

---

# Глава 1. Notepad

## Содержание

Краткое описание приложения .....	1
Скриншоты работы приложения .....	1
Пример разработанного кода приложения .....	2

## Краткое описание приложения

Приложение Notepad предназначено для редактирования текстовых файлов формата .txt

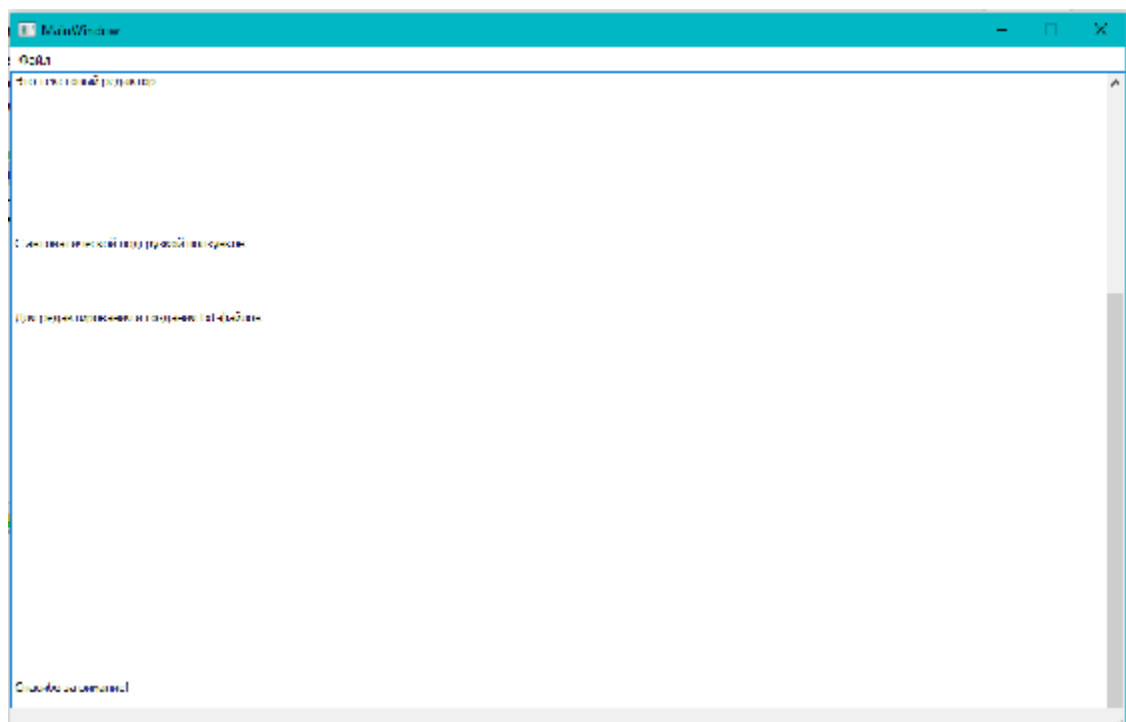
Интерфейс программы спроектирован таким образом, чтобы имитировать внешний вид блокнота в операционной системе Windows. Меню содержит один пункт "Файл" и три подпункта для открытия существующих txt-файлов и сохранения новых/измененных

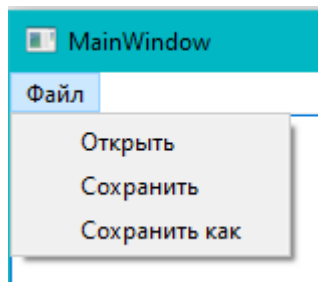
Пункт меню "Файл" содержит следующие операции:

1. Операция открытия существующего файла "Открыть"
2. Операция сохранения изменений в файле "Сохранить"
3. Операция сохранения изменений/нового содержимого в другом месте носителя "Сохранить как"

## Скриншоты работы приложения

Рисунок 1.1. Интерфейс пользователя



**Рисунок 1.2. Пункты меню**

## Пример разработанного кода приложения

Для разработки пользовательского интерфейса использовался QtDesigner для создания кроссплатформенного GUI. Пример разметки приведен ниже.

**Пример 1.1. Пример разметки пользовательского интерфейса**

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1024, 720)
        MainWindow.setMinimumSize(QtCore.QSize(800, 600))
        MainWindow.setMaximumSize(QtCore.QSize(3440, 1440))
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.plainTextEdit = QtWidgets.QPlainTextEdit(self.centralwidget)
        self.plainTextEdit.setGeometry(QtCore.QRect(0, 0, 1024, 720))
        self.plainTextEdit.setObjectName("plainTextEdit")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 1024, 21))
        self.menubar.setObjectName("menubar")
        self.menu = QtWidgets.QMenu(self.menubar)
        self.menu.setObjectName("menu")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)
        self.action_open = QtWidgets.QAction(MainWindow)
        self.action_open.setObjectName("action_open")
        self.action_save = QtWidgets.QAction(MainWindow)
        self.action_save.setObjectName("action_save")
        self.action_save_as = QtWidgets.QAction(MainWindow)
        self.action_save_as.setObjectName("action_save_as")
        self.menu.addAction(self.action_open)
        self.menu.addAction(self.action_save)
        self.menu.addAction(self.action_save_as)
        self.menubar.addAction(self.menu.menuAction())

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)
```

```
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.menu.setTitle(_translate("MainWindow", "Файл"))
    self.action_open.setText(_translate("MainWindow", "Открыть"))
    self.action_save.setText(_translate("MainWindow", "Сохранить"))
    self.action_save_as.setText(_translate("MainWindow", "Сохранить как"))

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

Для реализации работы с текстовыми файлами использовался язык программирования Python.

### **Пример 1.2. Листинг приложения**

```
from PyQt5 import QtWidgets
from ui.frame import Ui_MainWindow

class WndProc(QMainWindow):
    def __init__(self):
        super(WndProc, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.path = ''
        self.ui.action_open.connect(self.open_file)
        self.ui.action_save.connect(self.save_file)
        self.ui.action_save_as.connect(self.save_as_file)

    def open_file():
        fileName, _ = QtWidgets.QFileDialog.getOpenFileName(self, "Открыть", "")
        if fileName:
            self.path = fileName
            self.ui.plainTextEdit.setText(fileName)

    def save_file():
        if path != '':
            file = open(path, 'w').write(self.ui.plainTextEdit.getText())
        else:
            self.save_file_as()

    def save_file_as():
        fileName, _ = QtWidgets.QFileDialog.getSaveFileName(self, "Сохранить как", "")
        if fileName:
            open(fileName, 'w').write(self.ui.plainTextEdit.getText())

if __name__ == '__main__':
    app = QtWidgets.QApplication([])
```

```
application = WndProc()  
application.show()  
sys.exit(app.exec())
```

### **Примечание**

Код приложения находится в свободном доступе и может быть найден в репозитории на GitHub.

Notepad [<https://github.com/pi17v-ppsc/Baev>]