Mattia Zanini 15/09/2022 5F

Analisi Tecnica

```
# prezzi al metro cubo
GAS_OLD: Final[float] = 0.5
GAS: Final[float] = 1.049988

# prezzi a kWh
ENERGIA_OLD: Final[float] = 0.3
ENERGIA: Final[float] = 0.276

# tasse, prezzo annuo
QVD: Final[int] = 70
ONERI_SISTEMA: Final[int] = 47
SPESE_TRASPORTO: Final[int] = 8*12

POTERE_CALORIFERO: Final[float] = 10.7
```

All'inizio del programma sono state dichiarate delle costanti che rappresentano il prezzo delle materie prime, delle tasse e il consumo di una famiglia media.

Vengono dichiarate le funzioni principali per l'interazione utente/programma:

```
# Confronta i nomi dei vari dispositivi e ritorna 0 per il dispositivo selezionato, altrimenti ritorna il costo del dispositivo
def CostoInstallazione(disp, dispScelto):
    if riscaldamento(dispScelto).name == disp.GetNome():
        return 0
    return disp.GetPrezzoDispositivo()


def SceltaDispositivo():  # Chiede all'utente quale dispositivo tra quelli elencati utilizza
    print("\n\nQuale dispositivo possiedi?:\n" + "1)Caldaia economica\n" + "2)Caldaia a condensazione\n" +
          "3)Stufa\n" + "4)Pompa di calore economica\n" + "5)Pompa di calore di buon livello\n")
    while True:
        val = InputCorretto()
        if (val == 1 or val == 2 or val == 3 or val == 4 or val == 5 or val == 6):
            break
    return val


# Confronta tutte le bollette calcolate e ritorna quella più economica
def DispositivoConveniente(b):
    min = b[0]
    for i in range(0, len(b)):
        if min.GetCostoDecennale() > b[i].GetCostoDecennale():
            min = b[i]
    return min


def InputCorretto():  # Controlla che l'utente inserisca bene i dati (int)
    while True:
        try:
            val = int(input())
            break
        except:
            print("(solo valori interi)")
    return val
```

```python
def FormattaNome(name):  # Formatta il nome del dispositivo per far capire meglio all'utente qual è il risultato ottenuto
    match name:
        case "caldaia_eco":
            return "Caldaia economica"
        case "caldaia":
            return "Caldaia a condensazione"
        case "stufa":
            return "Stufa"
        case "pompa_eco":
            return "Pompa di calore economica"
        case "pompa":
            return "Pompa di calore di buon livello"
```

Vieni richiesto all'utente il quantitativo annuo di kWh e smc con l'ausilio della funzione **InputCorretto()**.

Successivamente vengono creati degli oggetti per ogni dispositivo, gli vengono passati i parametri necessari per ottenere i risultati sul calcolo dell'utilizzo e del costo, tramite le funzioni **CalcUtilizzo()** e **CalcCosto()**. Ogni dispositivo simile, come per esempio **caldaia** e **caldaia_eco**, sono oggetti della stessa classe ed ogni dispositivo deriva da un unica classe **Dispositivo**

```python
class Dispositivo:
    def __init__(self, rendimento, nomeDisp):
        self.rendimento = rendimento
        self.utilizzo = None
        self.prezzoDispositivo = None
        self.nome = nomeDisp

    def CalcUtilizzo(self, consumo, potereCalorifero):
        self.utilizzo = (consumo / (potereCalorifero*self.rendimento))

    def CalcCosto(self, smc_annui, prezzo, tasse):
        self.costo = round(((self.utilizzo + smc_annui)*prezzo)+tasse, 2)

    def GetUtilizzo(self):
        return self.utilizzo

    def GetCosto(self):
        return self.costo

    def GetRendimento(self):
        return self.rendimento

    def GetPrezzoDispositivo(self):
        return self.prezzoDispositivo

    def GetNome(self):
        return self.nome
```

La classe caldaia varia dalla classe padre solo per il metodo **__init__**

```
from dispositivo import Dispositivo


class Caldaia(Dispositivo):
    def __init__(self, rendimento, nomeDisp):
        super().__init__(rendimento, nomeDisp)
        self.prezzoDispositivo = 2000 if rendimento == 1 else 1800
```

Mentre le classi **Stufa** e **Pompa** cambiano i metodi **CalcUtilizzo** e **CalcCosto** e **__init__**

```
from dispositivo import Dispositivo


class Stufa(Dispositivo):
    def __init__(self, rendimento, nomeDisp):
        super().__init__(rendimento, nomeDisp)
        self.prezzoDispositivo = 400

    def CalcUtilizzo(self, consumo, potereCalorifero):
        self.utilizzo = ((consumo * potereCalorifero) / self.rendimento)

    def CalcCosto(self, kwh_annui, prezzo, tasse):
        self.costo = round((self.utilizzo + kwh_annui)*prezzo+tasse, 2)
```

```
from dispositivo import Dispositivo


class Pompa(Dispositivo):
    def __init__(self, rendimento, nomeDisp):
        super().__init__(rendimento, nomeDisp)
        self.prezzoDispositivo = 3000 if rendimento == 3.6 else 1000

    def CalcUtilizzo(self, consumo, potereCalorifero):
        self.utilizzo = ((consumo * potereCalorifero) / self.rendimento)

    def CalcCosto(self, kwh_annui, prezzo, tasse):
        self.costo = round(((self.utilizzo + kwh_annui)
                            * prezzo)+tasse, 2)
```

Dopo la creazione degli oggetti di tutti gli impianti, viene chiesto il dispositivo in possesso dall'utente con tanto di controllo dell'input, tramite le funzioni **SceltaDispositivo** e **InputCorretto**.

Dopo vengono aggiunte alla lista **bollette** vari oggetti **Bolletta** differenti per ogni dispositivo

```
class Bolletta:
    def __init__(self, nome, c, inst=0):
        self.nomeDispositivo = nome
        self.costo = c
        self.costoDecennale = self.CalcCostoDecennale(inst)

    # Questo metodo viene richiamato quando viene usato l'operatore di uguaglianza
    def __eq__(self, other):
        return self.costo == other.GetCosto()

    def CalcCostoDecennale(self, costoInstallazione):
        return round(self.costo*10 + costoInstallazione, 2)

    def GetCosto(self):
        return self.costo

    def GetCostoDecennale(self):
        return self.costoDecennale

    def GetNomeDispositivo(self):
        return self.nomeDispositivo
```

Il metodo **CalcCostoDecennale** calcola il costo di utilizzo del dispositivo per 10 anni sommano infine l'eventuale costo di installazione del dispositivo, arrotondando il suo valore fino a 2 cifre dopo la virgola.

Infine viene mostrato a schermo l'impianto che a seconda dei consumi annui e del dispositivo che l'utente possiede è più conveniente con l'utilizzo della funzione **DispositivoConveniente.**