

# Soluzione Gestione Accessori Aspirapolvere 🖌️

## Descrizione

Questo progetto implementa un sistema per gestire gli accessori specifici per diversi modelli di aspirapolvere utilizzando il design pattern Abstract Factory. Ogni modello di aspirapolvere è dotato di tre accessori dedicati: spazzola, filtro e sacchetto.

## Struttura del Progetto

```
aspirapolvere_project/
├── factories/
│   ├── abc456_factory.py      # Factory concreta per il modello ABC456
│   ├── accessorio_factory.py # Interfaccia astratta delle factory
│   └── xyz123_factory.py      # Factory concreta per il modello XYZ123
├── models/
│   ├── aspirapolvere.py      # Classe principale Aspirapolvere
│   ├── accessori/
│   │   ├── filtro.py         # Classi per i filtri
│   │   ├── sacchetto.py      # Classi per i sacchetti
│   │   ├── spazzola.py       # Classi per le spazzole
│   │   └── __init__.py       # File package accessori
│   └── __init__.py          # File package models
└── main.py                  # Punto di ingresso dell'applicazione
```

## Requisiti di Sistema

- Python 3.7 o versioni successive
- Ambiente di sviluppo con pip configurato

## Avvio del Progetto

1. Aprire il terminale
2. Navigare alla directory del progetto
3. Eseguire:

```
python main.py
```



## Output di Esempio

Aspirapolvere: SuperClean XYZ123

- ✓ Spazzola compatibile con il modello XYZ123
- ✓ Filtro compatibile con il modello XYZ123
- ✓ Sacchetto compatibile con il modello XYZ123

Aspirapolvere: PowerVac ABC456

- ✓ Spazzola compatibile con il modello ABC456
- ✓ Filtro compatibile con il modello ABC456
- ✓ Sacchetto compatibile con il modello ABC456



## Design Pattern

- Abstract Factory Pattern per la creazione di famiglie di accessori compatibili



## Note Aggiuntive

Il progetto è strutturato per essere facilmente estendibile con nuovi modelli di aspirapolvere e relativi accessori.

---

*Sviluppato come esempio di implementazione del pattern Abstract Factory*