# PolyInterface (Interface)

- power: int
- coefficient: double
- + plus: PolyInterforce
- + multiply: PolyInterface
- + translorte: PolyInterface
- + getDegree(): int
- + get Coefficient (int): double
- + get leading Coefficient (): double
- + get { Intercept(): obuble
- + get String(): String
- + evaluateAt (double): double
- + is Root(Mt): Boolean
- + isEquallo (PolyInterface): boolean
- + &Strny: Strny

## Constant

- + getDegree(): int
- + get Coefficient (int): double
- + Set Learning Coefficient () double
- + get { Intercept(): obuble
- + get String(): String
- + evaluateAt (double): double
- + is Root(mt): Boolean
- + isEqualTo (PolyInterface): booleon
- + & String: String

### Line

#### -Constant constant

- + getDegree(): int
- + get Coefficient (int): double
- + getleading Coefficient(): double
- + gerYIntercept(): okuble
- + get Fing(): String
- + evaluateAt (double): double
- + is Root(mt): Boolean
- + isEqualTo (PolyInterface): boolean
- + &String: String
- + plus; PolyInterforce
- + multiply: PolyInterface
- + translate: PolyInterfore

### Quadratic

- coefficient 1: double
- coefficient 2: double
- + getDegree(): int
- + get Goefficient (Int): double
- + getleading Coefficient(). double
- t get (Intercept (): obuble
- + get Fing(): String
- + evaluateAt (double): double
- t is Root(mt): Boolean
- t isEquatTo (PolyInterface); boolean
- + &Stry: Stry
- + plus: PolyInterforce
- + multiply: PolyInterface
- + translate: PolyInterfore