

# 42组-Qt作业报告

李佳伦

孙非寒

赵易帆

2024年6月30日



北京大学  
PEKING UNIVERSITY

# 目录

## CONTENTS



北京大学  
PEKING UNIVERSITY

01

功能介绍

02

类设计细节

03

成员分工

04

总结反思



# 01 功能介绍

## 产品效果截图:



### 1. PDF文件的打开与查看

### 2. 翻页功能

页码框可以显示页码；点击按钮“上一页”、“下一页”可以实现翻页；在页码框输入页码号可以实现页面跳转。

### 3. 书签功能

点击书签按钮，并输入书签名可以添加当前页为书签，在相同页面再次点击书签按钮会取消当前页面的书签，点击书签栏可以查看已有书签，点击书签栏中的书签名可以跳转至对应页面，每个PDF文件的书签相互独立，并且即使关闭PDF文件再下次打开时书签仍然存在。

### 4. 缩放功能

缩放框可以显示缩放比例，点击“放大”、“缩小”按钮可以实现缩放，下拉缩放框可以选择提供的几个常用缩放比例，还有一个“适应页面”按钮为用户提供自适应的最佳缩放比例。

### 5. 目录功能

自动识别PDF文件的目录并显示在左边目录栏，点击目录栏对应栏目可以跳转至相应页面。





## (1) 阅读器类

PDF\_reader类实现了PDF文件的打开、目录功能，以及和其他类的互动，是程序最核心的类。

①page\_selected槽函数：实现页码跳转 `page_selected(int page_number);`

②menu\_selected槽函数：实现目录跳转 `menu_selected(const QModelIndex &index);`

③bookmark\_selected槽函数：实现书签跳转 `bookmark_selected(const QModelIndex &index);`

④load\_bookmark\_view槽函数：实现书签显示 `load_bookmark_view();`

⑤open\_pdf\_file与on\_actionopen\_triggered槽函数：实现PDF文件的点击、显示、打开 `open_pdf_file(const QUrl &file_location);`  
`on_actionopen_triggered();`

```
class PDF_reader : public QMainWindow
{
    Q_OBJECT

public:
    PDF_reader(QWidget *parent = nullptr);
    ~PDF_reader();

public slots:
    void page_selected(int page_number);
    void menu_selected(const QModelIndex &index);
    void bookmark_selected(const QModelIndex &index);
    void load_bookmark_view();

private slots:
    void on_actionopen_triggered();
    void open_pdf_file(const QUrl &file_location);

    void on_actionprevious_page_triggered();

    void on_actionnext_page_triggered();

    void on_actiontest_window_triggered();

    void test_(int x);

    void on_actionzoom_in_triggered();

    void on_actionzoom_out_triggered();

    void on_actionmark_this_page_triggered();

private:
    Ui::PDF_reader *ui;

    QPdfDocument *document;
    QPdfPageSelector *page_selector;
    QFileDialog *file_selection = nullptr;
    zoom_selector *z_selector;
    QStringListModel bookmark_model;
    Bookmark_ManageTool bmk_tool;

    Test_Window_1 *f;
};
#endif // PDF_READER_H
```



⑥on\_actionprevious\_page\_triggered与on\_actionnext\_page\_triggered槽函数：实现PDF前后翻页

```
on_actionprevious_page_triggered();
```

```
on_actionnext_page_triggered();
```

⑦on\_actionzoom\_out\_triggered与on\_actionzoom\_in\_triggered槽函数：实现PDF缩放（1.1倍）

```
on_actionzoom_out_triggered();
```

```
on_actionzoom_in_triggered();
```

⑧on\_actionmark\_this\_page\_triggered槽函数：实现书签增删

```
on_actionmark_this_page_triggered();
```

## (2) 书签管理类

Book\_Mark是一个简单的书签类，用页码和书签名来唯一标识一个书签：

```
class Book_Mark{
public:
    int page;
    QString text;
    Book_Mark(){};
    Book_Mark(int x, QString s):page(x),text(s){}
};
```

```
class Bookmark_ManageTool
{
private:
    QString bookmark_file_path;
    QString currentpath = QDir::currentPath();
public:
    Bookmark_ManageTool();
    void set_file(QUrl file_URL);
    void load_bookmarks();
    void save();
    void add_bookmark(Book_Mark b);
    bool page_marked(int number);
    void remove_mark(int number);
    Book_Mark * bmk;
    int bookmark_count;
};
```

Bookmark\_ManageTool是书签管理类, 成员变量有: 书签文件存放的路径、当前程序的工作路径、书签指针、当前书签的数量; 书签管理类还有与书签相关的一系列操作函数:



## 书签管理操作函数：

①函数void set\_file：设置书签。

```
void set_file(QUrl file_URL);
```

②函数save：保存当前书签信息。

```
void save();
```

③函数load\_bookmarks：加载已保存的书签信息。

```
void load_bookmarks();
```

④函数add\_bookmark：添加新书签。

```
void add_bookmark(Book_Mark b);
```

⑤函数remove\_mark：移除当前页面的书签。

```
void remove_mark(int number);
```

⑥函数page\_marked：这是一个逻辑辅助函数，用于防止为同一页面添加重复书签的操作，具体实现：检查本页面是否已经被标记为书签，如果已经被标记过，返回true，否则返回false。

```
bool page_marked(int number);
```

```
class Bookmark_ManageTool
{
private:
    QString bookmark_file_path;
    QString currentpath = QDir::currentPath();

public:
    Bookmark_ManageTool();
    void set_file(QUrl file_URL);
    void load_bookmarks();
    void save();
    void add_bookmark(Book_Mark b);
    bool page_marked(int number);
    void remove_mark(int number);
    Book_Mark * bmk;
    int bookmark_count;
};
```



### (3) 缩放类

①构造函数：调用 `reset()` 方法将初始缩放比例设置为 100%，提供了几个常用的缩放选项（适应页面、20%、33%、50%、75%、100%、150%），连接信号和槽，用于处理用户选择的缩放比例变化。

②槽函数 `on_zoom_text_changed`：处理用户手动输入的缩放比例

```
void on_zoom_text_changed(const QString &text);
```

③槽函数 `set_zoom`：以百分数显示当前缩放比例

```
void set_zoom(qreal factor);
```

④信号 `zoom_modeChange` 与 `zoom_factorChange`：当前缩放模式/当前缩放比例已改变的信号。

```
void zoom_modeChange(QPdfView::ZoomMode mode);
```

```
void zoom_factorChange(qreal factor);
```

```
zoom_selector::zoom_selector(QWidget *parent)
: QComboBox(parent)
{
    setEditable(true);

    addItem(QString("适应页面"));
    addItem(QLatin1String("20%"));
    addItem(QLatin1String("33%"));
    addItem(QLatin1String("50%"));
    addItem(QLatin1String("75%"));
    addItem(QLatin1String("100%"));
    addItem(QLatin1String("150%"));

    connect(this, &QComboBox::currentTextChanged
            ,this, &zoom_selector::on_zoom_text_changed);
    connect(lineEdit(), &QLineEdit::editingFinished, this,
            [this]() {on_zoom_text_changed(lineEdit()->text()); });

    reset();
}
```

```
class zoom_selector: public QComboBox
{
    Q_OBJECT

public:
    zoom_selector(QWidget *parent = nullptr);

public slots:
    void set_zoom(qreal factor);
    void reset();
    void on_zoom_text_changed(const QString &text);

signals:
    void zoom_modeChange(QPdfView::ZoomMode mode);
    void zoom_factorChange(qreal factor);
};

#endif // ZOOM_SELECTOR_H
```



李佳伦：思路提出，视频录制，代码调试，统筹对接

孙非寒：PDF打开与查看，翻页功能，书签功能，缩放功能，目录功能

赵易帆：书签功能，ui优化：添加图标，报告制作，资源整理



1. 用户界面还不够美观，界面控件比较原始，还有进一步优化的空间。
2. 小组合作开发经验生疏，但经过本次作业得到了一定锻炼。
3. 小组从实际应用和生活经验出发，为PDF阅读器增添了几个核心实用功能，尤其是带记忆功能的书签和自适应页面的缩放功能，可以服务实际工作和生活的需要。

