

Shape

```
-shapeID: int
-qShapeType: shapeType
-shapeDimensions: vector<do
-qPenColor: penColor
-penWidth: int
-qPenStyle: penStyle
-qPenCapStyle: penCapStyle
-qPenJoinStyle: penJoinStyl
-qBrushColor: brushColor
-qBrushStyle: brushStyle
-textString: std::string
-qTextColor: textColor
-qTextAlignment: textAlignm
-textPointSize: int
-textFontFamily: std::strin
-qTextFontStyle: textFontSt
-qTextFontWeight: textFontW
```

```
+shape()
+getPenCapStyle(): penCapSt
+getPenColor(): penColor
+getPenWidth(): int
+getPenStyle(): penStyle
+getPenJoinStyle(): penJoin
```

```
uble>* = nullptr
```

e

ient

g

yle

eight

yle

.Style

```
-size_v: i  
-elem: Typ  
-space: in
```

```
+vector()  
+vector(s:  
+vector(sr  
+~vector()  
+operator[  
+operator[  
+size(): c  
+capacity(  
+resize(ne  
+push_back  
+reserve(n  
+begin():  
+begin():  
+end(): It  
+end(): co  
+insert(p:  
+erase(p:I
```

Vector

nt
e*
t

int)
c:const vector&)

](n:int): Type&
](n:int): const Type&
onst int
) : const int
wsiz: int): void
(d:Type): void
ewalloc: int): void
Iterator
const const_iterator
erator
nst const_iterator
Iterator, val: const double&): Iterator
erator): Iterator

Shape_Par

```
-sList: vector<shape>* = r
-temp: shape* = nullptr
-fin: std::ifstream
-fout: std::ofstream
-trash: std::string
-shapeID: int
-sType: shapeType
-shapeDimensions: vector<c
-pColor: penColor
-pWidth: int
-pStyle: penStyle
-pCapStyle: penCapStyle
-pJoinStyle: penJoinStyle
-bColor: brushColor
-bStyle: brushStyle
-tString: std::string
-tColor: textColor
-tAlignment: textAlignment
-tPointSize: int
-tFontFamily: std::string
-tFontStyle: textFontStyle
-tFontWeight: textFontWeig
+shapeParser()
```

ser

```
new vector<shape>
```

```
double>* = nullptr
```

```
;
```

```
>
```

```
ght
```

```

+getBrushColor(): brushColor
+getBrushStyle(): brushStyle
+getShapeId(): int
+getShapeDimensions(): vect
+setPenCapStyle(style:penCapStyle): void
+setPenColor(color:penColor): void
+setPenWidth(width:int): void
+setPenStyle(style:penStyle): void
+setPenJoinStyle(style:penJoinStyle): void
+setBrushColor(color:brushColor): void
+setBrushStyle(style:brushStyle): void
+setShapeId(iD:int): void
+setShapeDimensions(dim:vector): void
+getArea(): double
+getPerimeter(): double
+~shape()

```



Line	Polygon
<pre> +line() +getPerimeter(): double +getArea(): double </pre>	<pre> +polyline() +getPerimeter(): double +getArea(): double </pre>

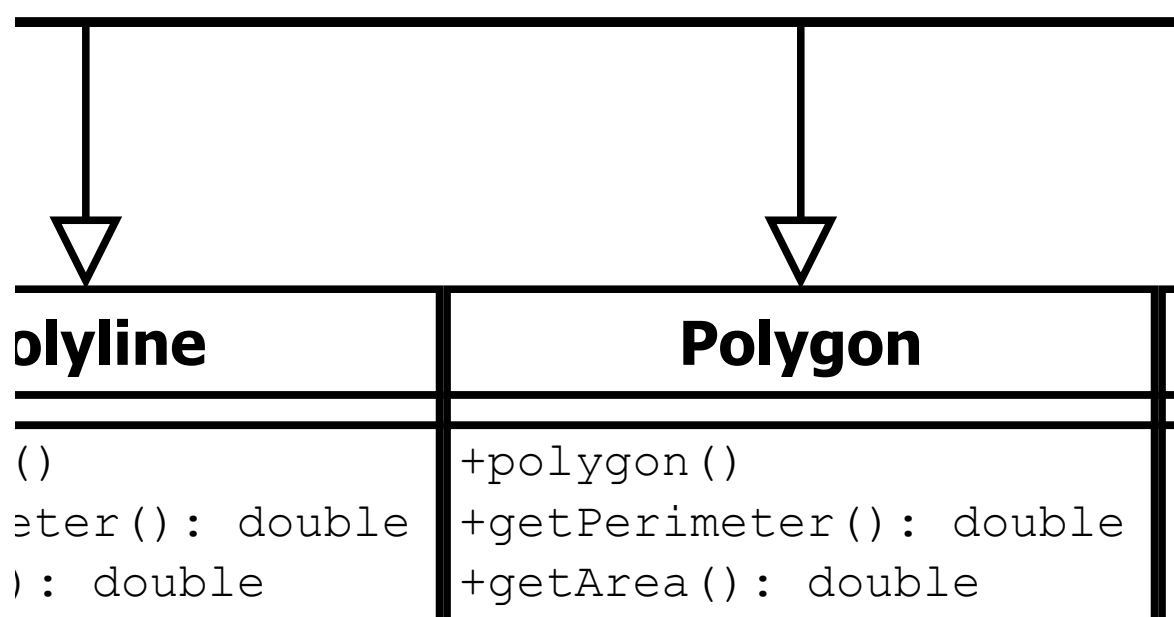
```

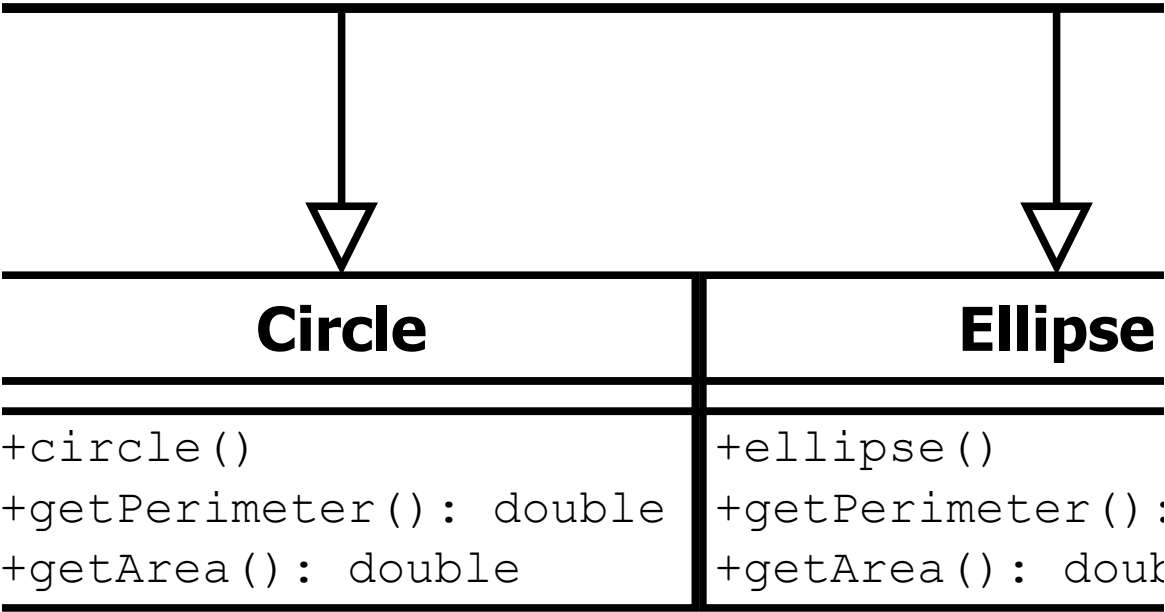
r
e

or<double>*
pStyle): void
): void
id
): void
oinStyle): void
olor): void
tyle): void

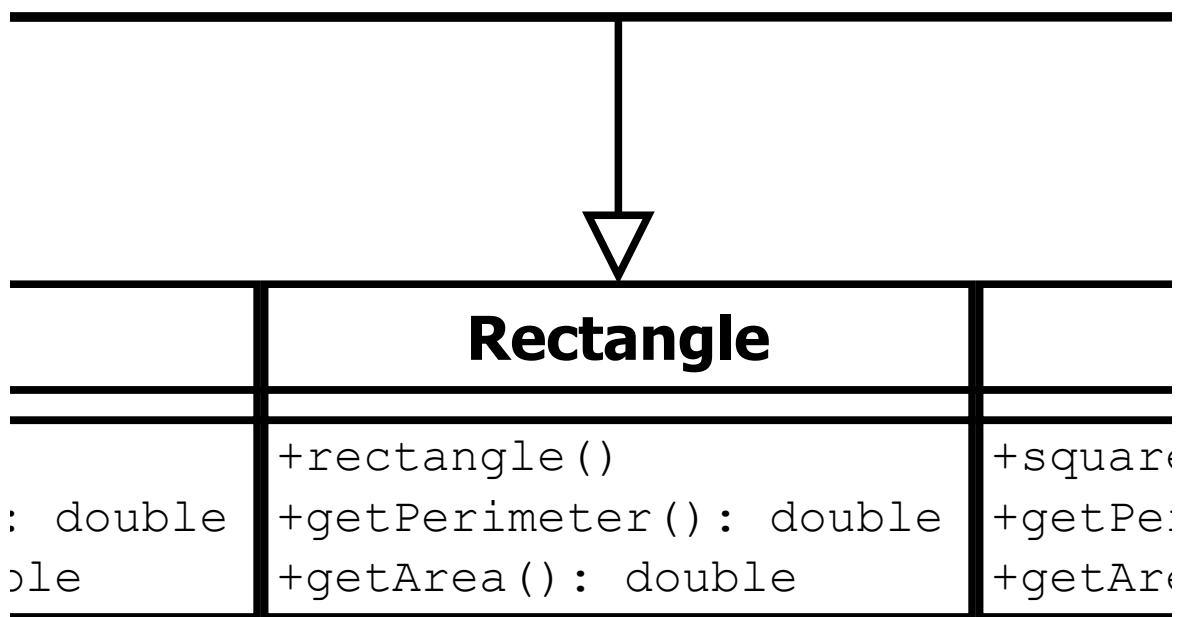
tor<double>*): void

```






```
+initializeVector(): vecto
```



`or<shape>*`

