosing criterion for the boys gType = type of the girl status = relationship status

`happinessCalculator(self)`

4 Module boyGeek

4.1 Variables

Name	Description
<code>--package--</code>	Value: None

4.2 Class boyGeek

Boys.Boy └─
 boyGeek.boyGeek

Boy class for boyType = 'Geek'

4.2.1 Methods

`__init__(self, boy)`

constructor , calls the parent constructor and initializes other attributes as:- happiness = happiness of the boy amountSpent = amount spent on gifting gfName = name of the girlfriend

Overrides: Boys.Boy.__init__

`happinessCalculator(self, gIntelligence)`

Calculates happiness for Geek boys

Overrides: Boys.Boy.happinessCalculator

`gifting(self, gMaintenanceCost, Gifts, giftBasket)`

Sets up the gift basket for Miser boys

Overrides: Boys.Boy.gifting

5 Module *boyGenerous*

5.1 Variables

Name	Description
<code>--package--</code>	Value: None

5.2 Class *boyGenerous*

Boys.Boy └─ **boyGenerous.boyGenerous**

Boy class for boyType = 'Generous'

5.2.1 Methods

<code>--init--(self, boy)</code>
constructor , calls the parent constructor and initializes other attributes as:- happiness = happiness of the boy amountSpent = amount spent on gifting gfName = name of the girlfriend
Overrides: Boys.Boy. <code>--init--</code>

<code>happinessCalculator(self, gHappiness)</code>
Calculates happiness for Generous boys
Overrides: Boys.Boy. <code>happinessCalculator</code>

<code>gifting(self, gMaintenanceCost, Gifts, giftBasket)</code>
Sets up the gift basket for Generous boys
Overrides: Boys.Boy. <code>gifting</code>

6 Module boyMiser

6.1 Variables

Name	Description
<code>--package--</code>	Value: None

6.2 Class boyMiser

Boys.Boy └─
 boyMiser.boyMiser

Boy class for boyType = 'Miser'

6.2.1 Methods

`__init__(self, boy)`

constructor , calls the parent constructor and initializes other attributes as:- happiness = happiness of the boy amountSpent = amount spent on gifting gfName = name of the girlfriend

Overrides: Boys.Boy.__init__

`gifting(self, gMaintainanceCost, Gifts, giftBasket)`

Sets up the gift basket for Miser boys

Overrides: Boys.Boy.gifting

`happinessCalculator(self)`

Calculates happiness for Miser boys

Overrides: Boys.Boy.happinessCalculator

7 Module couple

7.1 Variables

Name	Description
<code>--package--</code>	Value: None

7.2 Class couple

Couple class with attributes:- bName = Name of the boy gName = Name of the girl happiness = happiness of the couple compatibility = compatibility of the couple

7.2.1 Methods

<code>--init--(self, boy, girl)</code>
constructor

<code>happinessCalcuator(self, boy, girl)</code>
calculates happiness of the couple

<code>compatibilityCalculator(self, boy, girl)</code>
calculates compatibility of the couple

8 Module driver

8.1 Functions

generateRandomInput()
Can be used to generate random Inputs

8.2 Variables

Name	Description
--package--	Value: None

9 Module droughtPeriod

Module containing the class which has fuctions which are responsible for everyday breakup of couples

9.1 Variables

Name	Description
Couples	Value: []
brokenUpGirls	Value: []
Gifts	Value: []
__package__	Value: None
arrBoys	Value: []
arrGirls	Value: []

9.2 Class CoupleBreaker

9.2.1 Methods

makeCouples(*self*)

Makes Couples

breakLeastHappy(*self*, *k*)

Performs break up of Couples with happiness less than k

PatchThemUp(*self*)

Allocates boyfriends to newly broke up girls and form new couples

10 Module generator

10.1 Variables

Name	Description
<code>--package--</code>	Value: None

10.2 Class *randomGenerator*

10.2.1 Methods

<code>__init__(self, totalBoys, totalGirls, totalGifts)</code>
--

<code>generateBoys(self)</code>

<code>generateGirls(self)</code>

<code>generateGifts(self)</code>

11 Module `giftEssential`

11.1 Variables

Name	Description
<code>--package--</code>	Value: None

11.2 Class `giftEssential`

Gifts.Gift —
 giftEssential.giftEssential

Gift Class for gifttype = 'Essential'

11.2.1 Methods

<code>__init__(self, gift)</code>
constructor
Overrides: Gifts.Gift. <code>__init__</code>

12 Module **giftLuxury**

12.1 Variables

Name	Description
<code>--package--</code>	Value: None

12.2 Class **giftLuxury**

Gifts.Gift —
 giftLuxury.giftLuxury

Gift Class for gifttype = 'Luxury'

12.2.1 Methods

<code>__init__(self, gift)</code>
constructor
Overrides: Gifts.Gift.__init__

13 Module **giftUtility**

13.1 Variables

Name	Description
<code>--package--</code>	Value: None

13.2 Class **giftUtility**

Gifts.Gift —
giftUtility.giftUtility

Gift Class for gifttype = 'Utility'

13.2.1 Methods

<code>__init__(self, gift)</code>
constructor
Overrides: Gifts.Gift.__init__

14 Module *girlChoosy*

14.1 Variables

Name	Description
<code>--package--</code>	Value: None

14.2 Class *girlChoosy*

Girls.Girl —
 girlChoosy.girlChoosy

Girl class for `girlType = 'Choosy'`

14.2.1 Methods

<code>__init__(self, girl)</code>
constructor
Overrides: Girls.Girl. <code>__init__</code>

<code>happinessCalculator(self, giftBasket, amount)</code>
Calculates the happiness for girls of type Choosy
Overrides: Girls.Girl. <code>happinessCalculator</code>

15 Module *girlDesperate*

15.1 Variables

Name	Description
<code>--package--</code>	Value: None

15.2 Class *girlDesperate*

Girls.Girl —
 girlDesperate.girlDesperate

Girl class for girlType = 'Choosy'

15.2.1 Methods

<code>__init__(self, girl)</code>
constructor
Overrides: Girls.Girl.__init__

<code>happinessCalculator(self, giftBasket, amount)</code>
Calculates happiness for girls of type Normal
Overrides: Girls.Girl.happinessCalculator

16 Module *girlNormal*

16.1 Variables

Name	Description
<code>--package--</code>	Value: None

16.2 Class *girlNormal*

Girls.Girl —
 girlNormal.girlNormal

Girl class for girlType = 'Choosy'

16.2.1 Methods

<code>__init__(self, girl)</code>
constructor
Overrides: Girls.Girl.__init__

<code>happinessCalculator(self, giftBasket, amount)</code>
Calculates happiness for girls of type Normal
Overrides: Girls.Girl.happinessCalculator

17 Module q6

17.1 Functions

main()
main function for question6 , performs breakups of couples with happiness less than k and patchups of newly broke couples for t days in a year

18 Module reader

18.1 Variables

Name	Description
arrBoys	Value: []
arrGirls	Value: []
Gifts	Value: []
--package--	Value: None

18.2 Class read

18.2.1 Methods

```
--init--(self)
```

```
readcsvfile(self, charType)
```

19 Script script-CouplesNew.txt

20 Script script-girls_csv

Index

boyGeek (*module*), 5
 boyGeek.boyGeek (*class*), 5
boyGenerous (*module*), 6
 boyGenerous.boyGenerous (*class*), 6
boyMiser (*module*), 7
 boyMiser.boyMiser (*class*), 7
Boys (*module*), 2
 Boys.Boy (*class*), 2
 Boys.Boy.__init__ (*method*), 2
 Boys.Boy.gifting (*method*), 2
 Boys.Boy.happinessCalculator (*method*), 2
couple (*module*), 8
 couple.couple (*class*), 8
 couple.couple.__init__ (*method*), 8
 couple.couple.compatibilityCalculator (*method*),
 8
 couple.couple.happinessCalcuator (*method*), 8
driver (*module*), 9
 driver.generateRandomInput (*function*), 9
droughtPeriod (*module*), 10
 droughtPeriod.CoupleBreaker (*class*), 10
 droughtPeriod.CoupleBreaker.breakLeastHappy
 (*method*), 10
 droughtPeriod.CoupleBreaker.makeCouples (*method*),
 10
 droughtPeriod.CoupleBreaker.PatchThemUp (*method*),
 10
generator (*module*), 11
 generator.randomGenerator (*class*), 11
 generator.randomGenerator.__init__ (*method*),
 11
 generator.randomGenerator.generateBoys (*method*),
 11
 generator.randomGenerator.generateGifts (*method*),
 11
 generator.randomGenerator.generateGirls (*method*),
 11
giftEssential (*module*), 12
 giftEssential.giftEssential (*class*), 12
giftLuxury (*module*), 13
 giftLuxury.giftLuxury (*class*), 13
Gifts (*module*), 3
 Gifts.Gift (*class*), 3
 Gifts.Gift.__init__ (*method*), 3
giftUtility (*module*), 14
 giftUtility.giftUtility (*class*), 14
girlChoosy (*module*), 15
 girlChoosy.girlChoosy (*class*), 15
girlDesperate (*module*), 16
 girlDesperate.girlDesperate (*class*), 16
girlNormal (*module*), 17
 girlNormal.girlNormal (*class*), 17
Girls (*module*), 4
 Girls.Girl (*class*), 4
 Girls.Girl.__init__ (*method*), 4
 Girls.Girl.happinessCalculator (*method*), 4
q6 (*module*), 18
 q6.main (*function*), 18
reader (*module*), 19
 reader.read (*class*), 19
 reader.read.__init__ (*method*), 19
 reader.read.readcsvfile (*method*), 19
script-CouplesNew_txt (*script*), 20
script-girls_csv (*script*), 21