PPL Assignment - Question7

API Documentation

April 10, 2017

Contents

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

__init__(*self*, *gift*)

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 allocationMethods

4.1 Variables

Name	Description
Couples	**Value:** `[]`
C	**Value:** {}
Gifts	**Value:** `[]`
__package__	**Value:** `None`
arrBoys	**Value:** `[]`
arrGirls	**Value:** `[]`

4.2 Class allocAndFind

allocates votfriends to all girlfriends and then uses different implementations such as list , sorted list and hashTable to find whethere the particular boy in the list is committed or not

4.2.1 Methods

makeCouples(*self*)

Makes Couples and prints the gifting details of each

FindGf_List(*self*)

Checks whether a boy in the given list of boys has a girlfriend or not by using list

FindGf_SortedList(*self*)

Checks whether a boy in the given list of boys has a girlfriend or not by using sorted list

FindGf_Hash(*self*)

Checks whether a boy in the given list of boys has a girlfriend or not by using HashTable

bSearch(*self*, *list*, *b*)

Binary search to search whether boy is present in the sorted Couples list or not

5 Module boyGeek

5.1 Variables

Name	Description
__package__	**Value:** `None`

5.2 Class boyGeek

Boys.Boy
└ **boyGeek.boyGeek**

Boy class for boyType = 'Geek'

5.2.1 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*, *gMaintainanceCost*, *Gifts*, *giftBasket*)

Sets up the gift basket for Miser boys

Overrides: Boys.Boy.gifting

6 Module boyGenerous

6.1 Variables

Name	Description
__package__	**Value:** `None`

6.2 Class boyGenerous

Boy class for boyType = 'Generous'

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__

happinessCalculator(*self*, *gHappiness*)

Calculates happiness for Generous boys

Overrides: Boys.Boy.happinessCalculator

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

Sets up the gift basket for Generous boys

Overrides: Boys.Boy.gifting

7 Module boyMiser

7.1 Variables

Name	Description
__package__	**Value:** `None`

7.2 Class boyMiser

Boys.Boy
└ **boyMiser.boyMiser**

Boy class for boyType = 'Miser'

7.2.1 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

8 Module couple

8.1 Variables

Name	Description
__package__	**Value:** `None`

8.2 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

8.2.1 Methods

__init__(*self*, *boy*, *girl*)

constructor

happinessCalcuator(*self*, *boy*, *girl*)

calculates happiness of the couple

compatibilityCalculator(*self*, *boy*, *girl*)

calculates compatibility of the couple

9 Module driver

9.1 Functions

generateRandomInput()

Can be used to generate random Inputs

9.2 Variables

Name	Description
__package__	**Value:** `None`

10 Module generator

10.1 Variables

Name	Description
__package__	**Value:** `None`

10.2 Class randomGenerator

10.2.1 Methods

__init__(*self*, *totalBoys*, *totalGirls*, *totalGifts*)

generateBoys(*self*)

generateGirls(*self*)

generateGifts(*self*)

11 Module giftEssential

11.1 Variables

Name	Description
__package__	**Value:** `None`

11.2 Class giftEssential

Gifts.Gift
└
giftEssential.giftEssential

Gift Class for gifttype = 'Essential'

11.2.1 Methods

__init__(*self*, *gift*)

constructor

Overrides: Gifts.Gift.__init__

12 Module giftLuxury

12.1 Variables

Name	Description
__package__	**Value:** `None`

12.2 Class giftLuxury

Gifts.Gift
└
giftLuxury.giftLuxury

Gift Class for gifttype = 'Luxury'

12.2.1 Methods

__init__(*self*, *gift*)

constructor

Overrides: Gifts.Gift.__init__

13 Module giftUtility

13.1 Variables

Name	Description
__package__	**Value:** `None`

13.2 Class giftUtility

Gifts.Gift
└
giftUtility.giftUtility

Gift Class for gifttype = 'Utility'

13.2.1 Methods

__init__(*self*, *gift*)

constructor

Overrides: Gifts.Gift.__init__

14 Module girlChoosy

14.1 Variables

Name	Description
__package__	**Value:** `None`

14.2 Class girlChoosy

Girls.Girl
└
girlChoosy.girlChoosy

Girl class for girlType = 'Choosy'

14.2.1 Methods

__init__(*self*, *girl*)

constructor

Overrides: Girls.Girl.__init__

happinessCalculator(*self*, *giftBasket*, *amount*)

Calculates the happiness for girls of type Choosy

Overrides: Girls.Girl.happinessCalculator

15 Module girlDesperate

15.1 Variables

Name	Description
__package__	**Value:** `None`

15.2 Class girlDesperate

Girls.Girl
└
girlDesperate.girlDesperate

Girl class for girlType = 'Choosy'

15.2.1 Methods

__init__(*self*, *girl*)

constructor

Overrides: Girls.Girl.__init__

happinessCalculator(*self*, *giftBasket*, *amount*)

Calculates happiness for girls of type Normal

Overrides: Girls.Girl.happinessCalculator

16 Module girlNormal

16.1 Variables

Name	Description
__package__	**Value:** `None`

16.2 Class girlNormal

Girls.Girl
└
girlNormal.girlNormal

Girl class for girlType = 'Choosy'

16.2.1 Methods

__init__(*self*, *girl*)

constructor

Overrides: Girls.Girl.__init__

happinessCalculator(*self*, *giftBasket*, *amount*)

Calculates happiness for girls of type Normal

Overrides: Girls.Girl.happinessCalculator

17 Module q7

17.1 Functions

main()

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-girls_csv

20 Script script-logCouples_txt

20.1 Variables

Name	Description
__package__	**Value:** `None`

Index