



Layout



핵심 정리

예제 Project

Project Name	LAYOUT_1
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp layout1.h layout2.h layout3.h layout4.h



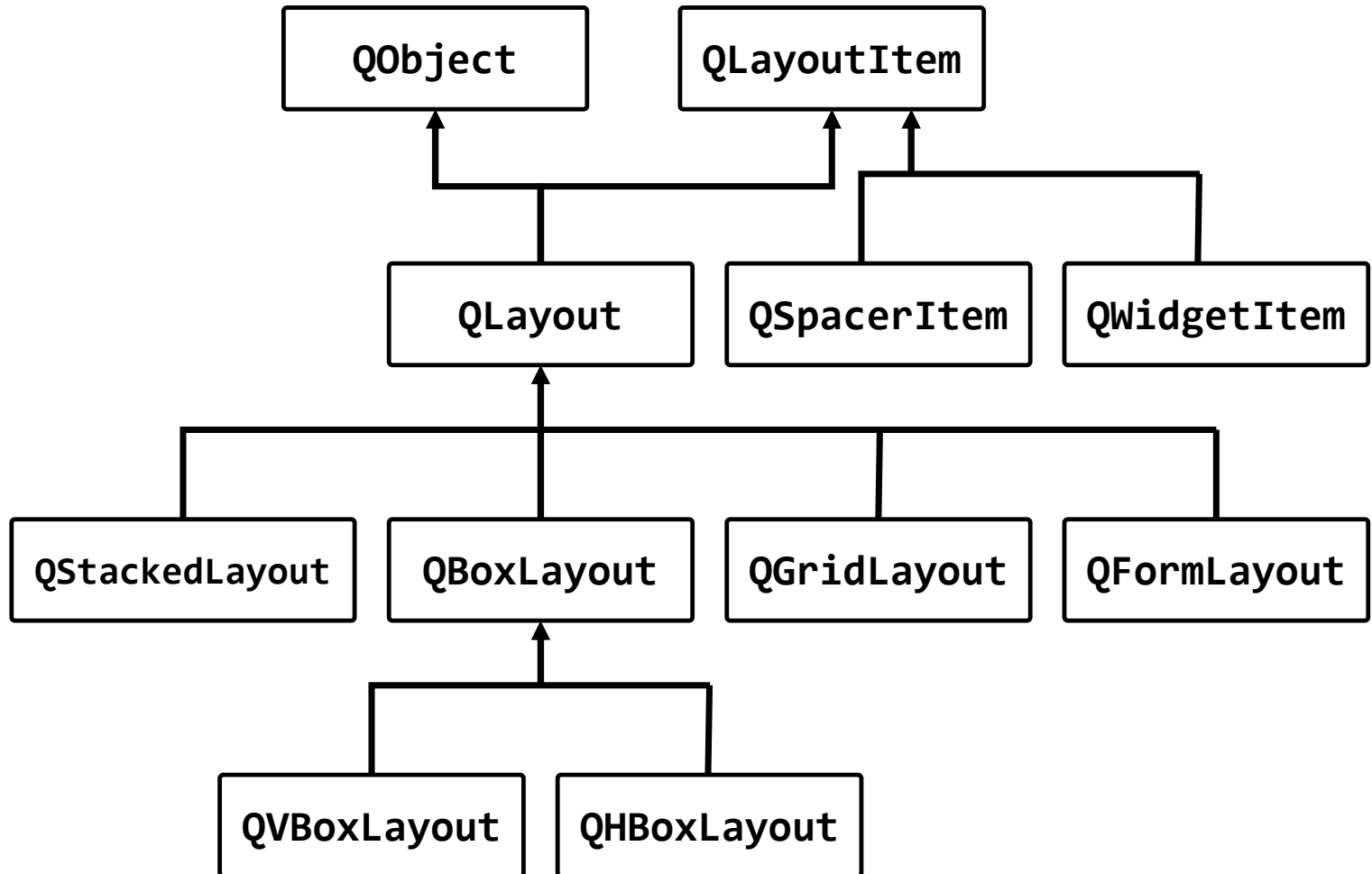
핵심 정리

- 자식 윈도우를 만든 경우 위치/크기를 적절하게 배치해야 한다.
- 방법 1. 픽셀 좌표로 직접 지정하는 방법
 - ⇒ **QWidget::setGeometry()** 멤버 함수 사용
 - ⇒ 윈도우 크기가 고정되어 있고, 화면 크기가 항상 동일한 환경에서는 나쁘지 않은 방법.
- 방법 2. Layout 클래스를 사용
 - ⇒ QT는 다양한 정책을 가진 Layout 클래스 제공



핵심 정리

QT Layout 관련 클래스





핵심 정리

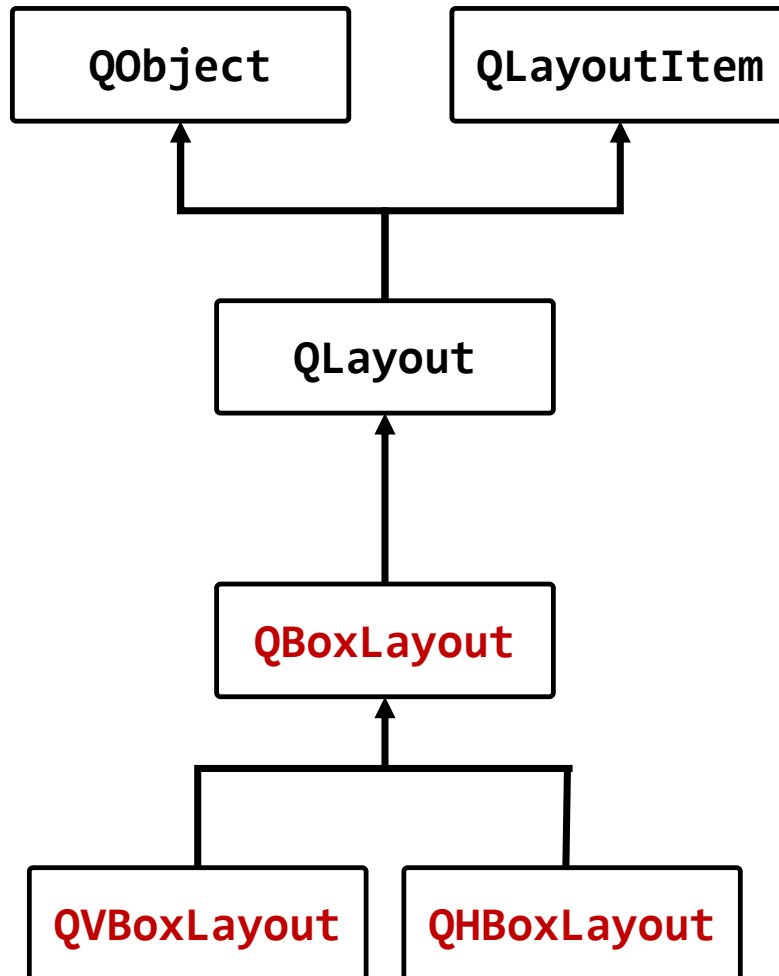
● **QBoxLayout**

⇒ 등록된 위젯(Widgets)들을 가로(또는 세로)로 차례로 배치하는 Layout

Constant		Description
QBoxLayout::LeftToRight	0	Horizontal from left to right.
QBoxLayout::RightToLeft	1	Horizontal from right to left.
QBoxLayout::TopToBottom	2	Vertical from top to bottom.
QBoxLayout::BottomToTop	3	Vertical from bottom to top.



● **QBoxLayout class hierachy**



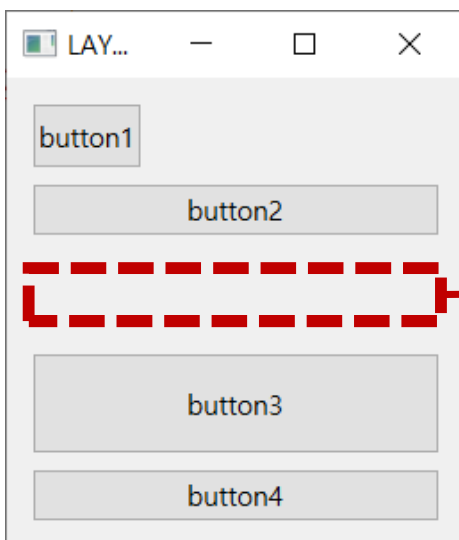


핵심 정리

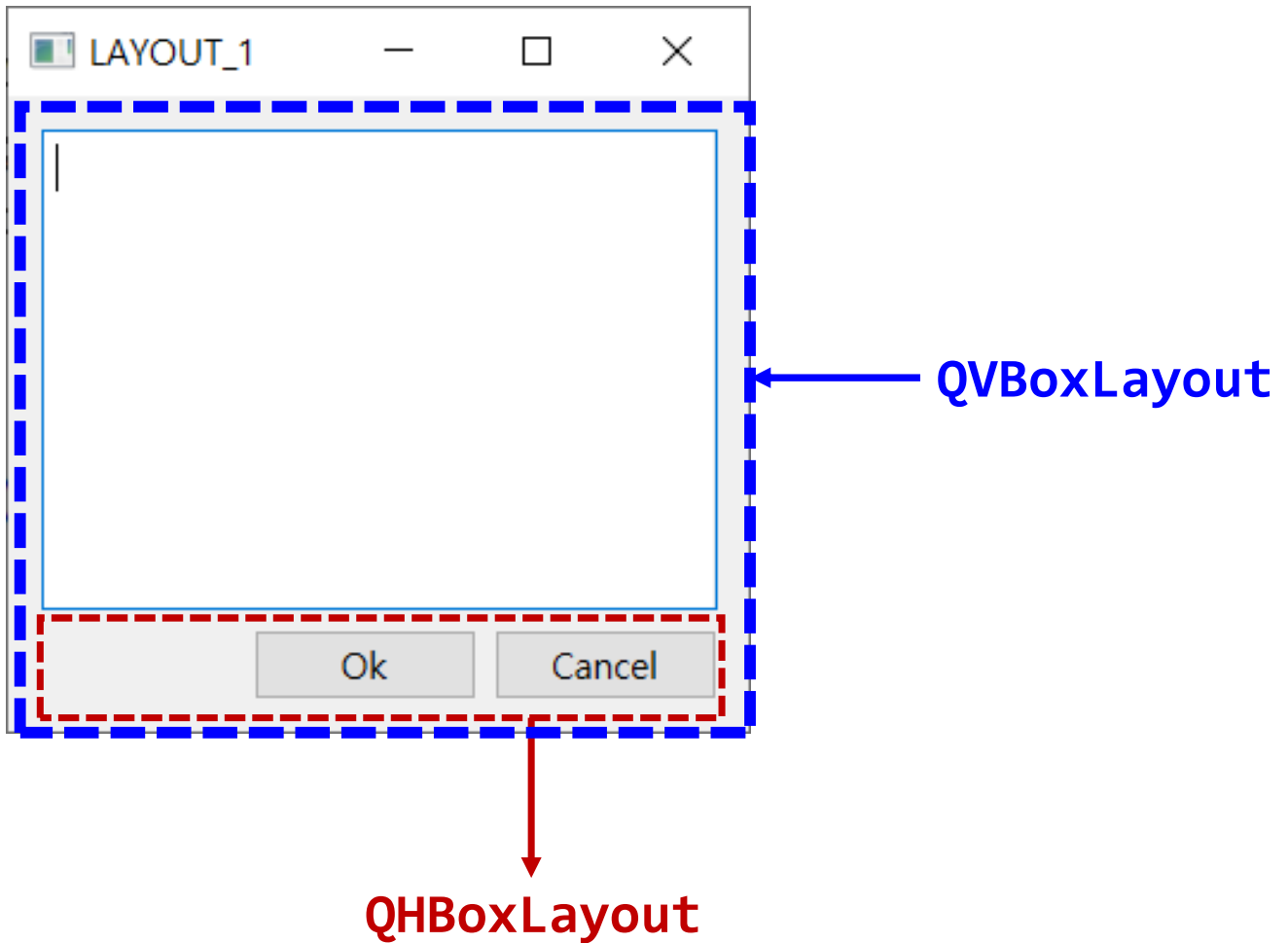
- **QWidget::setSizePolicy()**

⇒ Layout 의 크기 변경시 자신의 크기를 어떻게 변경 할지 정책을 지정.

- **QLayout::addStretch()**



size 조절시
Stretch 가 증가



- **중첩된 layout**

⇒ layout 은 다른 layout 을 포함할 수 있다.



QGridLayout

QFormLayout



핵심 정리

예제 Project

Project Name	LAYOUT_2
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp grid.h form.h



핵심 정리

● QGridLayout

⇒ 자식 Widget를 Row, Col 의 격자 형태로 배치

ID	<input type="text"/>
PASS	<input type="text"/>
EMAIL	<input type="text"/>



● QFormLayout

⇒ `addRow()` 함수를 사용해서 줄 단위로 항목을 추가

QFormLayout

ID	<input type="text"/>
PASS	<input type="text"/>
EMAIL	<input type="text"/>



Menu



핵심 정리

예제 Project

Project Name	MENU
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp menu1.h menu2.h



핵심 정리

QObject

대부분의 QT 클래스가
가져야 하는 공통의 특징
(RTTI, signal slot 등)

QWidget

모든 UI 관련 클래스가
가져야 하는 공통의 특징

QMainWindow

프로그램의 주 윈도우가
가져야 하는 특징 추가
(메뉴, 툴바, 상태바 등)



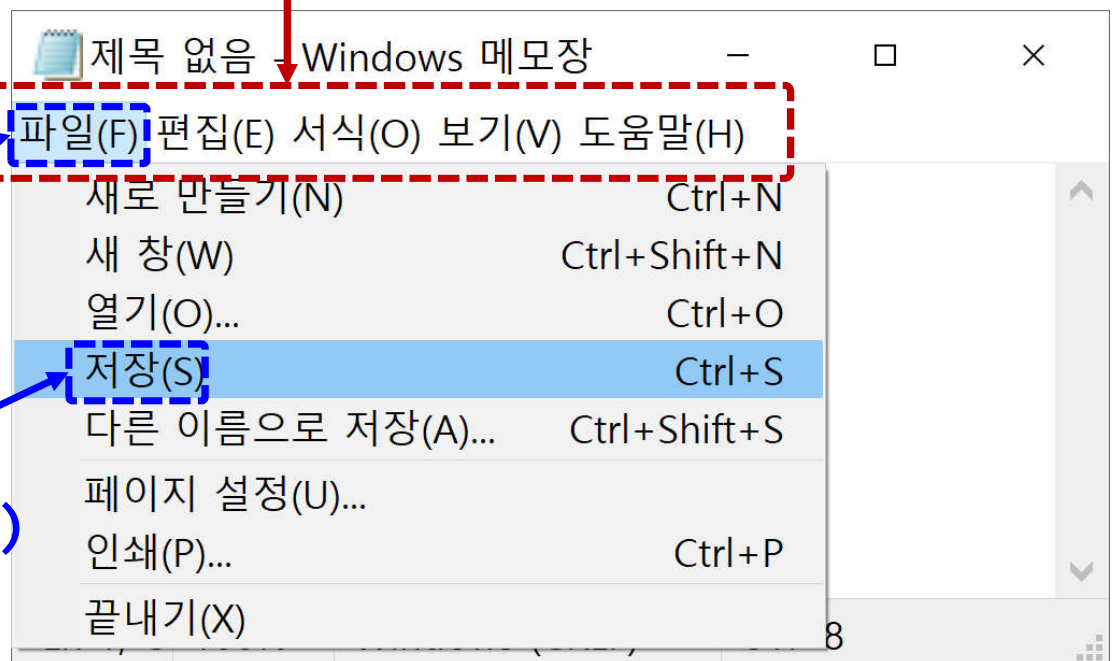
핵심 정리

● 메뉴 관련 용어 와 클래스

메뉴 바(QMenuBar)

메뉴(QMenu)

액션(QAction)





QMainWindow



핵심 정리

예제 Project

Project Name	MAINWINDOW
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp mainwindow.h



핵심 정리

QObject

대부분의 QT 클래스가
가져야 하는 공통의 특징
(RTTI, signal slot 등)

QWidget

모든 UI 관련 클래스가
가져야 하는 공통의 특징

QMainWindow

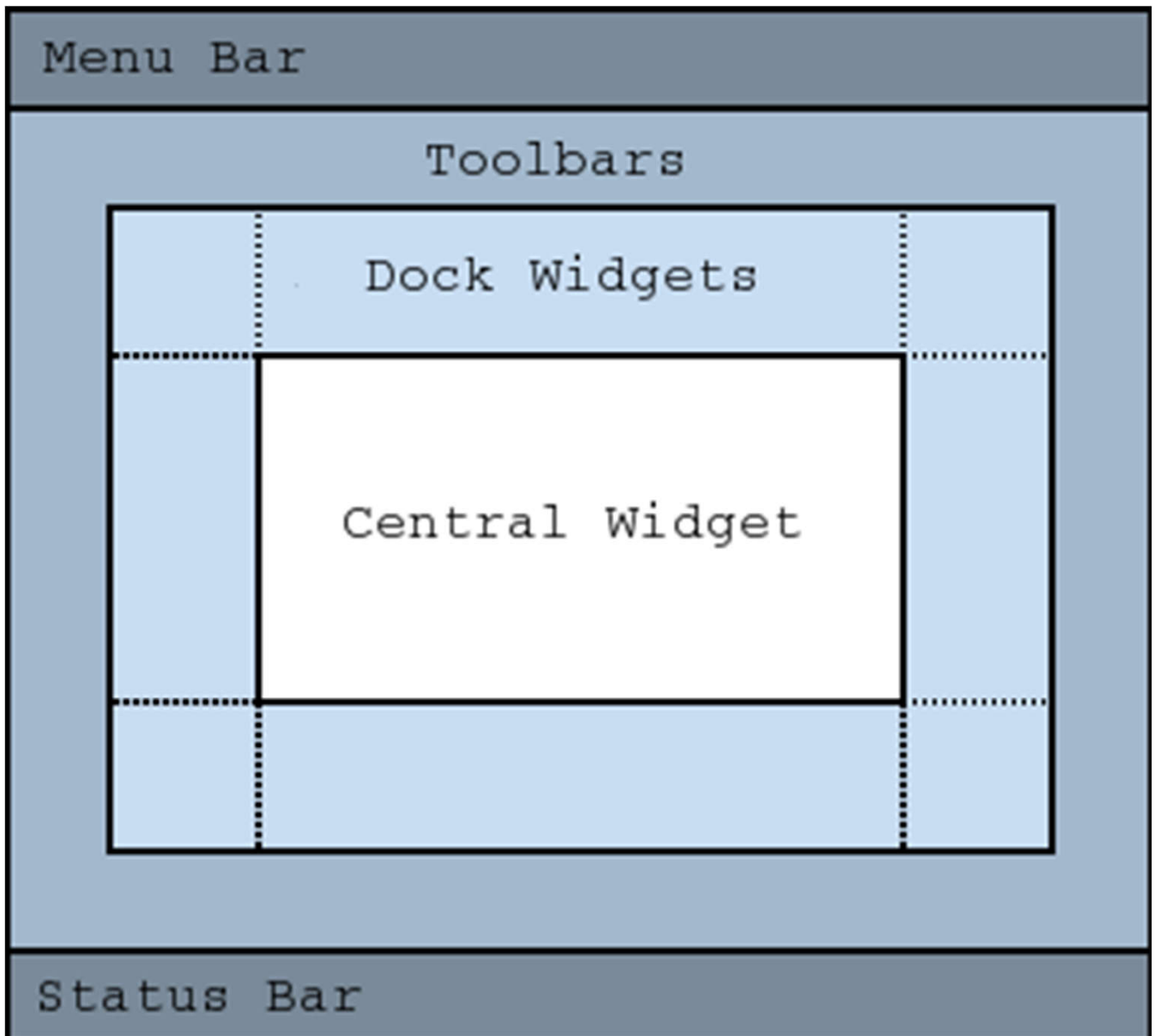
프로그램의 주 윈도우가
가져야 하는 특징 추가
(메뉴, 툴바, 상태바 등)



핵심 정리

- QMainWindow

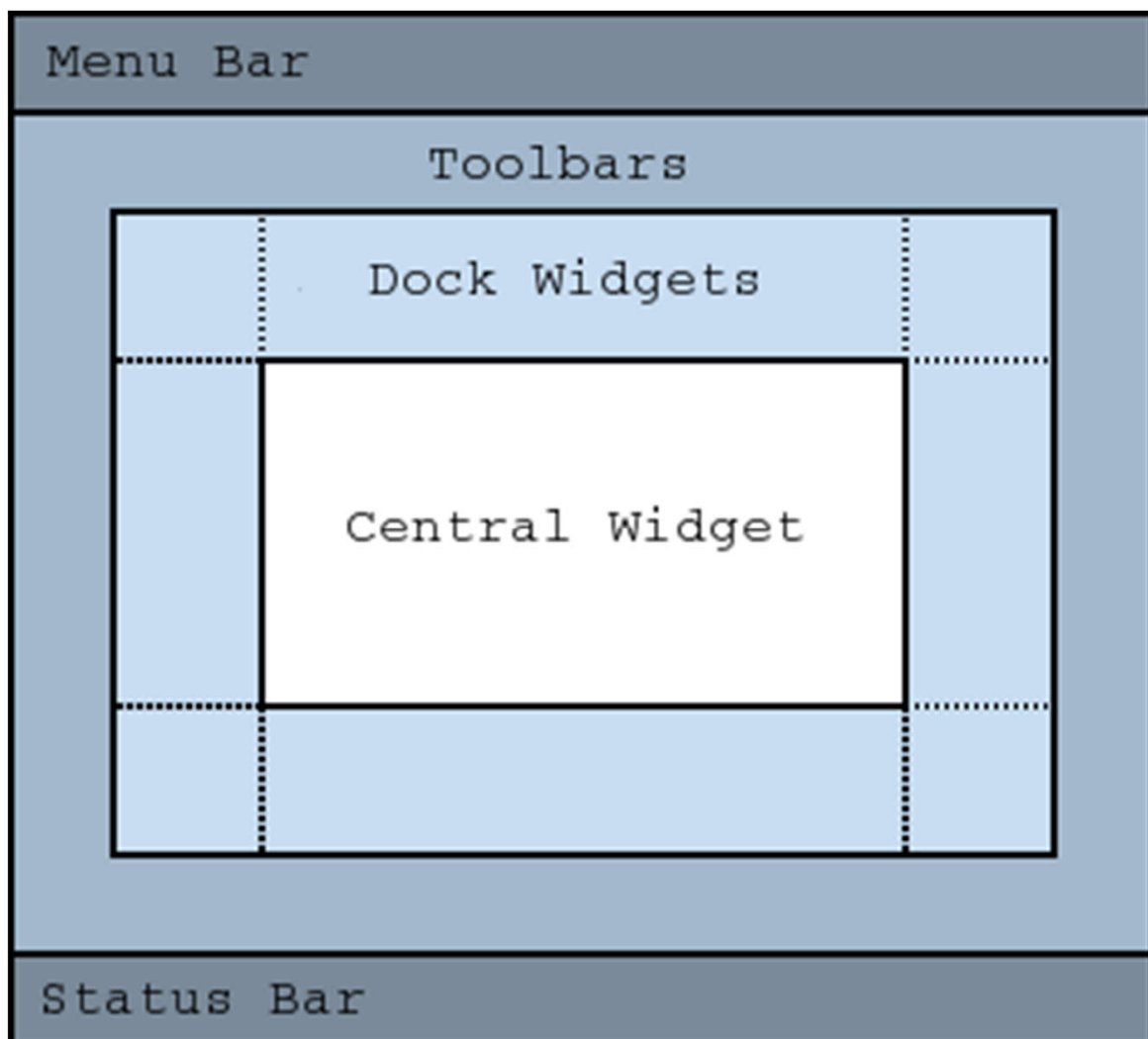
- ⇒ 내부적으로 Layout 을 이미 가지고 있음.
- ⇒ 사용자가 새로운 layout 을 지정할 수 없음.





핵심 정리

- `QMainWindow::setCentralWidget()`
 - ⇒ 일반적으로 `QTextEdit` 나 `QGraphicsView` 등을 사용.
 - ⇒ `Layout` 을 사용하고 싶다면, 별도의 `Widget` 을 지정하고, `Widget` 에 `Layout` 지정





Menu & QWidget



핵심 정리

예제 Project

Project Name	WIDGETMENU
Project Type	Empty qmake Project
Module	QT += widgets
Source	main.cpp widgetmenu.h



핵심 정리

