

UMEÅ UNIVERSITET

January 10, 2015

Institutionen för Tillämpad fysik och elektronik

Laborationsrapport

Laboration 6

Script och webprogrammering 7,5 hp

Projekt

Namn Christer Jakobsson

E-mail dv12cjn@cs.umu.se

Handledare

Ola Ågren, Kalle Prorok

Contents

Contents	1
I Uppgiftsrapport	3
1 Problembeskrivning	3
2 Åtkomst och användarhandledning	3
2.1 Filer som ingår i lösningen	3
2.1.1 Python Klient	3
2.1.2 PHP server	4
2.2 Request	4
2.3 Körning	5
2.4 Handledning för olika åtgärder	5
2.4.1 Ta bort bok	5
2.4.2 Ändra en boks värden	7
2.4.3 Lägg till bok	8
2.4.4 Visa en boks data	9
2.4.5 Avsluta programmet	10
2.5 Notifikationer	10
3 Systembeskrivning	10
3.1 Client	10
3.2 BookTable	12
3.3 Book	12
3.4 TabWidget	12
4 Testkörningar	13
4.0.1 Test 1	13
4.0.2 Test 2	14
4.0.3 Test 3	15
4.0.4 Test 4	16
5 Diskussion	16
II PHP kod	16
III Genererat av Epydoc	21
6 Module ClientClass	22
6.1 Functions	22
6.2 Variables	22
6.3 Class Client	22
6.3.1 Methods	22
6.3.2 Properties	26

6.3.3	Class Variables	26
7	Module TabClass	27
7.1	Variables	27
7.2	Class TabWidget	27
7.2.1	Methods	27
7.2.2	Properties	31
7.2.3	Class Variables	31
8	Module BookClass	32
8.1	Variables	32
8.2	Class Book	32
8.2.1	Methods	32
9	Module BookTable	33
9.1	Variables	33
9.2	Class BookTable	33
9.2.1	Methods	34
9.2.2	Properties	38
9.2.3	Class Variables	38

Part I

Uppgiftsrapport

1 Problembeskrivning

Tanken med denna laboration är att jag ska använda mig av det jag har lärt mig genom kursens gång och skapa något som en slutanvändare kan använda sig av. De ansvariga för kursen gav två stycken förslag på vad detta kunde vara och jag valde att göra ett bokhanteringssystem, där man med ett grafiskt program kan lägga upp böcker man läst som ska lagras i en databas på en server.

Laborationen består av ett grafiskt klientprogram skapat i python för användaren. Samt ett php script på en webbserver som i sin tur kommunicerar med en *MySql databas*.

2 Åtkomst och användarhandledning

2.1 Filer som ingår i lösningen

2.1.1 Python Klient

`ClientClass.py` Denna klass är huvudprogrammet, som skapar ett fönster som befolkas med objekt för att visa ett grafiskt gränssnitt. Innehåller metoder som gör http requests till PHP servern.

`BookClass.py` En klass som är databärare för en boks data.

Klassen lagrar:

- Author
- Title
- Genre
- Genre2
- Date Read
- Grade
- Comments

`BookTable.py` En klass som implementerar `QTableWidget` och skapar en tabell utifrån det data den får, i detta fall böckernas data.

TabClass.py En klass som skapar flikar i programmet och där varje flik innehåller textfält, labels och knappar för att användaren ska kunna utföra olika uppgifter.

- **Table:** Visar tabellen och användaren kan välja att ta bort en bok, ändra en boks data eller uppdatera tabellen.
- **Add Book:** Innehåller fält och knappar där användaren kan lägga till en bok i databasen.
- **Get Book:** Innehåller textfält, labels och knappar så en användaren kan få ut informationen för en titel och enkelt kunna kopiera värdena för att använda till annat.
- **Change Book:** Liknar Add Book förutom att användaren med denna flik kan *ändra* en boks värde.

Rekomenderar att använda *pip* för att installera biblioteket.

2.1.2 PHP server

BookHandler.php Innehåller PHP kod som tar emot http requests och kommunicerar med databasen för att åstadkomma det givna kommandot och returnera datat som efterfrågats eller lägga in data i databasen.

2.2 Request

För att kunna köra python klienten så måste användaren ha ett bibliotek vid namn *Request* installerat. Bruksanvisning för hur man installerar detta hittar man på

<http://docs.python-requests.org/en/latest/user/install>.

Rekommenderar att använda *pip* för enklast installation.

2.3 Körning

Användare kommer att använda sig av den grafiska klienten som är skriven i python. För att starta klienten så öppnar an en terminal i den katalog som python filerna ligger i och skriver *python ClientClass.py*

Detta kommer att starta programmet som kommer att visa ett grafiskt gränssnitt:

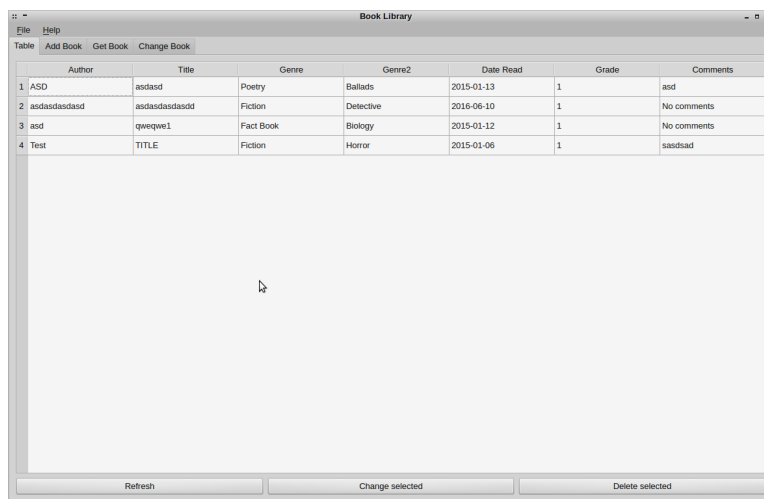


Figure 1: Program vid start

Användaren kan sedan använda detta program för att utföra olika uppgifter.

2.4 Handledning för olika åtgärder

Här kommer det att visas hur man går tillväga för att utföra de funktioner som programmet tillhandahåller.

2.4.1 Ta bort bok

1. Öppna *Table* filen.
2. Välj den bok som ska raderas. (Klicka på den så den markeras).
3. Klicka på knappen *Delete selected*.
4. En popup visar om det lyckades.

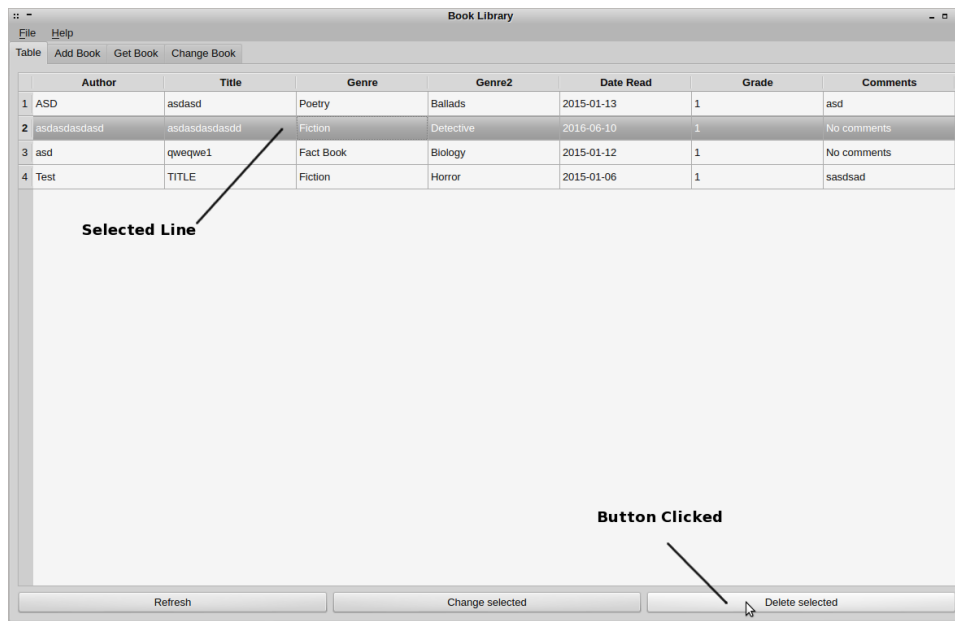


Figure 2: Steps 2-3

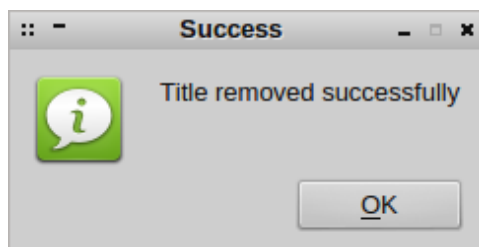


Figure 3: Step 4

2.4.2 Ändra en boks värden

1. Öppna *Table* fliken.
2. Välj den bok som ska ändras. (Klicka på den så den markeras).
3. Klicka på knappen *Change selected*, detta kommer att öppna fliken *Change Book*.
4. Ändra de värden som ska ändras.
5. Klicka på knappen *Submit changes*, om alla ändrade värden är korrekta för dess respektive fält så utförs ändringen.
6. En popup visar om det lyckades.

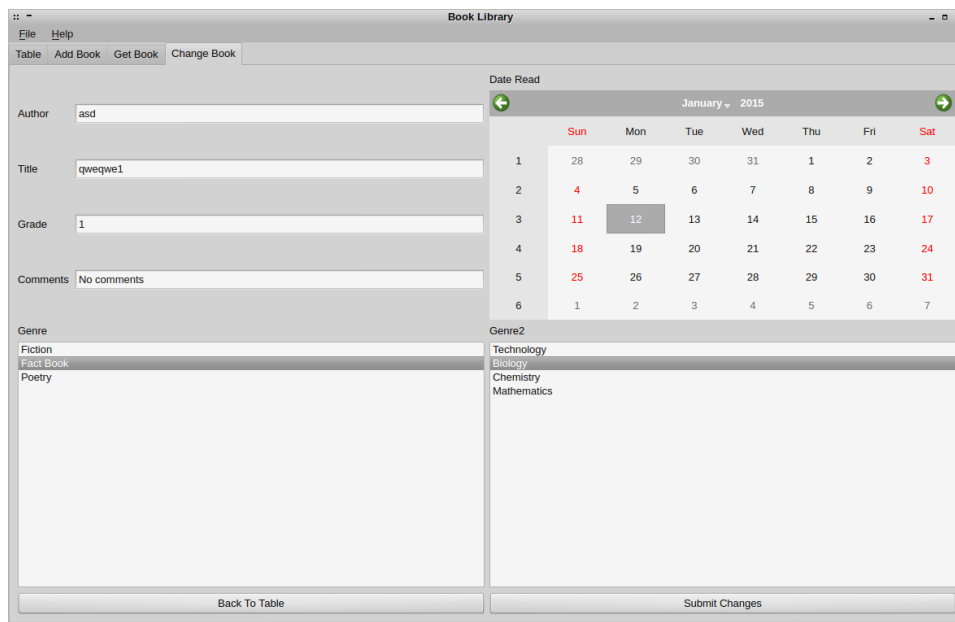


Figure 4: Vy över *Change Book* fliken efter att en bok har valts i steg 2-3

2.4.3 Lägg till bok

1. Öppna fliken *Add Book*.

2. Fyll i värden i respektive fält.

Author får ej innehålla siffror

Alla fält förutom *Comments* är nödvändiga.

Listan *Genre* måste få ett val innan listan *Genre2* visas.

3. Klicka på *Submit Data*.

4. En popup visas om det lyckades.

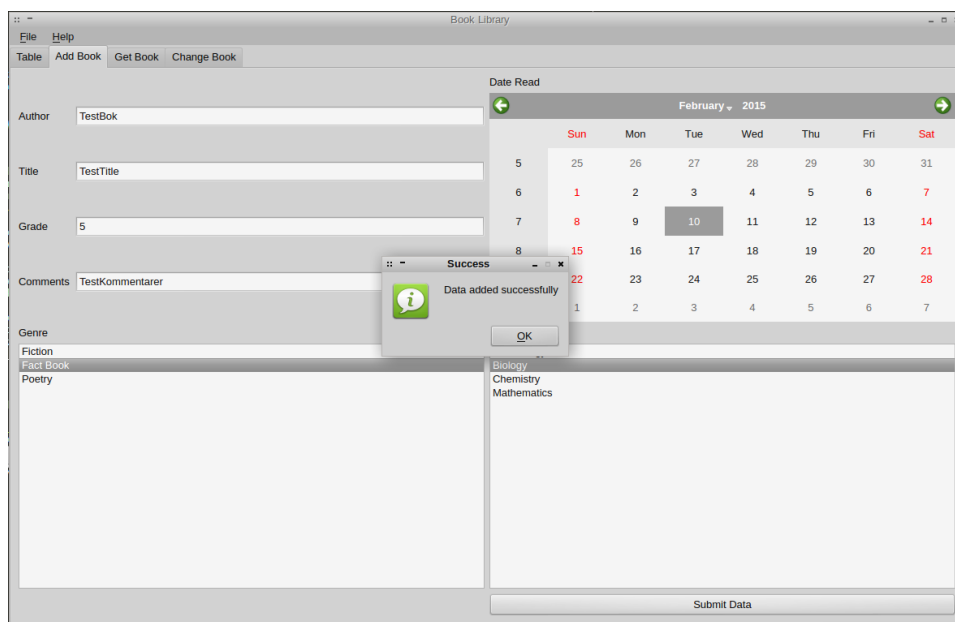


Figure 5: Vy när bok lyckats läggas in i databasen. Steg 4

2.4.4 Visa en boks data

1. Öppna fliken *Get Book*.
2. Klicka på *Refresh list*.
3. Klicka på den titel som ska visas.
4. Klicka på *Get title data*
5. Nu finns den titelns värden i fälten ovanför titel-listan.
6. En popup visas vid fel.

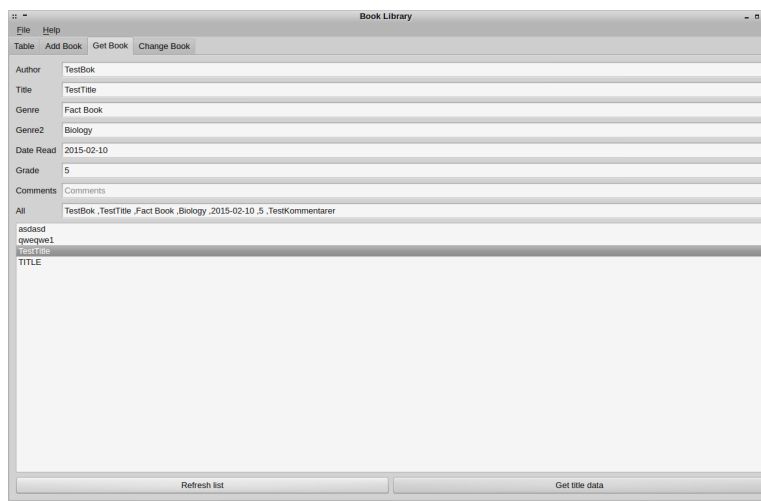


Figure 6: Steg 5

2.4.5 Avsluta programmet

Programmet avslutas som ett vanligt program genom att klicka på krysset för fönstret, trycka ALT+F4 med flera.

2.5 Notifikationer

För varje handling som kommunicerar med PHP servern så visas en notifikation om begäran misslyckades, i vissa fall så visas även en notifikation när begäran lyckades.

3 Systembeskrivning

3.1 Client

Huvudklassen i programmet, skapar ett QMainWindow som visar innehållet från *TabClass*. Innehåller metoder som skickar http requests till PHP servern genom att använda sig av *JSON* objekt.

De requests som kan göras är:

`add_book`: Skapar ett *JSON* objekt innehållande information om boken som ska läggas in och skickar det till servern. Om servern svarar med 201 som status code så lyckades boken läggas till.

[Format på JSON]

```
{'command': 'add_book',  
  'title': book.title,  
  'author': book.author,  
  'genre': book.genre,  
  'genre2': book.genre2,  
  'dateRead': book.date\_read,  
  'grade': book.grade,  
  'comments': book.comments}
```

`get_title`: Skapar ett *JSON* objekt med titeln som efterfrågas och skickar det till servern. Servern skickar tillbaka datat för titeln i ett som JSON och klienten skapar en *Book* som har sparat värdena i respektive variabel.

[Klient JSON]

```
{'command': 'get_title', 'title': 'TITLE_TO_GET'}
```

[Server JSON]

```
{'dateRead': 'DATE',  
  'genre2': 'GENRE2',  
  'title': 'TITLE',
```

```
'grade': 'GRADE',  
'author': 'AUTHOR',  
'comments': 'COMMENTS',  
'genre': 'GENRE'
```

`get_all_books`: Skapar ett *JSON* objekt med ett kommando att servern ska skicka alla böckers information. Om förfrågan lyckades (status code 202) så skapas en *Book* för varje bok och dess värden sätts till värdena utifrån *JSON* objektet. Returnerar en lista med *Books*

[Klient JSON]

```
{'command': 'get_whole_list'}
```

[Server JSON]

```
[{'dateRead': 'DATE',  
'genre2': 'GENRE2',  
'title': 'TITLE',  
'grade': 'GRADE',  
'author': 'AUTHOR',  
'comments': 'COMMENTS',  
'genre': 'GENRE'}, {BOK_2}....]
```

`get_all_titles`: Skapar ett *JSON* objekt:

```
{'command': 'get_list'}
```

och om requesten lyckades (status code 202) så formateras *JSON* objektet om till en lista som innehåller alla titlar.

[Server JSON]

```
[{'title': 'TITLE_1'}, {'title': 'TITLE_2'},...]
```

`remove_title`: Skapar ett *JSON*:

```
{'command': 'remove_book', 'titleToRemove': 'TITLE'}
```

Servern svarar med 200 som status om det lyckades.

`submit_changes`: Skapar ett *JSON* objekt och skickar det till servern. Om status response är 200 så lyckades uppdateringen av boken.

[Klient JSON]

```
{'command': 'add_book',  
'title': book.title,  
'author': book.author,  
'genre': book.genre,
```

```
'genre2': book.genre2,  
'dateRead': book.date\_read,  
'grade': book.grade,  
'comments': book.comments}
```

3.2 BookTable

Denna klass använder sig av `QTableWidget` för att visa en tabell med den sorts data den får, i detta fall bokdata. Det går bara att markera en rad åt gången i tabellen och klassen innehåller metoder för att få ut den titel boken man klickade på har samt all data som en bok har, i detta fall så används klassen *Book* för att lagra datat.

Används i *TabClass* klassens flik *Table* för att visa alla böcker.

3.3 Book

Denna klass används som en databärare, och innehåller 7 stycken variabler vars data är den data som en bok kan ha.

Klassen innehåller variablerna:

- author
- title
- genre
- genre2
- date_read
- grade
- comments

Klassen används i *Client* och i *TabWidget* och används för att enkelt kunna spara och komma åt datat för en bok.

3.4 TabWidget

Denna klass skapar en layout som innehåller 4 flikar där varje flik innehåller objekt för att utföra en viss funktion.

De flikar som finns är:

Table: Visar en tabell med all data som finns i databasens tabell, dvs alla böcker som användaren har lagt in. Samt knappar för att utföra

uppgifter med den markerade boken.

Add Book: Innehåller inputfält för det data som en bok har. Samt en knapp för att lägga in boken i databasen förutsatt att alla fält är korrekt ifyllda.

Get Book: Under denna flik kan användaren få fram en boks information utifrån en viss titel, informationen visas i textfält och går att kopiera för användning till annat.

Change Book: Avänds för att ändra en specifik boks data. Ifrån *Table* fliken kan man markera en bok, klicka på *Change selected* för att sedan flyttas till denna flik där man kan ändra valfritt fält för boken och sedan uppdatera boken i databasen. Förutsatt att alla fält är korrekt ifyllda.

Klassen används av *ClientClass* som huvudlayout. och innehåller metoder som lyssnar på när knapparna blir tryckta, listorna blir ändrade och när ett nytt datum blir markerat.

4 Testkörningar

4.0.1 Test 1

Submit changes utan att ha valt och ändrat en bok.

Jag har öppnat fliken utan att gå via att markera i tabellen och klickar på *Change selected*.

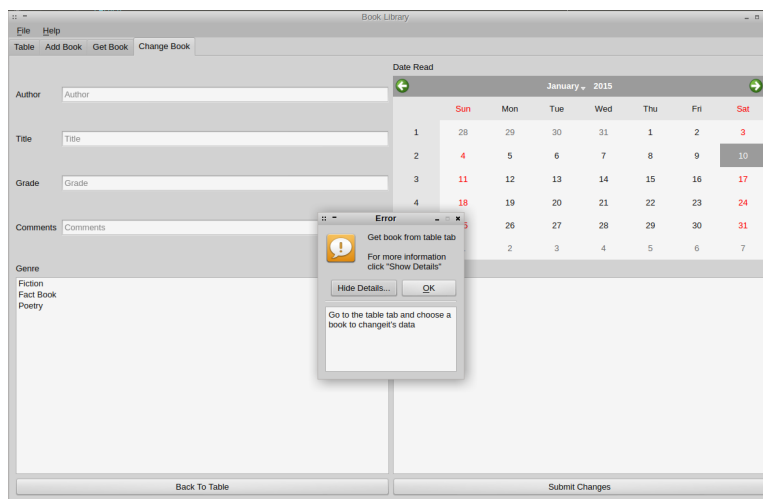


Figure 7: Händelse vid klick av *Submit changes*

Kommentar: Fungerar som tänkt, kort förklaring för vad som är fel om användaren klickar på *Show Details*.

4.0.2 Test 2

Siffror i author

Jag fyller i formuläret korrekt på alla punkter förutom att jag fyller i *123* som author, vilket är ett inkorrekt värde för fältet.

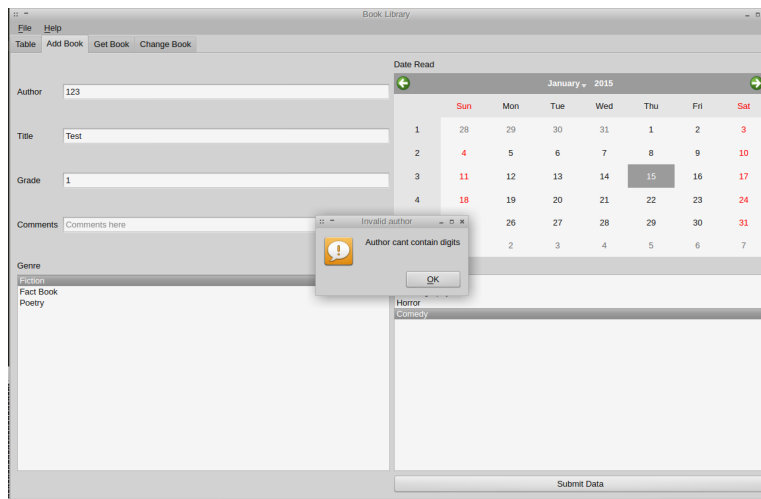


Figure 8: Händelse vid klick av *Submit Data*

Kommentar: Förväntat resultat. Visar vilket fält som är fel och vad som är fel.

4.0.3 Test 3

Lägga till en bok med en titel som redan finns.

Jag lägger in en bok som heter *Test* i databasen och försöker sedan att lägga in en likadan.

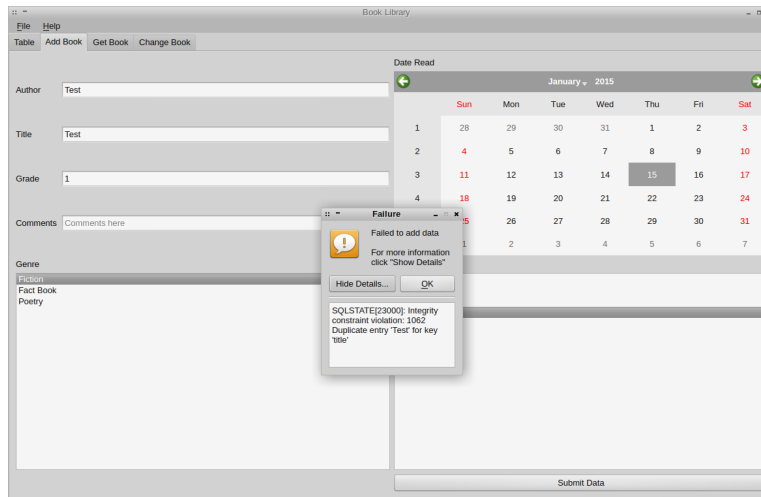


Figure 9: Händelse vid klick av *Submit Data*

Kommentar: Förväntat resultat. Här kan vi även se att under Show Details så visas det exception som *MySql* kastade när vi försökte stoppa in boken igen.

4.0.4 Test 4

Vad händer om servern är nere?

Jag byter namn på php filen som ligger på servern som klienten skickar requests till.

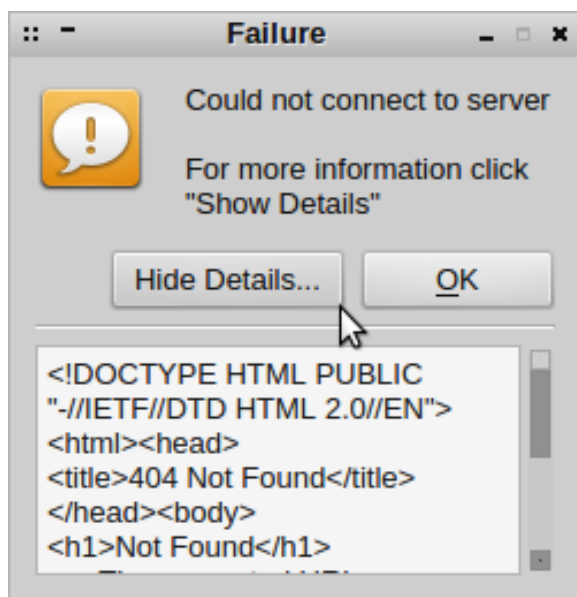


Figure 10: Händelse vid klick av *Submit Data*

Kommentar: Ok resultat. notifikationen visas innan huvudfönstret visas på mitt system. när jag trycker på *Ok* på notifikationsrutan så visas huvudfönstret.

5 Diskussion

Det har varit kul att göra något där man har haft lite fria tyglar och fått göra något som skulle kunna användas i verkligheten. Det känns som att de flesta program som har högt nyttaovärde använder sig av någon sorts server som lagrar datat åt klienterna eller som skickar data emellan klienterna så att få öva mera det har varit bra.

Kul att få hålla på med http requests via både python och php. Det var lite klurigt innan jag hittade ett sätt som kändes både smidigt, snyggt och säkert. Kul att få hålla på med *Querys* till en databas, har liten erfarenhet av att jobba med databaser. Jag skulle vilja utöka min lösning med mera queries och kanske ändra tillvägagångssätten för vissa åtgärder så att det mesta sköts via tabellen istället, men annars är jag ganska nöjd.

Part II

PHP kod

```
1
2 <?php
3
4 $data = json_decode(file_get_contents('php://input'),
    true);
5
6 $dbhost = "localhost";
7 $dbusername = "root";
8 $dbuserpass = "XXXXXXXXX";
9 $dbname = "myDb";
10
11 if(is_null($data)) {
12     http_response_code(404);
13     echo "Error: No data found";
14 }
15 else if ($data['command'] == "add_book") {
16     $author = $data['author'];
17     $title = $data['title'];
18     $genre = $data['genre'];
19     $genre2 = $data['genre2'];
20     $date = $data['dateRead'];
21     $grade = $data['grade'];
22     $comments = $data['comments'];
23
24     try {
25         $conn = new PDO("mysql:host=$dbhost;dbname=
            $dbname", $dbusername, $dbuserpass);
26         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::
            ERRMODE_EXCEPTION);
27
28         $create = "CREATE TABLE IF NOT EXISTS
            book_table (
29             author VARCHAR(30) NOT NULL,
30             title VARCHAR(30) NOT NULL UNIQUE,
31             genre VARCHAR(30) NOT NULL,
32             genre2 VARCHAR(30) NOT NULL,
33             dateRead DATE NOT NULL,
34             grade ENUM('1','2','3','4','5') NOT NULL,
35             comments VARCHAR(50))";
36         $conn->exec($create);
37
38         $sql = "INSERT INTO book_table (author,
            title, genre, genre2, dateRead, grade
            , comments) VALUES
39             (:author, :title, :genre, :genre2, :dateRead, :
            grade, :comments)";
40
41         $smtp = $conn->prepare($sql);
42         $smtp->bindParam(':author', $author);
```

```
43         $smtp->bindParam(':title', $title);
44         $smtp->bindParam(':genre', $genre);
45         $smtp->bindParam(':genre2', $genre2);
46         $smtp->bindParam(':dateRead', $date);
47         $smtp->bindParam(':grade', $grade);
48         $smtp->bindParam(':comments', $comments);
49         $smtp->execute();
50         http_response_code(201);
51     }
52     catch(PDOException $e) {
53         echo $e->getMessage();
54         http_response_code(404);
55     }
56     $conn = null;
57 }
58 else if ($data['command'] == "update_book") {
59     $author = $data['author'];
60     $title = $data['title'];
61     $genre = $data['genre'];
62     $genre2 = $data['genre2'];
63     $dateRead = $data['dateRead'];
64     $grade = $data['grade'];
65     $comments = $data['comments'];
66
67     try {
68         $conn = new PDO("mysql:host=$dbhost;dbname=
69             $dbname", $dbusername, $dbuserpass);
70
71         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::
72             ERRMODE_EXCEPTION);
73         $sql = "UPDATE book_table SET
74             author=:author,
75             title=:title,
76             genre=:genre,
77             genre2=:genre2,
78             dateRead=:dateRead,
79             grade=:grade,
80             comments=:comments WHERE title=:title";
81
82         $smtp = $conn->prepare($sql);
83         $smtp->bindParam(':author', $author);
84         $smtp->bindParam(':title', $title);
85         $smtp->bindParam(':genre', $genre);
86         $smtp->bindParam(':genre2', $genre2);
87         $smtp->bindParam(':dateRead', $date);
88         $smtp->bindParam(':grade', $grade);
89         $smtp->bindParam(':comments', $comments);
90         $smtp->execute();
91
92         if($smtp->rowCount() == 0) {
93             echo "No book updated.";
94             http_response_code(404);
95         } else {
```

```
95             http_response_code(202);
96         }
97     }
98     catch(PDOException $e) {
99         echo $e->getMessage();
100         http_response_code(404);
101     }
102     $conn = null;
103 }
104 else if($data['command'] == "get_list") {
105     try {
106
107         $conn = new PDO("mysql:host=$dbhost;
108             dbname=$dbname", $dbusername,
109             $dbuserpass);
110         $conn->setAttribute(PDO::ATTR_ERRMODE,
111             PDO::ERRMODE_EXCEPTION);
112
113         $array = $conn->query("SELECT title FROM
114             book_table")->fetchAll(PDO::
115             FETCH_ASSOC);
116         echo json_encode($array);
117         http_response_code(202);
118     }
119     catch(PDOException $e) {
120         http_response_code(404);
121         echo $e->getMessage();
122     }
123     $conn = null;
124 }
125 else if($data['command'] == "get_title") {
126     try {
127         $title = $data['title'];
128
129         $conn = new PDO("mysql:host=$dbhost;
130             dbname=$dbname", $dbusername,
131             $dbuserpass);
132         $conn->setAttribute(PDO::ATTR_ERRMODE,
133             PDO::ERRMODE_EXCEPTION);
134
135         $array = $conn->query("SELECT * FROM
136             book_table WHERE title='$title'")->
137             fetchAll(PDO::FETCH_ASSOC);
138
139         if(count($array) == 0) {
140             echo "Title ".$title." not found
141             .";
142             http_response_code(404);
143         } else {
144             echo json_encode($array);
145         }
146     }
147     catch(PDOException $e) {
```

```
138             echo $e->getMessage();
139             http_response_code(404);
140         }
141         $conn = null;
142     }
143     else if($data['command'] == "get_whole_list") {
144         try {
145             $title = $data['title'];
146
147             $conn = new PDO("mysql:host=$dbhost;
148                             dbname=$dbname", $dbusername,
149                             $dbuserpass);
150             $conn->setAttribute(PDO::ATTR_ERRMODE,
151                                 PDO::ERRMODE_EXCEPTION);
152
153             $return_arr = array();
154
155             $array = $conn->query("SELECT * FROM
156                                   book_table")->fetchAll(PDO::
157                                       FETCH_ASSOC);
158             http_response_code(202);
159             echo json_encode($array);
160         }
161         catch(PDOException $e) {
162             echo $e->getMessage();
163             http_response_code(404);
164         }
165         $conn = null;
166     }
167     else if($data['command'] == "remove_book") {
168         try {
169             $titleToRemove = $data['titleToRemove'];
170
171             $conn = new PDO("mysql:host=$dbhost;
172                             dbname=$dbname", $dbusername,
173                             $dbuserpass);
174             $conn->setAttribute(PDO::ATTR_ERRMODE,
175                                 PDO::ERRMODE_EXCEPTION);
176
177             $sql = 'DELETE from book_table where
178                     title=:title';
179             $preparedStatement = $conn->prepare($sql)
180                 ;
181             $preparedStatement->execute(array(':title
182                 ' => $titleToRemove));
183
184             if($preparedStatement->rowCount() == 0) {
185                 http_response_code(400);
186                 echo "Title \"\".$titleToRemove
187                     .\"\" not found, nothing was
188                     deleted";
189             } else {
190                 http_response_code(202);
191             }
192         }
193     }
```

```
179         }
180     }
181     catch(PDOException $e) {
182         http_response_code(404);
183         echo $e->getMessage();
184     }
185     $conn = null;
186 }
187 else {
188     echo 'POST ';
189     http_response_code(404);
190 }
191
192 ?>
```

Part III

Genererat av Epydoc

6 Module ClientClass

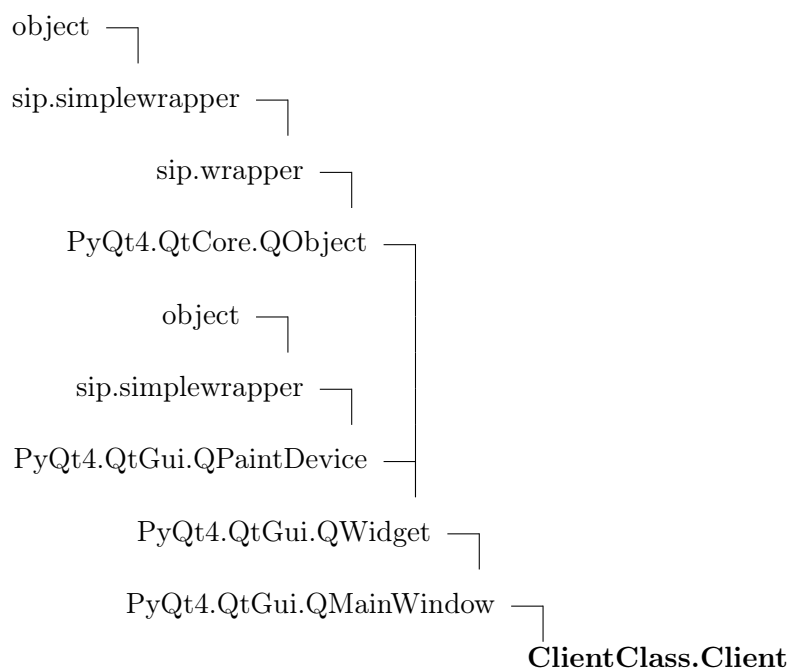
6.1 Functions

main()

6.2 Variables

Name	Description
<code>__package__</code>	Value: None

6.3 Class Client



6.3.1 Methods

<code>__init__(self, parent=None)</code> <code>x.__init__(...)</code> initializes x; see <code>help(type(x))</code> for signature Overrides: <code>object.__init__</code> <code>exitit</code> (inherited documentation)
--

<code>get_whole_list()</code>

<code>make_menu(self)</code>

`exit_action()``help()``send_book_data(book)``center_on_screen(self)``closeEvent(self, event)`Overrides: `PyQt4.QtGui.QWidget.closeEvent``get_title_data(title_to_get)``show_popup(set_text, detailed, window_title,
message_type)``get_all_titles()``remove_title(title_to_remove)``submit_changes(book)`***Inherited from PyQt4.QtGui.QMainWindow***

`addDockWidget()`, `addToolBar()`, `addToolBarBreak()`, `centralWidget()`, `contextMenuEvent()`, `corner()`, `createPopupMenu()`, `dockOptions()`, `dockWidgetArea()`, `documentMode()`, `event()`, `iconSize()`, `iconSizeChanged()`, `insertToolBar()`, `insertToolBarBreak()`, `isAnimated()`, `isDockNestingEnabled()`, `isSeparator()`, `menuBar()`, `menuWidget()`, `removeDockWidget()`, `removeToolBar()`, `removeToolBarBreak()`, `restoreDockWidget()`, `restoreState()`, `saveState()`, `setAnimated()`, `setCentralWidget()`, `setCorner()`, `setDockNestingEnabled()`, `setDockOptions()`, `setDocumentMode()`, `setIconSize()`, `setMenuBar()`, `setMenuWidget()`, `setStatusBar()`, `setTabPosition()`, `setTabShape()`, `setToolButtonStyle()`, `setUnifiedTitleAndToolBarOnMac()`, `splitDockWidget()`, `statusBar()`, `tabPosition()`, `tabShape()`, `tabifiedDockWidgets()`, `tabifyDockWidget()`, `toolBarArea()`, `toolBarBreak()`, `toolButtonStyle()`, `toolButtonStyleChanged()`, `unifiedTitleAndToolBarOnMac()`

Inherited from PyQt4.QtGui.QWidget

`acceptDrops()`, `accessibleDescription()`, `accessibleName()`, `actionEvent()`, `actions()`, `activateWindow()`, `addAction()`, `ad-`

dActions(), adjustSize(), autoFillBackground(), background-
Role(), baseSize(), changeEvent(), childAt(), childrenRect(),
childrenRegion(), clearFocus(), clearMask(), close(), contents-
Margins(), contentsRect(), contextMenuPolicy(), create(),
cursor(), customContextMenuRequested(), destroy(), dev-
Type(), dragEnterEvent(), dragLeaveEvent(), dragMoveEvent(),
dropEvent(), effectiveWinId(), enabledChange(), ensurePol-
ished(), enterEvent(), find(), focusInEvent(), focusNextChild(),
focusNextPrevChild(), focusOutEvent(), focusPolicy(), fo-
cusPreviousChild(), focusProxy(), focusWidget(), font(), fontChange(),
fontInfo(), fontMetrics(), foregroundRole(), frameGeometry(),
frameSize(), geometry(), getContentsMargins(), grabGesture(),
grabKeyboard(), grabMouse(), grabShortcut(), graphicsEf-
fect(), graphicsProxyWidget(), handle(), hasFocus(), has-
MouseTracking(), height(), heightForWidth(), hide(), hideEvent(),
inputContext(), inputMethodEvent(), inputMethodHints(),
inputMethodQuery(), insertAction(), insertActions(), isAc-
tiveWindow(), isAncestorOf(), isEnabled(), isEnabledTo(),
isEnabledToTLW(), isFullScreen(), isHidden(), isLeftToRight(),
isMaximized(), isMinimized(), isModal(), isRightToLeft(),
isTopLevel(), isVisible(), isVisibleTo(), isWindow(), isWin-
dowModified(), keyPressEvent(), keyReleaseEvent(), keyboard-
Grabber(), languageChange(), layout(), layoutDirection(),
leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(),
mapFromParent(), mapTo(), mapToGlobal(), mapToParent(),
mask(), maximumHeight(), maximumSize(), maximumWidth(),
metric(), minimumHeight(), minimumSize(), minimumSize-
Hint(), minimumWidth(), mouseDoubleClickEvent(), mouseG-
rabber(), mouseMoveEvent(), mousePressEvent(), mouseRe-
leaseEvent(), move(), moveEvent(), nativeParentWidget(),
nextInFocusChain(), normalGeometry(), overrideWindowFlags(),
overrideWindowState(), paintEngine(), paintEvent(), palette(),
paletteChange(), parentWidget(), pos(), previousInFocusChain(),
raise_(), rect(), releaseKeyboard(), releaseMouse(), release-
Shortcut(), removeAction(), render(), repaint(), resetInput-
Context(), resize(), resizeEvent(), restoreGeometry(), save-
Geometry(), scroll(), setAcceptDrops(), setAccessibleDescrip-
tion(), setAccessibleName(), setAttribute(), setAutoFillBack-
ground(), setBackgroundRole(), setBaseSize(), setContents-
Margins(), setContextMenuPolicy(), setCursor(), setDisabled(),
setEnabled(), setFixedHeight(), setFixedSize(), setFixedWidth(),
setFocus(), setFocusPolicy(), setFocusProxy(), setFont(), set-
ForegroundRole(), setGeometry(), setGraphicsEffect(), setHid-
den(), setInputContext(), setInputMethodHints(), setLay-
out(), setLayoutDirection(), setLocale(), setMask(), setMax-
imumHeight(), setMaximumSize(), setMaximumWidth(), set-

MinimumHeight(), setMinimumSize(), setMinimumWidth(),
 setMouseTracking(), setPalette(), setParent(), setShortcu-
 tAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIn-
 crement(), setSizePolicy(), setStatusTip(), setStyle(), set-
 StyleSheet(), setTabOrder(), setToolTip(), setUpdatesEn-
 abled(), setVisible(), setWhatsThis(), setWindowFilePath(),
 setWindowFlags(), setWindowIcon(), setWindowIconText(),
 setWindowModality(), setWindowModified(), setWindowOpac-
 ity(), setWindowRole(), setWindowState(), setWindowTi-
 tle(), show(), showEvent(), showFullScreen(), showMaximized(),
 showMinimized(), showNormal(), size(), sizeHint(), sizeIn-
 crement(), sizePolicy(), stackUnder(), statusTip(), style(),
 styleSheet(), tabletEvent(), testAttribute(), toolTip(), topLevel-
 Widget(), underMouse(), ungrabGesture(), unsetCursor(),
 unsetLayoutDirection(), unsetLocale(), update(), updateGe-
 ometry(), updateMicroFocus(), updatesEnabled(), visibleRe-
 gion(), whatsThis(), wheelEvent(), width(), winId(), win-
 dow(), windowActivationChange(), windowFilePath(), win-
 dowFlags(), windowIcon(), windowIconText(), windowModal-
 ity(), windowOpacity(), windowRole(), windowState(), win-
 dowTitle(), windowType(), x(), x11Info(), x11PictureHandle(),
 y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), con-
 nect(), connectNotify(), customEvent(), deleteLater(), de-
 stroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(),
 dumpObjectTree(), dynamicPropertyNames(), emit(), event-
 Filter(), findChild(), findChildren(), inherits(), installEvent-
 Filter(), isWidgetType(), killTimer(), metaObject(), move-
 ToThread(), objectName(), parent(), property(), pyqtCon-
 figure(), receivers(), removeEventFilter(), sender(), sender-
 SignalIndex(), setObjectName(), setProperty(), signalsBlocked(),
 startTimer(), thread(), timerEvent(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalD-
 piY(), numColors(), paintingActive(), physicalDpiX(), phys-
 icalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(),

```
__reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()
```

6.3.2 Properties

Name	Description
<i>Inherited from object</i>	
<code>__class__</code>	

6.3.3 Class Variables

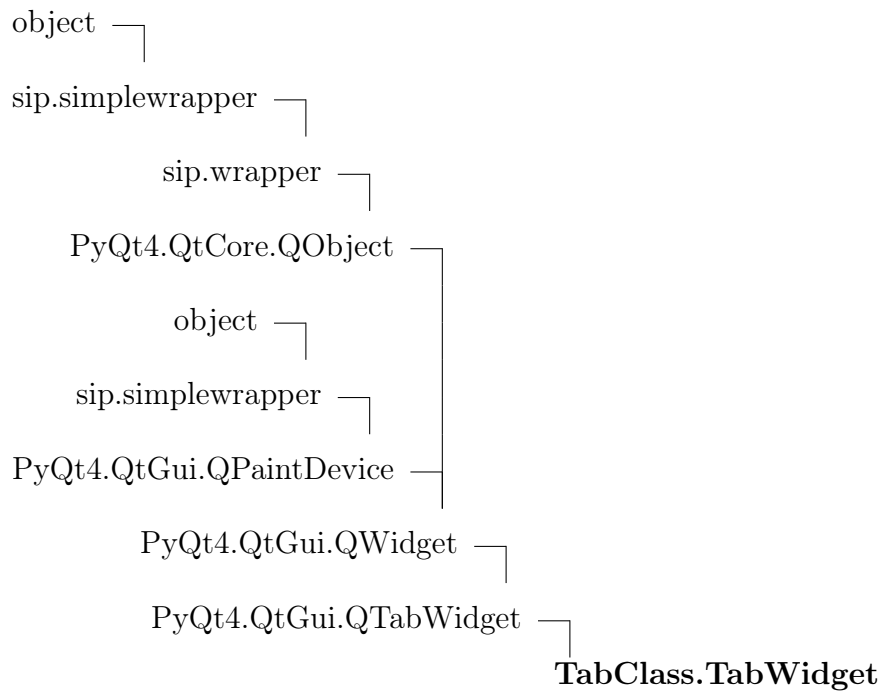
Name	Description
<i>Inherited from PyQt4.QtGui.QMainWindow</i>	
AllowNestedDocks, AllowTabbedDocks, AnimatedDocks, ForceTabbedDocks, VerticalTabs	
<i>Inherited from PyQt4.QtGui.QWidget</i>	
DrawChildren, DrawWindowBackground, IgnoreMask	
<i>Inherited from PyQt4.QtCore.QObject</i>	
staticMetaObject	
<i>Inherited from PyQt4.QtGui.QPaintDevice</i>	
PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM	

7 Module TabClass

7.1 Variables

Name	Description
<code>__package__</code>	Value: None

7.2 Class TabWidget



7.2.1 Methods

```
__init__(self, parent=None)
```

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

```
make_change_book_tab(self, vbox)
```

```
make_table_tab(self, vbox)
```

```
make_add_data_tab(self, grid)
```

```
make_get_data_tab(self, vbox_inner)
```

```
date_changed(self)
```

```
init_ui(self)
```

```
add_tab_item_clicked(self)
```

```
change_item_clicked(self)
```

```
validate_fields(self, book)
```

```
contains_digit(string)
```

```
values_has_changed(new_book, real_book)
```

```
set_list_item_selected(in_list, t)
```

```
change_selected(self)
```

```
delete_selected(self)
```

```
submit_changes(self)
```

```
submit_book(self)
```

```
button_clicked(self)
```

*Inherited from **PyQt4.QtGui.QTabWidget***

`__len__()`, `addTab()`, `changeEvent()`, `clear()`, `cornerWidget()`, `count()`, `currentChanged()`, `currentIndex()`, `currentWidget()`, `documentMode()`, `elideMode()`, `event()`, `heightForWidth()`, `iconSize()`, `indexOf()`, `initStyleOption()`, `insertTab()`, `isMovable()`, `isTabEnabled()`, `keyPressEvent()`, `minimumSizeHint()`, `paintEvent()`, `removeTab()`, `resizeEvent()`, `setCornerWidget()`, `setCurrentIndex()`, `setCurrentWidget()`, `setDocumentMode()`, `setElideMode()`, `setIconSize()`, `setMovable()`, `setTabBar()`, `setTabEnabled()`, `setTabIcon()`, `setTab-`

Position(), setTabShape(), setTabText(), setTabToolTip(),
 setTabWhatsThis(), setTabsClosable(), setUsesScrollButtons(),
 showEvent(), sizeHint(), tabBar(), tabCloseRequested(), tabI-
 con(), tabInserted(), tabPosition(), tabRemoved(), tabShape(),
 tabText(), tabToolTip(), tabWhatsThis(), tabsClosable(),
 usesScrollButtons(), widget()

Inherited from PyQt4.QtGui.QWidget

acceptDrops(), accessibleDescription(), accessibleName(), ac-
 tionEvent(), actions(), activateWindow(), addAction(), ad-
 dActions(), adjustSize(), autoFillBackground(), background-
 Role(), baseSize(), childAt(), childrenRect(), childrenRegion(),
 clearFocus(), clearMask(), close(), closeEvent(), contents-
 Margins(), contentsRect(), contextMenuEvent(), contextMenuPol-
 icy(), create(), cursor(), customContextMenuRequested(),
 destroy(), devType(), dragEnterEvent(), dragLeaveEvent(),
 dragMoveEvent(), dropEvent(), effectiveWinId(), enabled-
 Change(), ensurePolished(), enterEvent(), find(), focusIn-
 Event(), focusNextChild(), focusNextPrevChild(), focusOutEvent(),
 focusPolicy(), focusPreviousChild(), focusProxy(), focusWid-
 get(), font(), fontChange(), fontInfo(), fontMetrics(), fore-
 groundRole(), frameGeometry(), frameSize(), geometry(),
 getContentsMargins(), grabGesture(), grabKeyboard(), grab-
 Mouse(), grabShortcut(), graphicsEffect(), graphicsProxy-
 Widget(), handle(), hasFocus(), hasMouseTracking(), height(),
 hide(), hideEvent(), inputContext(), inputMethodEvent(),
 inputMethodHints(), inputMethodQuery(), insertAction(),
 insertActions(), isActiveWindow(), isAncestorOf(), isEnabled(),
 isEnabledTo(), isEnabledToTLW(), isFullScreen(), isHidden(),
 isLeftToRight(), isMaximized(), isMinimized(), isModal(),
 isRightToLeft(), isTopLevel(), isVisible(), isVisibleTo(), isWin-
 dow(), isWindowModified(), keyReleaseEvent(), keyboard-
 Grabber(), languageChange(), layout(), layoutDirection(),
 leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(),
 mapFromParent(), mapTo(), mapToGlobal(), mapToParent(),
 mask(), maximumHeight(), maximumSize(), maximumWidth(),
 metric(), minimumHeight(), minimumSize(), minimumWidth(),
 mouseDoubleClickEvent(), mouseGrabber(), mouseMoveEvent(),
 mousePressEvent(), mouseReleaseEvent(), move(), moveEvent(),
 nativeParentWidget(), nextInFocusChain(), normalGeome-
 try(), overrideWindowFlags(), overrideWindowState(), paintEngine(),
 palette(), paletteChange(), parentWidget(), pos(), previous-
 InFocusChain(), raise_(), rect(), releaseKeyboard(), release-
 Mouse(), releaseShortcut(), removeAction(), render(), re-
 paint(), resetInputContext(), resize(), restoreGeometry(), save-
 Geometry(), scroll(), setAcceptDrops(), setAccessibleDescrip-

tion(), setAccessibleName(), setAttribute(), setAutoFillBack-
 ground(), setBackgroundRole(), setBaseSize(), setContents-
 Margins(), setContextMenuPolicy(), setCursor(), setDisabled(),
 setEnabled(), setFixedHeight(), setFixedSize(), setFixedWidth(),
 setFocus(), setFocusPolicy(), setFocusProxy(), setFont(), set-
 ForegroundRole(), setGeometry(), setGraphicsEffect(), setHid-
 den(), setInputContext(), setInputMethodHints(), setLay-
 out(), setLayoutDirection(), setLocale(), setMask(), setMax-
 imumHeight(), setMaximumSize(), setMaximumWidth(), set-
 MinimumHeight(), setMinimumSize(), setMinimumWidth(),
 setMouseTracking(), setPalette(), setParent(), setShortcu-
 tAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIn-
 crement(), setSizePolicy(), setStatusTip(), setStyle(), set-
 StyleSheet(), setTabOrder(), setToolTip(), setUpdatesEn-
 abled(), setVisible(), setWhatsThis(), setWindowFilePath(),
 setWindowFlags(), setWindowIcon(), setWindowIconText(),
 setWindowModality(), setWindowModified(), setWindowOpac-
 ity(), setWindowRole(), setWindowState(), setWindowTitle(),
 show(), showFullScreen(), showMaximized(), showMin-
 imized(), showNormal(), size(), sizeIncrement(), sizePolicy(),
 stackUnder(), statusTip(), style(), styleSheet(), tabletEvent(),
 testAttribute(), toolTip(), topLevelWidget(), underMouse(),
 ungrabGesture(), unsetCursor(), unsetLayoutDirection(), un-
 setLocale(), update(), updateGeometry(), updateMicroFo-
 cus(), updatesEnabled(), visibleRegion(), whatsThis(), wheelEvent(),
 width(), winId(), window(), windowActivationChange(), win-
 dowFilePath(), windowFlags(), windowIcon(), windowIcon-
 Text(), windowModality(), windowOpacity(), windowRole(),
 windowState(), windowTitle(), windowType(), x(), x11Info(),
 x11PictureHandle(), y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), con-
 nect(), connectNotify(), customEvent(), deleteLater(), de-
 stroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(),
 dumpObjectTree(), dynamicPropertyNames(), emit(), event-
 Filter(), findChild(), findChildren(), inherits(), installEvent-
 Filter(), isWidgetType(), killTimer(), metaObject(), move-
 ToThread(), objectName(), parent(), property(), pyqtCon-
 figure(), receivers(), removeEventFilter(), sender(), sender-
 SignalIndex(), setObjectName(), setProperty(), signalsBlocked(),
 startTimer(), thread(), timerEvent(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalD-
 piY(), numColors(), paintingActive(), physicalDpiX(), phys-

icalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattr__(), __hash__(),
__reduce__(), __reduce_ex__(), __repr__(), __se-
tattr__(), __sizeof__(), __str__(), __subclasshook__()

7.2.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

7.2.3 Class Variables

Name	Description
<i>Inherited from PyQt4.QtGui.QTabWidget</i>	
East, North, Rounded, South, Triangular, West	
<i>Inherited from PyQt4.QtGui.QWidget</i>	
DrawChildren, DrawWindowBackground, IgnoreMask	
<i>Inherited from PyQt4.QtCore.QObject</i>	
staticMetaObject	
<i>Inherited from PyQt4.QtGui.QPaintDevice</i>	
PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM	

8 Module BookClass

8.1 Variables

Name	Description
<code>__package__</code>	Value: None

8.2 Class Book

8.2.1 Methods

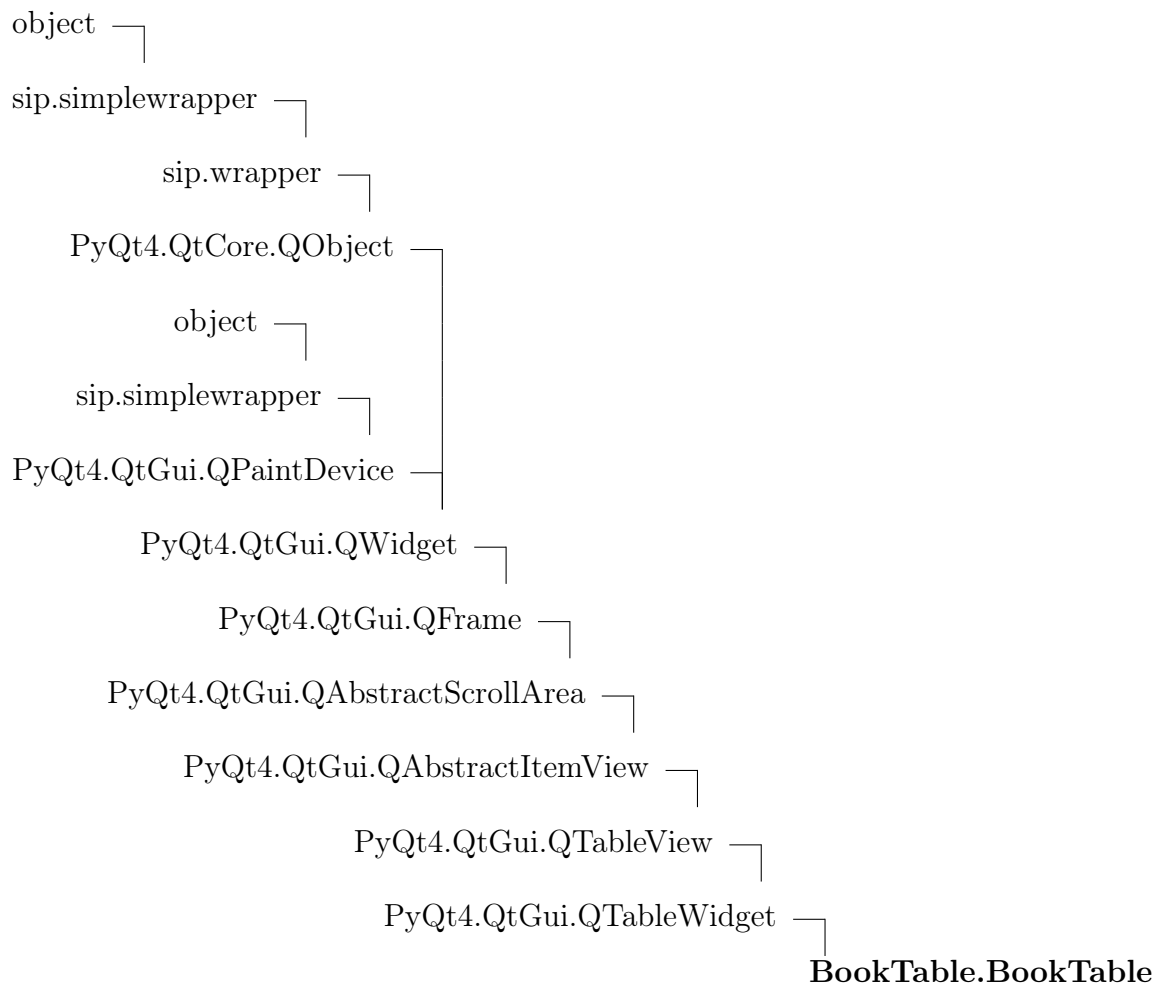
<code>__init__(self)</code>
<code>make_strings(self)</code>
<code>set_values(self, a, t, g, g2, d, gr, c)</code>
<code>__str__(self)</code>

9 Module BookTable

9.1 Variables

Name	Description
<code>__package__</code>	Value: None

9.2 Class BookTable



9.2.1 Methods

`__init__(self, data, *args)`

`x.__init__(...)` initializes `x`; see `help(type(x))` for signature

Overrides: `object.__init__` `exitit`(inherited documentation)

`set_data(self, data)`

`delete(self)`

`set_my_data(self)`

`get_selected_item(self)`

`get_selected_title(self)`

Inherited from `PyQt4.QtGui.QTableWidget`

`cellActivated()`, `cellChanged()`, `cellClicked()`, `cellDoubleClicked()`, `cellEntered()`, `cellPressed()`, `cellWidget()`, `clear()`, `clearContents()`, `closePersistentEditor()`, `column()`, `columnCount()`, `currentCellChanged()`, `currentColumn()`, `currentItem()`, `currentItemChanged()`, `currentRow()`, `dropEvent()`, `dropMimeData()`, `editItem()`, `event()`, `findItems()`, `horizontalHeaderItem()`, `indexFromItem()`, `insertColumn()`, `insertRow()`, `isItemSelected()`, `isSortingEnabled()`, `item()`, `itemActivated()`, `itemAt()`, `itemChanged()`, `itemClicked()`, `itemDoubleClicked()`, `itemEntered()`, `itemFromIndex()`, `itemPressed()`, `itemPrototype()`, `itemSelectionChanged()`, `items()`, `mimeData()`, `mimeTypes()`, `openPersistentEditor()`, `removeCellWidget()`, `removeColumn()`, `removeRow()`, `row()`, `rowCount()`, `scrollToItem()`, `selectedItems()`, `selectedRanges()`, `setCellWidget()`, `setColumnCount()`, `setCurrentCell()`, `setCurrentItem()`, `setHorizontalHeaderItem()`, `setHorizontalHeaderLabels()`, `setItem()`, `setItemPrototype()`, `setItemSelected()`, `setModel()`, `setRangeSelected()`, `setRowCount()`, `setSortingEnabled()`, `setVerticalHeaderItem()`, `setVerticalHeaderLabels()`, `sortItems()`, `supportedDropActions()`, `takeHorizontalHeaderItem()`, `takeItem()`, `takeVerticalHeaderItem()`, `verticalHeaderItem()`, `visualColumn()`, `visualItemRect()`, `visualRow()`

Inherited from `PyQt4.QtGui.QTableView`

clearSpans(), columnAt(), columnCountChanged(), columnMoved(), columnResized(), columnSpan(), columnViewportPosition(), columnWidth(), currentChanged(), gridStyle(), hideColumn(), hideRow(), horizontalHeader(), horizontalOffset(), horizontalScrollbarAction(), indexAt(), isColumnHidden(), isCornerButtonEnabled(), isIndexHidden(), isRowHidden(), moveCursor(), paintEvent(), resizeColumnToContents(), resizeColumnsToContents(), resizeRowToContents(), resizeRowsToContents(), rowAt(), rowCountChanged(), rowHeight(), rowMoved(), rowResized(), rowSpan(), rowViewportPosition(), scrollContentsBy(), scrollTo(), selectColumn(), selectRow(), selectedIndexes(), selectionChanged(), setColumnHidden(), setColumnWidth(), setCornerButtonEnabled(), setGridStyle(), setHorizontalHeader(), setRootIndex(), setRowHeight(), setRowHidden(), setSelection(), setSelectionModel(), setShowGrid(), setSpan(), setVerticalHeader(), setWordWrap(), showColumn(), showGrid(), showRow(), sizeHintForColumn(), sizeHintForRow(), sortByColumn(), timerEvent(), updateGeometries(), verticalHeader(), verticalOffset(), verticalScrollbarAction(), viewOptions(), visualRect(), visualRegionForSelection(), wordWrap()

Inherited from PyQt4.QtGui.QAbstractItemView

activated(), alternatingRowColors(), autoScrollMargin(), clearSelection(), clicked(), closeEditor(), commitData(), currentIndex(), dataChanged(), defaultDropAction(), dirtyRegionOffset(), doubleClicked(), dragDropMode(), dragDropOverwriteMode(), dragEnabled(), dragEnterEvent(), dragLeaveEvent(), dragMoveEvent(), dropIndicatorPosition(), edit(), editTriggers(), editorDestroyed(), entered(), executeDelayedItemsLayout(), focusInEvent(), focusNextPrevChild(), focusOutEvent(), hasAutoScroll(), horizontalScrollMode(), horizontalScrollbarValueChanged(), horizontalStepsPerItem(), iconSize(), indexWidget(), inputMethodEvent(), inputMethodQuery(), itemDelegate(), itemDelegateForColumn(), itemDelegateForRow(), keyPressEvent(), keyboardSearch(), model(), mouseDoubleClickEvent(), mouseMoveEvent(), mousePressEvent(), mouseReleaseEvent(), pressed(), reset(), resizeEvent(), rootIndex(), rowsAboutToBeRemoved(), rowsInserted(), scheduleDelayedItemsLayout(), scrollDirtyRegion(), scrollToBottom(), scrollToTop(), selectAll(), selectionBehavior(), selectionCommand(), selectionMode(), selectionModel(), setAlternatingRowColors(), setAutoScroll(), setAutoScrollMargin(), setCurrentIndex(), setDefaultDropAction(), setDirtyRegion(), setDragDropMode(), setDragDropOverwriteMode(), setDragEnabled(), setDropIndicatorShown(), setEditTriggers(), setHorizontalScrollMode(),

setHorizontalStepsPerItem(), setIconSize(), setIndexWidget(), setItemDelegate(), setItemDelegateForColumn(), setItemDelegateForRow(), setSelectionBehavior(), setSelectionMode(), setState(), setTabKeyNavigation(), setTextElideMode(), setVerticalScrollMode(), setVerticalStepsPerItem(), showDropIndicator(), sizeHintForIndex(), startDrag(), state(), tabKeyNavigation(), textElideMode(), update(), updateEditorData(), updateEditorGeometries(), verticalScrollMode(), verticalScrollbarValueChanged(), verticalStepsPerItem(), viewportEntered(), viewportEvent()

Inherited from PyQt4.QtGui.QAbstractScrollArea

addScrollBarWidget(), contextMenuEvent(), cornerWidget(), horizontalScrollBar(), horizontalScrollBarPolicy(), maximumViewportSize(), minimumSizeHint(), scrollbarWidgets(), setCornerWidget(), setHorizontalScrollBar(), setHorizontalScrollBarPolicy(), setVerticalScrollBar(), setVerticalScrollBarPolicy(), setViewport(), setViewportMargins(), setupViewport(), sizeHint(), verticalScrollBar(), verticalScrollBarPolicy(), viewport(), wheelEvent()

Inherited from PyQt4.QtGui.QFrame

changeEvent(), drawFrame(), frameRect(), frameShadow(), frameShape(), frameStyle(), frameWidth(), lineWidth(), midLineWidth(), setFrameRect(), setFrameShadow(), setFrameShape(), setFrameStyle(), setLineWidth(), setMidLineWidth()

Inherited from PyQt4.QtGui.QWidget

acceptDrops(), accessibleDescription(), accessibleName(), actionEvent(), actions(), activateWindow(), addAction(), addActions(), adjustSize(), autoFillBackground(), backgroundRole(), baseSize(), childAt(), childrenRect(), childrenRegion(), clearFocus(), clearMask(), close(), closeEvent(), contentsMargins(), contentsRect(), contextMenuPolicy(), create(), cursor(), customContextMenuRequested(), destroy(), devType(), effectiveWinId(), enabledChange(), ensurePolished(), enterEvent(), find(), focusNextChild(), focusPolicy(), focusPreviousChild(), focusProxy(), focusWidget(), font(), fontChange(), fontInfo(), fontMetrics(), foregroundRole(), frameGeometry(), frameSize(), geometry(), getContentsMargins(), grabGesture(), grabKeyboard(), grabMouse(), grabShortcut(), graphicsEffect(), graphicsProxyWidget(), handle(), hasFocus(), hasMouseTracking(), height(), heightForWidth(), hide(), hideEvent(), inputContext(), inputMethodHints(), insertAction(), insertActions(), isActiveWindow(), isAncestorOf(), isEnabled(),

isEnabledTo(), isEnabledToTLW(), isFullScreen(), isHidden(),
 isLeftToRight(), isMaximized(), isMinimized(), isModal(),
 isRightToLeft(), isTopLevel(), isVisible(), isVisibleTo(), isWin-
 dows(), isWindowModified(), keyReleaseEvent(), keyboard-
 Grabber(), languageChange(), layout(), layoutDirection(),
 leaveEvent(), locale(), lower(), mapFrom(), mapFromGlobal(),
 mapFromParent(), mapTo(), mapToGlobal(), mapToParent(),
 mask(), maximumHeight(), maximumSize(), maximumWidth(),
 metric(), minimumHeight(), minimumSize(), minimumWidth(),
 mouseGrabber(), move(), moveEvent(), nativeParentWid-
 get(), nextInFocusChain(), normalGeometry(), overrideWin-
 dowsFlags(), overrideWindowState(), paintEngine(), palette(),
 paletteChange(), parentWidget(), pos(), previousInFocusChain(),
 raise_(), rect(), releaseKeyboard(), releaseMouse(), release-
 Shortcut(), removeAction(), render(), repaint(), resetInput-
 Context(), resize(), restoreGeometry(), saveGeometry(), scroll(),
 setAcceptDrops(), setAccessibleDescription(), setAccessible-
 Name(), setAttribute(), setAutoFillBackground(), setBack-
 groundRole(), setBaseSize(), setContentsMargins(), setCon-
 textMenuPolicy(), setCursor(), setDisabled(), setEnabled(),
 setFixedHeight(), setFixedSize(), setFixedWidth(), setFo-
 cus(), setFocusPolicy(), setFocusProxy(), setFont(), setFore-
 groundRole(), setGeometry(), setGraphicsEffect(), setHid-
 den(), setInputContext(), setInputMethodHints(), setLay-
 out(), setLayoutDirection(), setLocale(), setMask(), setMax-
 imumHeight(), setMaximumSize(), setMaximumWidth(), set-
 MinimumHeight(), setMinimumSize(), setMinimumWidth(),
 setMouseTracking(), setPalette(), setParent(), setShortcu-
 tAutoRepeat(), setShortcutEnabled(), setShown(), setSizeIn-
 crement(), setSizePolicy(), setStatusTip(), setStyle(), set-
 StyleSheet(), setTabOrder(), setToolTip(), setUpdatesEn-
 abled(), setVisible(), setWhatsThis(), setWindowFilePath(),
 setWindowFlags(), setWindowIcon(), setWindowIconText(),
 setWindowModality(), setWindowModified(), setWindowOpac-
 ity(), setWindowRole(), setWindowState(), setWindowTi-
 tle(), show(), showEvent(), showFullScreen(), showMaximized(),
 showMinimized(), showNormal(), size(), sizeIncrement(), sizePol-
 icy(), stackUnder(), statusTip(), style(), styleSheet(), tabletEvent(),
 testAttribute(), toolTip(), topLevelWidget(), underMouse(),
 ungrabGesture(), unsetCursor(), unsetLayoutDirection(), un-
 setLocale(), updateGeometry(), updateMicroFocus(), update-
 sEnabled(), visibleRegion(), whatsThis(), width(), winId(),
 window(), windowActivationChange(), windowFilePath(), win-
 dowsFlags(), windowIcon(), windowIconText(), windowModal-
 ity(), windowOpacity(), windowRole(), windowState(), win-
 dowsTitle(), windowType(), x(), x11Info(), x11PictureHandle(),

y()

Inherited from PyQt4.QtCore.QObject

__getattr__(), blockSignals(), childEvent(), children(), connect(), connectNotify(), customEvent(), deleteLater(), destroyed(), disconnect(), disconnectNotify(), dumpObjectInfo(), dumpObjectTree(), dynamicPropertyNames(), emit(), eventFilter(), findChild(), findChildren(), inherits(), installEventFilter(), isWidgetType(), killTimer(), metaObject(), moveToThread(), objectName(), parent(), property(), pyqtConfigure(), receivers(), removeEventFilter(), sender(), senderSignalIndex(), setObjectName(), setProperty(), signalsBlocked(), startTimer(), thread(), tr(), trUtf8()

Inherited from PyQt4.QtGui.QPaintDevice

colorCount(), depth(), heightMM(), logicalDpiX(), logicalDpiY(), numColors(), paintingActive(), physicalDpiX(), physicalDpiY(), widthMM()

Inherited from sip.simplewrapper

__new__()

Inherited from object

__delattr__(), __format__(), __getattribute__(), __hash__(), __reduce__(), __reduce_ex__(), __repr__(), __setattr__(), __sizeof__(), __str__(), __subclasshook__()

9.2.2 Properties

Name	Description
<i>Inherited from object</i>	
__class__	

9.2.3 Class Variables

Name	Description
<i>Inherited from PyQt4.QtGui.QAbstractItemView</i>	

continued on next page

Name	Description
	AboveItem, AllEditTriggers, AnimatingState, AnyKeyPressed, BelowItem, CollapsingState, ContiguousSelection, CurrentChanged, DoubleClicked, DragDrop, DragOnly, DragSelectingState, DraggingState, DropOnly, EditKeyPressed, EditingState, EnsureVisible, ExpandingState, ExtendedSelection, InternalMove, MoveDown, MoveEnd, MoveHome, MoveLeft, MoveNext, MovePageDown, MovePageUp, MovePrevious, MoveRight, MoveUp, MultiSelection, NoDragDrop, NoEditTriggers, NoSelection, NoState, OnItem, OnViewport, PositionAtBottom, PositionAtCenter, PositionAtTop, ScrollPerItem, ScrollPerPixel, SelectColumns, SelectItems, SelectRows, SelectedClicked, SingleSelection
	<i>Inherited from PyQt4.QtGui.QFrame</i> Box, HLine, NoFrame, Panel, Plain, Raised, Shadow_Mask, Shape_Mask, StyledPanel, Sunken, VLine, WinPanel
	<i>Inherited from PyQt4.QtGui.QWidget</i> DrawChildren, DrawWindowBackground, IgnoreMask
	<i>Inherited from PyQt4.QtCore.QObject</i> staticMetaObject
	<i>Inherited from PyQt4.QtGui.QPaintDevice</i> PdmDepth, PdmDpiX, PdmDpiY, PdmHeight, PdmHeightMM, PdmNumColors, PdmPhysicalDpiX, PdmPhysicalDpiY, PdmWidth, PdmWidthMM