Question9

API Documentation

April 10, 2017

Contents

1 Package q9

1.1 Modules

- **Couple** *(Section 2, p. 3)*
- **app** *(Section 3, p. 4)*
- **boy** *(Section 4, p. 5)*
- **gift** *(Section 5, p. 7)*
- **girl** *(Section 6, p. 8)*
- **helper** *(Section 7, p. 11)*

1.2 Variables

Name	Description
__package__	**Value:** None

2 Module q9.Couple

2.1 Variables

Name	Description
__package__	**Value:** 'q9'

2.2 Class Couple

2.2.1 Methods

__init__(*self, boy, girl, gift_list, happiness*=0)

:param boy: boy object :param girl: girl object :param gift_list: list of gift object :param happiness: happiness of couple

---

__str__(*self*)

---

allocate_gifts(*self, gift_list*)

Allocates gift from given gift lisr :param gift_list: list of gift objects :return:

---

set_happiness(*self*)

Calculates boy's and girl's happiness individually and adds it. :return:

---

get_compatibility(*self*)

Calculates compatibility of the couple :return:

---

3   Module q9.app

3.1   Functions

main()

3.2   Variables

Name	Description
__package__	**Value:** `'q9'`

4 Module q9.boy

4.1 Variables

Name	Description
__package__	**Value:** None

4.2 Class Boy

Known Subclasses: q9.boy.GeekBoy, q9.boy.GenerousBoy, q9.boy.MiserBoy

Class for boy

4.2.1 Methods

___init___(*self*, *name*, *attractiveness*, *min_attraction*, *intelligence*, *budget*, *single*=1, *happiness*=0)

___str___(*self*)

change_commitment(*self*)

4.3 Class GeekBoy

q9.boy.Boy ⌐
            q9.boy.GeekBoy

4.3.1 Methods

___init___(*self*, *name*, *attractiveness*, *min_attraction*, *intelligence*, *budget*, *single*=1, *happiness*=0)

Overrides: q9.boy.Boy.__init__

get_happiness(*self*, *girlfriend*)

Inherited from q9.boy.Boy(Section 4.2)

__str__(), change_commitment()

4.4   Class MiserBoy

q9.boy.Boy ──┐

　　　q9.boy.MiserBoy

4.4.1   Methods

__init__(*self, name, attractiveness, min_attraction, intelligence, budget, single*=1, *happiness*=0)

Overrides: q9.boy.Boy.__init__

---
get_happiness(*self, girlfriend*)

Inherited from q9.boy.Boy(Section 4.2)

　　__str__(), change_commitment()

4.5   Class GenerousBoy

q9.boy.Boy ──┐

　　　q9.boy.GenerousBoy

4.5.1   Methods

__init__(*self, name, attractiveness, min_attraction, intelligence, budget, single*=1, *happiness*=0)
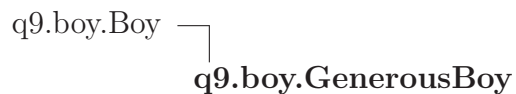
Overrides: q9.boy.Boy.__init__

---
get_happiness(*self, girlfriend*)

Inherited from q9.boy.Boy(Section 4.2)

　　__str__(), change_commitment()

5 Module q9.gift

5.1 Variables

Name	Description
__package__	**Value:** None

5.2 Class EssentialGift

5.2.1 Methods

__init__(*self, name, price, value*)

:param name: name of gift :param price: price of gift :param value: value of the gift

---

5.3 Class LuxuryGift

5.3.1 Methods

__init__(*self, name, price, value, difficulty, rating*)

:param name: name of gift :param price: price of gift :param value: value of gift :param difficulty: difficulty in obtaining the gift :param rating: rating of the gift

---

5.4 Class UtilityGift

5.4.1 Methods

__init__(*self, name, price, value, utility_value, utility_class*)

:param name: name of gift :param price: price of gift :param value: value of gift :param utility_value: utility value of gift :param utility_class: utlility class of gift

---

6 Module q9.girl

6.1 Variables

Name	Description
__package__	**Value:** 'q9'

6.2 Class Girl

Known Subclasses: q9.girl.ChoosyGirl, q9.girl.DesperateGirl, q9.girl.NormalGirl

Class for girl

6.2.1 Methods

__init__(*self*, *name*, *attractiveness*, *intelligence*, *budget*, *single*=1, *happiness*=0)

:param name: Name of the girl :param attractiveness: attractiveness of the girl :param intelligence: intelligence of the girl :param budget: maintenance budget if the girl :param category: Category of the girl, wither choosy, normal or desperate :param single: single status, either 0 or 1 :param happiness: happiness of girl

---
__str__(*self*)

---
change_commitment(*self*, *val*=0)

change commitment status to given value :param val: :return:

6.3 Class ChoosyGirl

q9.girl.Girl ┐
           └
        q9.girl.ChoosyGirl

6.3.1   Methods

__init__(*self, name, attractiveness, intelligence, budget, single*=1, *happiness*=0)

:param name: Name of the girl :param attractiveness: attractiveness of the girl :param intelligence: intelligence of the girl :param budget: maintenance budget if the girl :param category: Category of the girl, wither choosy, normal or desperate :param single: single status, either 0 or 1 :param happiness: happiness of girl

Overrides: q9.girl.Girl.__init__ extit(inherited documentation)

---

get_happiness(*self*)

Inherited from q9.girl.Girl(Section 6.2)

__str__(), change_commitment()

6.4   Class NormalGirl

q9.girl.Girl ┐
        q9.girl.NormalGirl

6.4.1   Methods

__init__(*self, name, attractiveness, intelligence, budget, single*=1, *happiness*=0)

:param name: Name of the girl :param attractiveness: attractiveness of the girl :param intelligence: intelligence of the girl :param budget: maintenance budget if the girl :param category: Category of the girl, wither choosy, normal or desperate :param single: single status, either 0 or 1 :param happiness: happiness of girl

Overrides: q9.girl.Girl.__init__ extit(inherited documentation)

---

get_happiness(*self*)

Inherited from q9.girl.Girl(Section 6.2)

__str__(), change_commitment()

6.5   Class DesperateGirl

q9.girl.Girl ⌐

         q9.girl.DesperateGirl

6.5.1   Methods

__**init**__(*self, name, attractiveness, intelligence, budget, single*=1, *happiness*=0)

:param name: Name of the girl :param attractiveness: attractiveness of the girl :param intelligence: intelligence of the girl :param budget: maintenance budget if the girl :param category: Category of the girl, wither choosy, normal or desperate :param single: single status, either 0 or 1 :param happiness: happiness of girl

Overrides: q9.girl.Girl.__init__ extit(inherited documentation)

---

get_happiness(*self*)

Inherited from q9.girl.Girl(Section 6.2)

    __str__(), change_commitment()

7 Module q9.helper

7.1 Functions

---
form_couple(*boy, girl*)

Takes a boy and girl and forms it couple :param boy: boy object :param girl: girl object :return:

---
get_boy_list()

Autogenerates a random boy list :return:

---
get_girl_list()

autogenerates a random girl list :return:

---
get_gift_list(*filename*)

Extracts gift data from csv file :param filename: filename of the csv file :return: list of gift objects

---

7.2 Variables

Name	Description
__package__	**Value:** 'q9'

8 Script script-log_txt

Index