

```

class DataBlock {
private:
    int size;
    double* values;
};

class BaseProcessor {
protected:
    DataBlock data;
    virtual void preprocess() = 0;
public:
    virtual ~BaseProcessor() {}
    virtual void process() = 0;
};

class StatisticalProcessor: virtual public BaseProcessor {
protected:
    double mean;
    double variance;
public:
    StatisticalProcessor();
    void preprocess() override;
    virtual double getMean() const;
};

class RegressionProcessor: virtual public BaseProcessor {
protected:
    double slope;
    double intercept;
public:
    RegressionProcessor();
    void preprocess() override;
    virtual double predict(double x) const;
};

class AdvancedProcessor: public StatisticalProcessor, public
RegressionProcessor {
private:
    bool normalized;
    DataBlock* extraData;
public:
    AdvancedProcessor(bool norm = false);
    AdvancedProcessor(const AdvancedProcessor& other);
    // Da implementare:
    // (1) ~AdvancedProcessor()
    // (2) AdvancedProcessor& operator=(const AdvancedProcessor& other)
    // (3) void process() override

```

```
// (4) AdvancedProcessor* clone() const  
};
```

Si considerino le definizioni sopra.

1. Ridefinire il distruttore della classe `DataBlock` in modo che gestisca correttamente la memoria allocata.
2. Ridefinire il costruttore di copia profonda della classe `DataBlock`.
3. Ridefinire l'operatore di assegnazione della classe `DataBlock` in modo che il suo comportamento coincida con quello dell'assegnazione standard.
4. Implementare il distruttore della classe `AdvancedProcessor`.
5. Implementare l'operatore di assegnazione della classe `AdvancedProcessor` in modo che faccia un'assegnazione profonda.
6. Implementare il metodo `process()` della classe `AdvancedProcessor` in modo che chiami entrambe le implementazioni di `preprocess()` delle classi base, calcoli `slope` e `intercept` basandosi sui dati, e imposti `normalized` a `true` se la varianza è minore di 1.0.
7. Implementare il metodo di clonazione della classe `AdvancedProcessor`.

```
// (1) Ridefinire il distruttore di DataBlock  
...  
  
// (2) Ridefinire il costruttore di copia di DataBlock  
...  
  
// (3) Ridefinire l'operatore di assegnazione di DataBlock  
...  
  
// (4) Implementare il distruttore di AdvancedProcessor  
...  
  
// (5) Implementare l'operatore di assegnazione di AdvancedProcessor  
...  
  
// (6) Implementare il metodo process() di AdvancedProcessor  
...  
  
// (7) Implementare il metodo di clonazione di AdvancedProcessor  
...
```