dati

February 19, 2020

Classi generiche permettono di avere un codice mantenuto

L'approccio di programmazione prevede la soluzione di un problema alla volta, il più atomico possibile, con vantaggi in debug e progettazione generale.

[1]:
```
class Contatto:
    def __init__(self, nome, cognome):
        self.nome = nome
        self.cognome = cognome
        self.attrs = {}

    def set_attr(self, nome_attributo, valore_attributo, unique = False):
        nome_attributo = nome_attributo.lower().strip()
        if unique:
            self.attrs[nome_attributo] = [valore_attributo]
        else:
            data = self.attrs.setdefault(nome_attributo, [])
            data.append(valore_attributo)

    def get_attr(self, nome_attributo):
        nome_attributo = nome_attributo.lower().strip()
        return self.attrs.get(nome_attributo)

    def __str__(self):
        res = [self.nome + " " + self.cognome]

        for k,v in self.attrs.items():
            res.append("%s %s" %(k,v))

        return '\n'.join(res) # func di stringhe aggancia con \n dalla␣
↪                         lista res
```

[2]:
```
c = Contatto("andrea", "prestini")
```

[3]:
```
c.set_attr("cell:", "123456")
```

[4]:
```
c.set_attr("cell:", "999999")
```

[5]:
```
print(str(c))
```

```
andrea prestini
cell: ['123456', '999999']
```

Estensione python per Winzozz = pyw

[6]:
```
class ContattoManager: # gestire i contatti, ricerca, cancellazione
    def __init__(self):
        self._nomi = {}
        self._cognomi = {}
        self._contatti = []

    def add(self, nome, cognome):
        cont = Contatto(nome, cognome)
        self._contatti.append(cont)
        self._nomi.setdefault(nome, []).append(cont)
        self._cognomi.setdefault(cognome, []).append(cont)
        return cont

    def find(self, nome="", cognome=""):
        if nome:
            return self._nomi.get(nome, []) # lista vuota se non presente
        return self._cognome.get(cognome, []) #dict[k] darebbe errore

    def lista(self):
        return self._contatti
```

[7]:
```
cm = ContattoManager()
```

[8]:
```
cont = cm.add("mario", "rossi")
cont = cm.add("mario", "bianchi")
cont = cm.add("sara", "bianchi")
cont = cm.add("paolo", "rossi")
cont.set_attr("tel:", "123456")
```

[9]:
```
for c in cm.lista():
    print(c)
```

```
mario rossi
mario bianchi
sara bianchi
paolo rossi
tel: ['123456']
```

[10]:
```
for x in cm.find("paolo"):
    print(x)
```

```
paolo rossi
tel: ['123456']
```