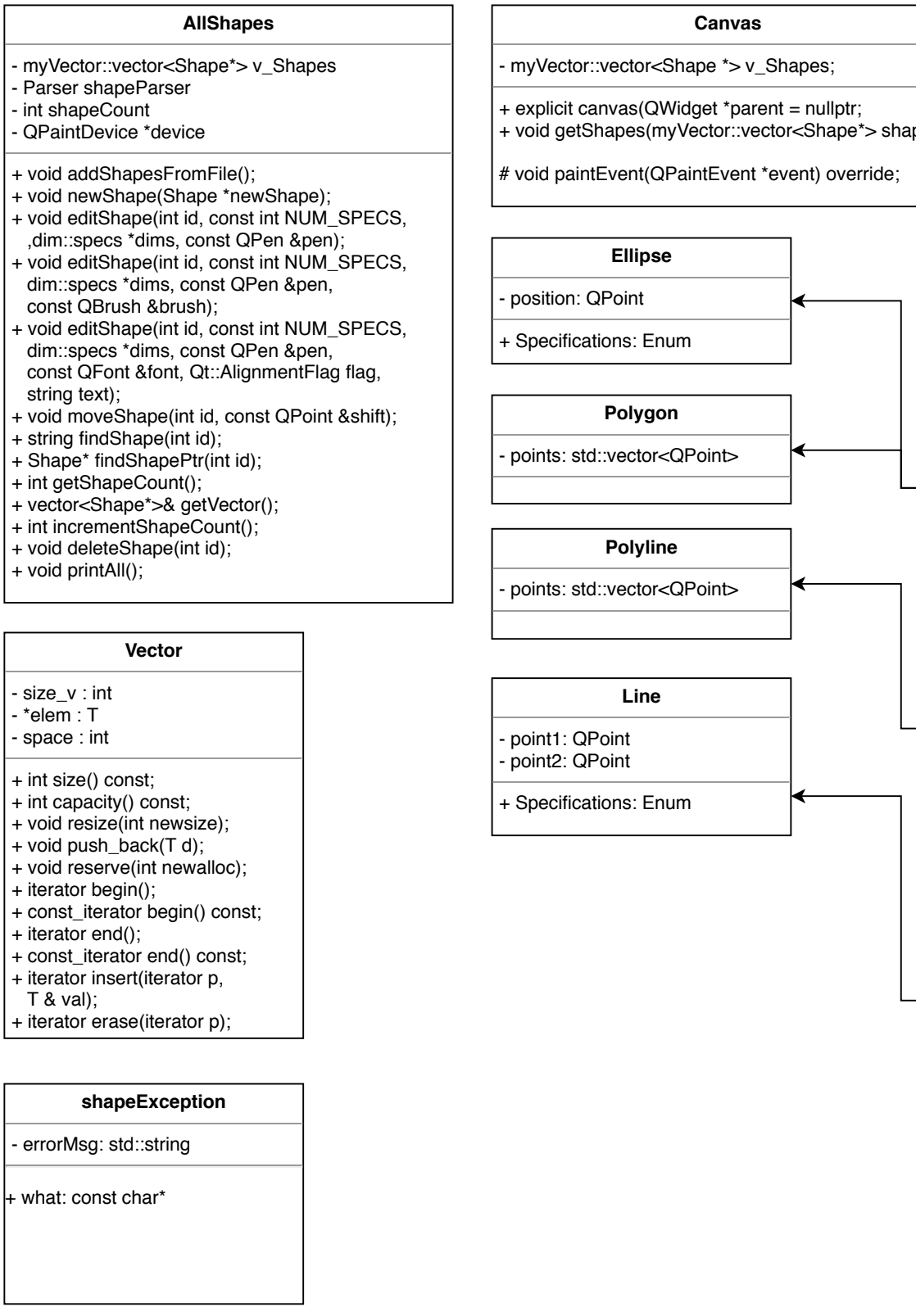
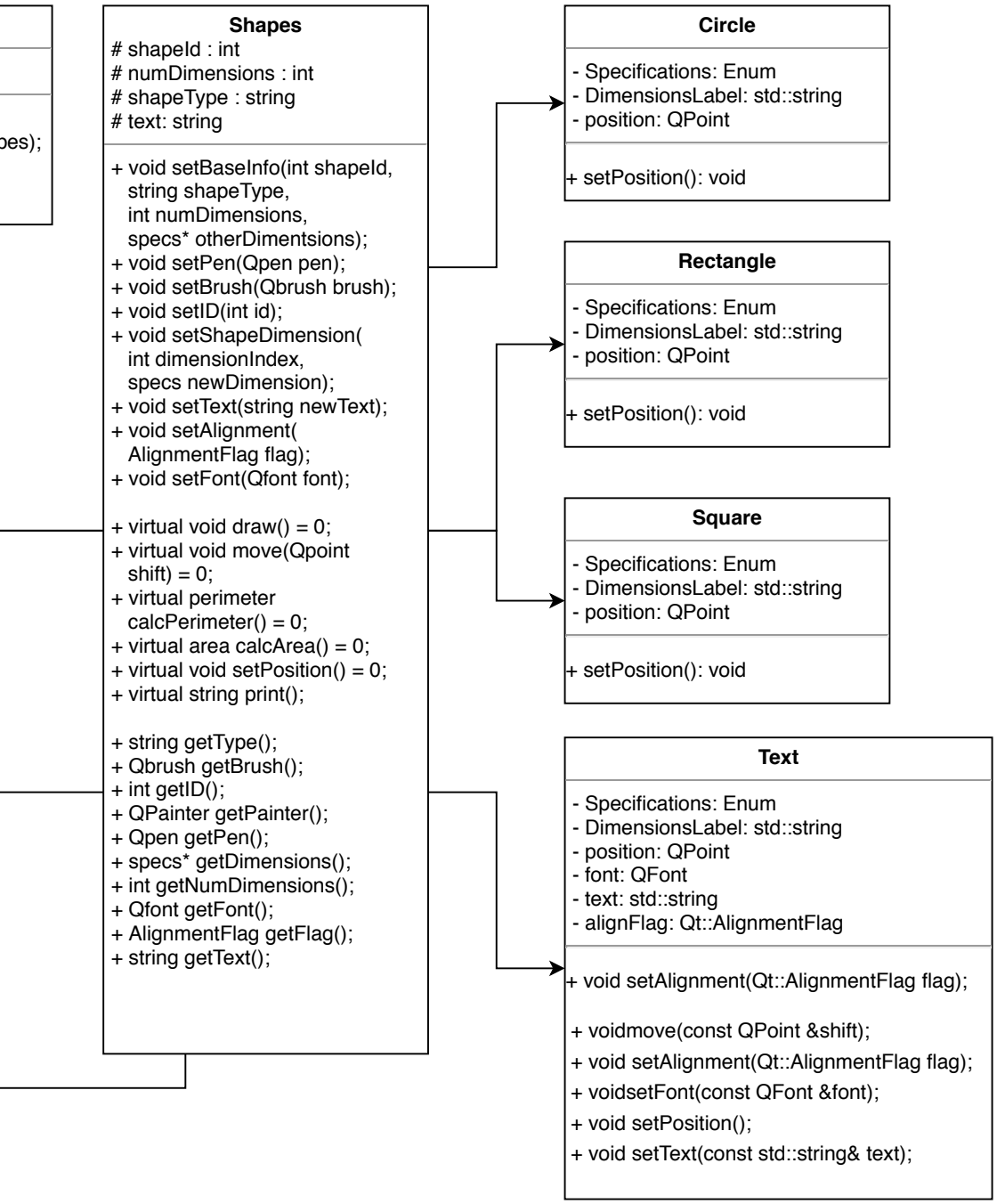

AllShapes
- myVector::vector<Shape*> v_Shapes
- Parser shapeParser
- int shapeCount
- QPaintDevice *device
+ void addShapesFromFile();
+ void newShape(Shape *newShape);
+ void editShape(int id, const int NUM_SPECS,
,dim::specs *dims, const QPen &pen);
+ void editShape(int id, const int NUM_SPECS,
dim::specs *dims, const QPen &pen,
const QBrush &brush);
+ void editShape(int id, const int NUM_SPECS,
dim::specs *dims, const QPen &pen,
const QFont &font, Qt::AlignmentFlag flag,
string text);
+ void moveShape(int id, const QPoint &shift);
+ string findShape(int id);
+ Shape* findShapePtr(int id);
+ int getShapeCount();
+ vector<Shape*>& getVector();
+ int incrementShapeCount();
+ void deleteShape(int id);
+ void printAll();
Canvas
- myVector::vector<Shape *> v_Shapes;
+ explicit canvas(QWidget *parent = nullptr;
+ void getShapes(myVector::vector<Shape*> shap
# void paintEvent(QPaintEvent *event) override;
Ellipse
- position: QPoint
+ Specifications: Enum
Polygon
- points: std::vector<QPoint>
Polyline
- points: std::vector<QPoint>
Line
- point1: QPoint
- point2: QPoint
+ Specifications: Enum
Vector
- size_v : int
- *elem : T
- space : int
+ int size() const;
+ int capacity() const;
+ void resize(int newsize);
+ void push_back(T d);
+ void reserve(int newalloc);
+ iterator begin();
+ const_iterator begin() const;
+ iterator end();
+ const_iterator end() const;
+ iterator insert(iterator p,
T & val);
+ iterator erase(iterator p);
shapeException
- errorMsg: std::string
+ what: const char*


Shapes
shapeId : int
numDimensions : int
shapeType : string
text: string
+ void setBaseInfo(int shapeId, string shapeType, int numDimensions, specs* otherDimentsions);
+ void setPen(Qpen pen);
+ void setBrush(Qbrush brush);
+ void setID(int id);
+ void setShapeDimension(int dimensionIndex, specs newDimension);
+ void setText(string newText);
+ void setAlignment(AlignmentFlag flag);
+ void setFont(Qfont font);
+ virtual void draw() = 0;
+ virtual void move(Qpoint shift) = 0;
+ virtual perimeter calcPerimeter() = 0;
+ virtual area calcArea() = 0;
+ virtual void setPosition() = 0;
+ virtual string print();
+ string getType();
+ Qbrush getBrush();
+ int getID();
+ QPainter getPainter();
+ Qpen getPen();
+ specs* getDimensions();
+ int getNumDimensions();
+ Qfont getFont();
+ AlignmentFlag getFlag();
+ string getText();
Circle
- Specifications: Enum
- DimensionsLabel: std::string
- position: QPoint
+ setPosition(): void
Rectangle
- Specifications: Enum
- DimensionsLabel: std::string
- position: QPoint
+ setPosition(): void
Square
- Specifications: Enum
- DimensionsLabel: std::string
- position: QPoint
+ setPosition(): void
Text
- Specifications: Enum
- DimensionsLabel: std::string
- position: QPoint
- font: QFont
- text: std::string
- alignFlag: Qt::AlignmentFlag
+ void setAlignment(Qt::AlignmentFlag flag);
+ voidmove(const QPoint &shift);
+ void setAlignment(Qt::AlignmentFlag flag);
+ voidsetFont(const QFont &font);
+ void setPosition();
+ void setText(const std::string& text);