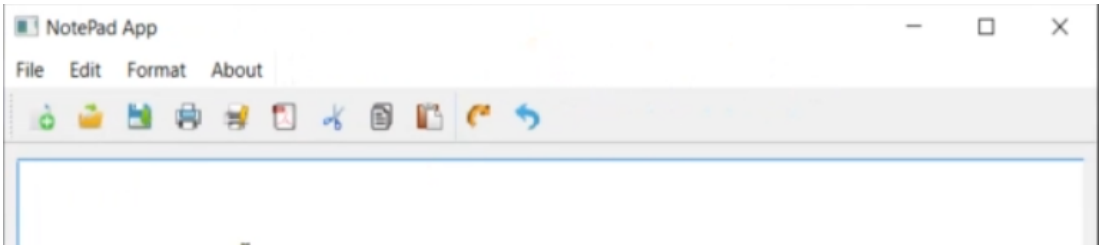
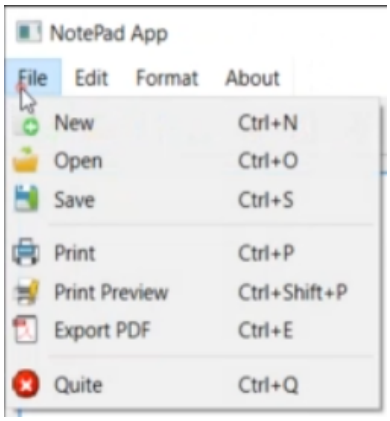
??????: notepad

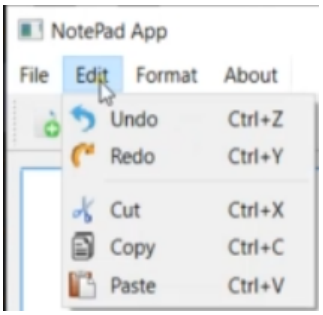
Начальная информация

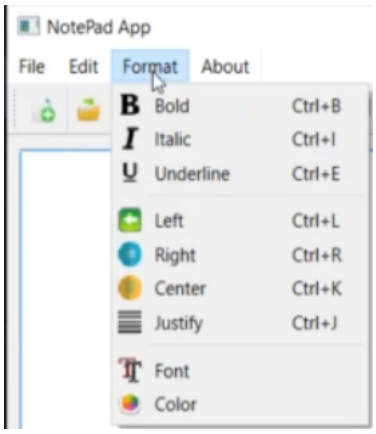
Внешний вид приложения:



Элементы интерфейса: меню, быстрые кнопки и многострочное поле ввода. Элементы меню:







Установленные модули:

```
pip install pyqt6
```

Параметры интерфейса:

Окно - MainWindow

```
from PyQt6.QtWidgets import QMainWindow, QApplication, QFileDialog, QMessageBox
from PyQt6.QtPrintSupport import QPrinter, QPrintDialog, QPrintPreviewDialog
from PyQt6.QtCore import QFileInfo
from PyQt6.QtGui import QFont
import sys
from notepadapp import Ui_MainWindow


class NotePadWindow(QMainWindow, Ui_MainWindow):
```

```
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.show()

        self.actionSave.triggered.connect(self.file_save)
        self.actionNew.triggered.connect(self.file_new)
        self.actionOpen.triggered.connect(self.file_open)
        self.actionPrint.triggered.connect(self.file_print)
        self.actionPrint_preview.triggered.connect(self.preview_dialog)
        self.actionExport_PDF.triggered.connect(self.exporting_pdf)
        self.actionQuit.triggered.connect(self.exit_app)

        self.actionUndo.triggered.connect(self.textEdit.undo)
        self.actionRedo.triggered.connect(self.textEdit.redo)

        self.actionCut.triggered.connect(self.textEdit.cut)
        self.actionCopy.triggered.connect(self.textEdit.copy)
        self.actionPaste.triggered.connect(self.textEdit.paste)

        self.actionBold.triggered.connect(self.text_bold)

    def maybe_save(self) -> bool:
        if not self.textEdit.document().isModified():
            return True

        ret = QMessageBox.warning(self, "Application",
                                  "The document changed \n Save working?",
                                  QMessageBox.StandardButton.Save |
                                  QMessageBox.StandardButton.Discard |
                                  QMessageBox.StandardButton.Cancel)
        if ret == QMessageBox.StandardButton.Save:
            self.file_save()
            return True
        elif ret == QMessageBox.StandardButton.Cancel:
            return False

        return True

    def file_new(self):
```

```
        if self.maybe_save():
            self.textEdit.clear()

    def file_save(self):
        filename = QFileDialog.getSaveFileName(self, 'Save file')
        if filename[0]:
            f = open(filename[0], 'w')
            with f:
                text = self.textEdit.toPlainText()
                f.write(text)
                QMessageBox.about(self, 'Save file', 'File saved successfully!')

    def file_open(self):
        self.maybe_save()
        filename = QFileDialog.getOpenFileName(self, 'Open file')
        if filename[0]:
            f = open(filename[0], 'r')
            with f:
                data = f.read()
                self.textEdit.setText(data)

    def file_print(self):
        printer = QPrinter(QPrinter.PrinterMode.HighResolution)
        dialog = QPrintDialog(printer)
        if dialog.exec() == QPrintDialog.DialogCode.Accepted:
            self.textEdit.print(printer)

    def print_preview(self, printer):
        self.textEdit.print(printer)

    def preview_dialog(self):
        printer = QPrinter(QPrinter.PrinterMode.HighResolution)
        preview_dialog = QPrintPreviewDialog(printer, self)
        preview_dialog.paintRequested.connect(self.print_preview)
        preview_dialog.exec()

    def exporting_pdf(self):
        #fn, _ = QFileDialog.getSaveFileName(self, 'Export PDF', "PDF Files (.pdf) ;;
AllFiles()")
        fn, _ = QFileDialog.getSaveFileName(self, 'Export PDF','',"PDF Files (.pdf) ;; All
```

```
Files (*)")
        if QFileInfo(fn).suffix() == "":
            fn += '.pdf'
        printer = QPrinter(QPrinter.PrinterMode.HighResolution)
        printer.setOutputFormat(QPrinter.OutputFormat.PdfFormat)
        printer.setOutputFileName(fn)
        self.textEdit.document().print(printer)

    def exit_app(self):
        self.close()

    def text_bold(self):
        font = QFont()
        font.setBold(True)
        self.textEdit.setFont(font)



app = QApplication(sys.argv)
Note = NotePadWindow()
sys.exit(app.exec())
```

Это неполный код, операции однотипные.

Revision #3
Created 9 February 2026 16:18:50 by Admin
Updated 11 February 2026 16:20:42 by Admin