```
<<class>>
MainWindow
[QT Main Window
Object]
```

I

```
                              <
                              |
-size_v: int
-vectorCapacity
-elem: T
+<<constructor
+<<constructor
+<<constructor
+size(): int
+capacity(): i
+reverse(new a
+resize(newsize
+push_back(newl
+begin(): iter
+end(): const_
+insert(p, &va
+erase(p): ite
```
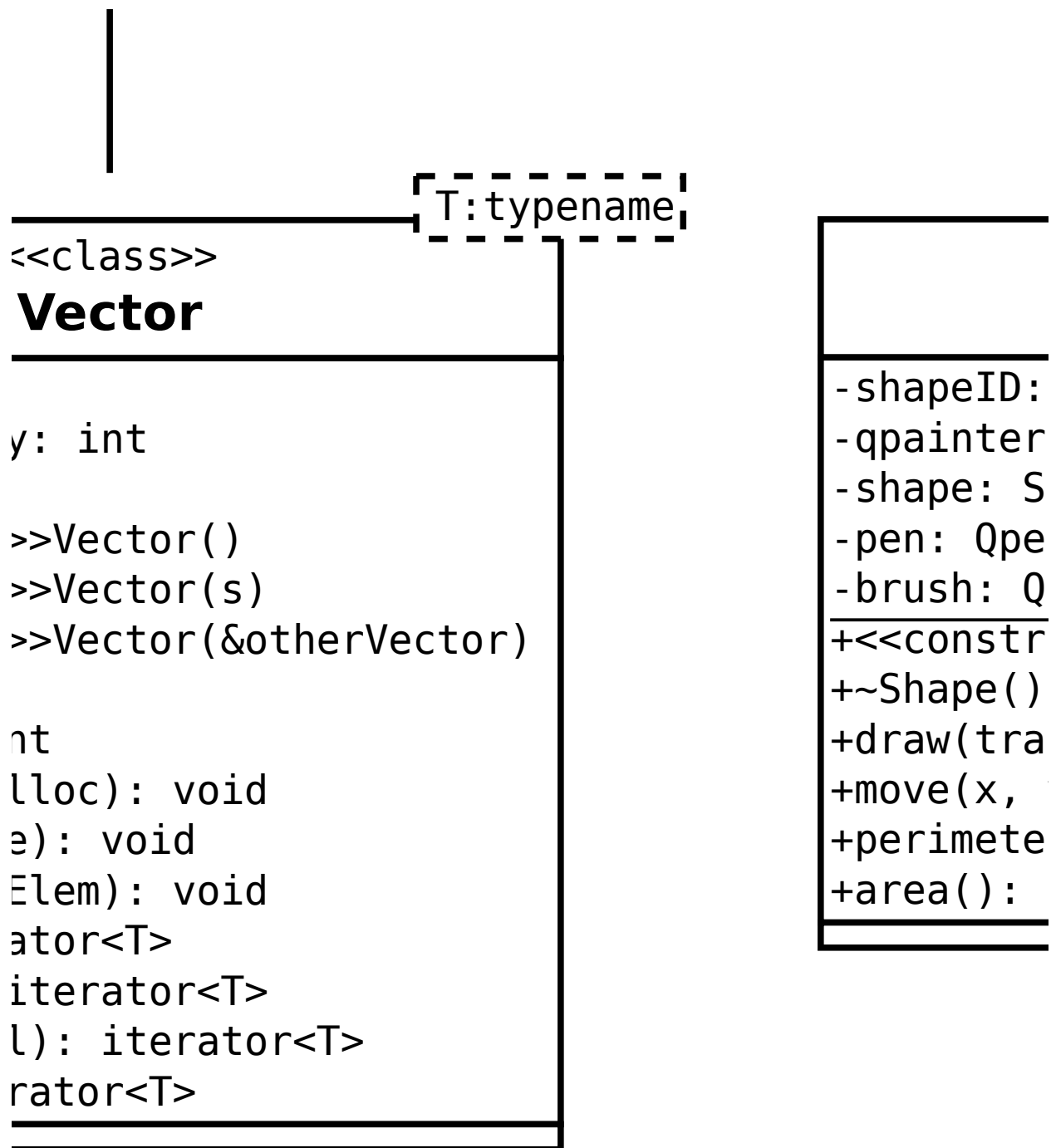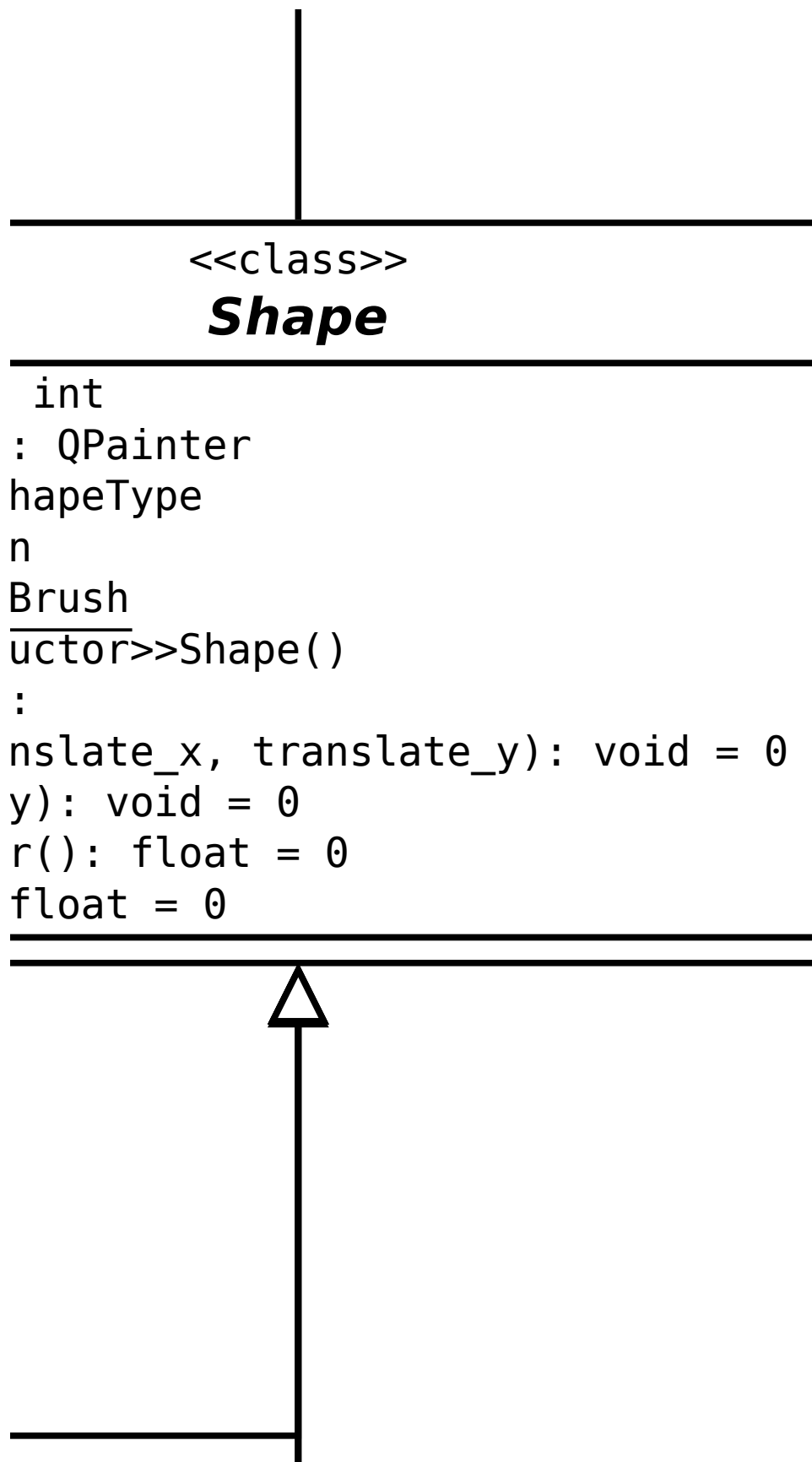
```
                              ┌ ─ ─ ─ ─ ─ ─ ─ ┐
                               T:typename
                              └ ─ ─ ─ ─ ─ ─ ─ ┘
<<class>>
**Vector**

y: int

>>Vector()
>>Vector(s)
>>Vector(&otherVector)

nt
lloc): void
e): void
Elem): void
ator<T>
iterator<T>
l): iterator<T>
rator<T>
```

```
-shapeID:
-qpainter
-shape: S
-pen: Qpe
-brush: Q
+<<constr
+~Shape()
+draw(tra
+move(x,
+perimete
+area():
```

```
<<class>>
Shape
```

int
: QPainter
hapeType
n
Brush
uctor>>Shape()
:
nslate_x, translate_y): void = 0
y): void = 0
r(): float = 0
float = 0

```
-rect: QRect
+<constructor>>Ellipse (QPaint *device
+~Ellipse()
+isCircle()const:bool
+set_rect(const QRect &rect):void
+draw (const int translate_x = 0, cons
+perimeter(): float
```

, int id, QColor xPenColor, qreal xPenWidth

t int translate_y = 0)void

```
point: Vector<QPoint>
-<<constructor>>Polyline (QPaintDevice *devi
-~polyline()
-set_point(const QPoint_begin, const QPoint&
-draw(const int Translate_x = 0, const int t
-area(): float
```

E

```
, Qt::PenStyle xPenStyle, Qt::PenCapStyle xl
```

Poly

ce, int id, QColor xPenColor, qreal xPenWid

point_end):void
ranslate_y = 0): void

llipse

PenCapStyle, Qt::PenJoinStyle xPenStyle, QC

Line

`th, Qt::PenStyle xPenStyle, Qt::PenCapStyle`

`olor xBrushColor, Qt::BrushStyle XBrushStyle`

xPenCapStyle, Qt::PenJoinStyle xPenStyle)

e, xTopLeftX, XTopLefty, xWidth, xHeight)

```
-point: Vector<QPoint>
+<<constructor>>Polygon (QPaintl
+~Polygon()
+set_point(const QPoint_begin, 
+draw(const int Translate_x = 0
+perimeter(): float
+area(): float
```

```
-point: Vector<QPoint>
+<<constructor>>Line (QPaintDev
+~Line()
+set_point(const QPoint_begin, 
+drow(const int Translate_x = 0
+area(): float
```
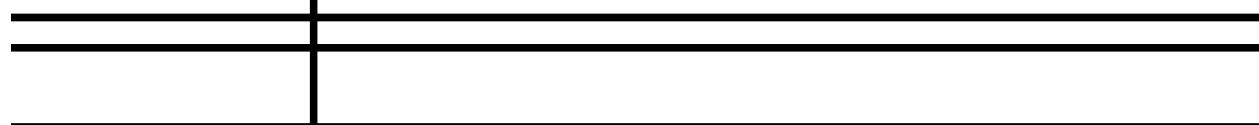
Device *device, int id, QColor xPenColor, qr

const QPoint& point_end):void
, const int translate_y = 0): void

ice *device, int id, QColor xPenColor, qreal

const QPoint& point_end):void
, const int translate_y = 0): void
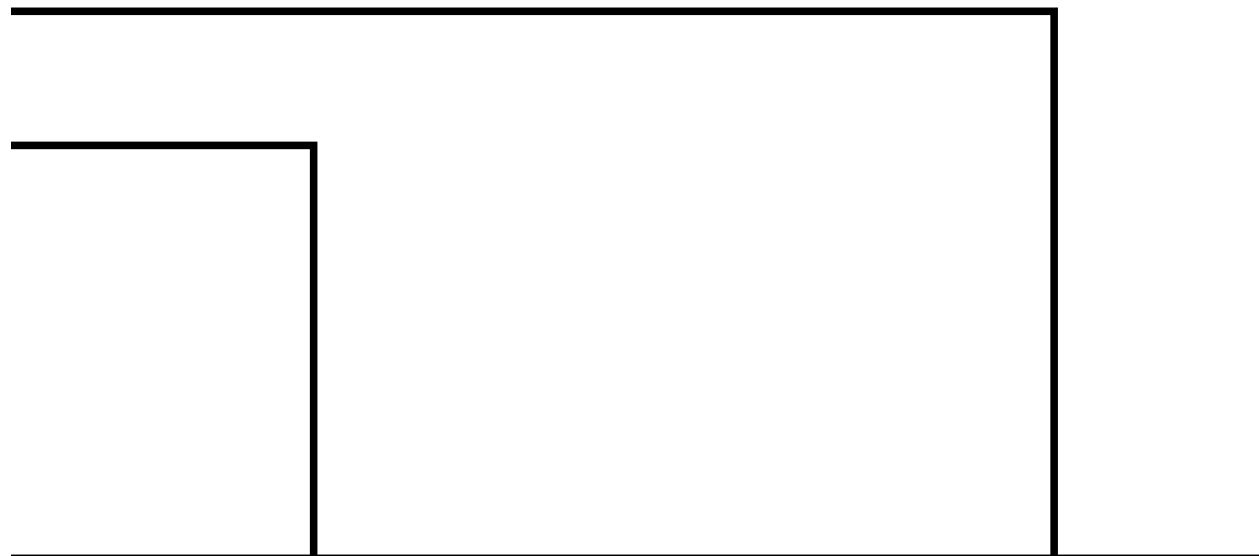
Polygon

real xPenWidth, Qt::PenStyle xPenStyle, Qt::

Line

l xPenWidth, Qt::PenStyle xPenStyle, Qt::Pen

PenCapStyle xPenCapStyle, Qt::PenJoinStyle

nCapStyle xPenCapStyle, Qt::PenJoinStyle xPe

xPenStyle, QColor xBrushColor, Qt::BrushSty

nStyle)

le xBrushStyle)

```
+area(): float
```

```
                                                          ═══════════════
                                                          ═══════════════


  ┌──────────────────────────────────────────────────────┄
  │                                                       
  │                                                       
  ├──────────────────────────────────────────────────────┄
  │ -upperleft: QPoint
  │ -lowerright: QPoint
  │ -rect: QRect
  │ +<<constructor>>Rectangle (QPaintDevice *
  │ +~Rectangle()
  │ +isSquare() const: bool
  │ +set_rect(const QRect& rect): void
  │ +draw(const int Translate_x = 0, const in
  │ +perimeter(): float
  │ +area(): float
  └──────────────────────────────────────────────────────┄
```

device, int id, QColor xPenColor, qreal xPe

t translate_y = 0): void

nWidth, Qt::PenStyle xPenStyle, Qt::PenCapSt

Rectangle

tyle xPenCapStyle, Qt::PenJoinStyle xPenStyl

e, QColor xBrushColor, Qt::BrushStyle xBrus

hStyle, int xTopLeftX, int xTopLefty, int x

Width, int xHeight)