

# Проект Умные заметки

Выполнил:

Салимов Р. Р.  
8 «В» Класс

Руководитель

проекта:

учитель информатики  
Горяинова Елена Васильевна



# Цели моей работы

1

**Языки**

Изучить язык программирования

2

**Написание**

Написать программу «умные заметки»

3

**Запуск**

Запустить саму программу



## Beginner-Level JavaScript

## Coding Project Ideas

FREE GUIDE →

MIKKE GOES  
CODING

```
1 import Neo from '.....src/Neo.js';
2 import * as core from '.....src/core/import.js';
3
4 StartTest(t => {
5   t.it('Module imports', t => {
6     t.ok(Neo, 'Neo is imported as a JS module');
7     t.ok(core.Base, 'core.Base is imported as a JS module');
8   });
9
10   t.it('Class config', t => {
11     let valueA = 3;
12     valueB = 2;
13     instance;
14     t.diag('a=1, b=2');
15
16     class TestClass extends core.Base {
17       static getConfig() { return {
18         className: 'TestClass',
19         a: valueA,
20         b: valueB
21       }}
22
23       afterSet(value, oldValue) {
24         t.isStrict(value, valueA, 'afterSet: a equals ' + valueA);
25         t.isStrict(this.b, valueB, 'afterSet: b equals ' + valueB);
26       }
27
28       afterSet(value, oldValue) {
29         t.isStrict(this.a, valueA, 'afterSet: a equals ' + valueA);
30         t.isStrict(value, valueB, 'afterSet: b equals ' + valueB);
31       }
32
33       beforeSet(value, oldValue) {
34         t.isStrict(value, valueA, 'beforeSet: a equals ' + valueA);
35         t.isStrict(this.b, valueB, 'beforeSet: b equals ' + valueB);
36       }
37       return value;
38     }
39
40     beforeSet(value, oldValue) {
41       t.isStrict(this.a, valueA, 'beforeSet: a equals ' + valueA);
42       t.isStrict(value, valueB, 'beforeSet: b equals ' + valueB);
43     }
44     return value;
45   }
46 }
47
48 Neo.applyClassConfig[TestClass];
49 instance = Neo.create[TestClass];
50
51 t.diag('a=2');
52 valueB = 2;
53 instance.a = valueA;
54
55 t.diag('b=3');
56 valueB = 3;
57 instance.b = valueB;
58
59 t.diag('a=1, b=4');
60 valueA = 3;
61 valueB = 4;
62
63 instance.set({
64   a: valueA,
65   b: valueB
66 });
67
68 });
69
70 });
```

# Языки которые я выбрал:

01

## Python

1. Легкий язык
2. Легко читать код
3. Бесплатный и с открытым кодом
4. Большое количество модулей
5. Интерпретируемый
6. Язык высокого уровня

02

## JavaScript

1. Скорость работы и производительность
2. Простота и рациональность применения
3. Легкость освоения языка
4. Удобство пользовательских интерфейсов

```
1 #file search engine:
2
3 import os
4 fileType = '.mp4'
5
6 anyfile = os.listdir('desktop')
7 for file in anyfile:
8   if file.endswith(fileType):
9     print(file)
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

PYTHON: HOW TO  
FIND A FILE  
--ORDUM--

# Функционал

## программы:



**Возможность создавать заметки**

Быстрые и надежные заметки всегда под рукой, отредактируй в любое время



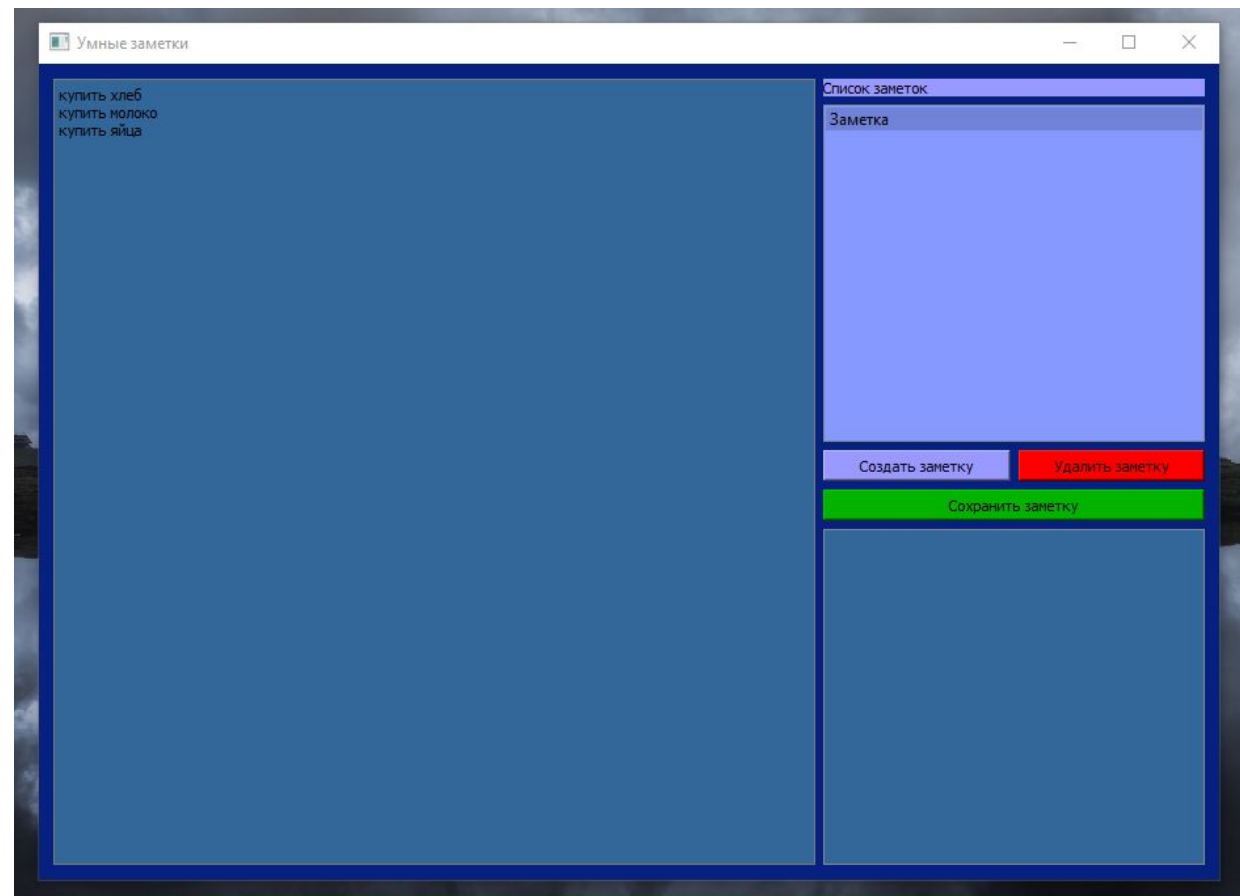
**Максимальная мобильность**

Программа быстрая и шустрая, мгновенно справляется с задачами



**Работа с Json файлами**

Работа не с обычными текстовыми файлами



# Фрагменты моего кода:

```
notes_txt.py
1  from PyQt5.QtCore import Qt
2
3  from PyQt5.QtWidgets import QApplication, QWidget,
4     QPushButton, QLabel, QListWidget,
5     QLineEdit, QTextEdit, QDialog, QHBoxLayout,
6     QVBoxLayout, QFormLayout
7
8
```

```
'''Интерфейс приложения'''
#параметры окна приложения
notes_win = QWidget()
notes_win.setWindowTitle('Умные заметки')
notes_win.resize(900, 600)
```

```
#виджеты окна приложения
```

```
list_notes = QListWidget()
```

```
list_notes.setStyleSheet('background: rgb(133, 153, 255);')
```

```
list_notes_label = QLabel('Список заметок')
```

```
list_notes_label.setStyleSheet('background: rgb(153, 153, 255);')
```

```
button_note_create = QPushButton('Создать заметку')
```

```
button_note_del = QPushButton('Удалить заметку')
```

```
button_note_save = QPushButton('Сохранить заметку')
```

```
button_note_save.setStyleSheet('background: rgb(4, 179, 0);')
```

```
button_note_del.setStyleSheet('background: rgb(255,0,0);')
```

```
button_note_create.setStyleSheet('background: rgb(153, 153, 255);')
```

```
field_tag = QLineEdit('')
```

```
field_tag.setPlaceholderText('Введите тег...')
```

```
field_text = QTextEdit()
```

```
field_text.setStyleSheet('background: rgb(51, 102, 153);')
```

```
while True:
    filename = str(name)+".txt"
    try:
        with open(filename, "r") as file:
            for line in file:
                line = line.replace('\n', '')
                note.append(line)
            tags = note[2].split(' ')
            note[2] = tags

        notes.append(note)

        note = []

        name += 1
```

```
def show_note():
    key = list_notes.selectedItems()[0].text()
    print(key)

    for note in notes:
        if note[0] == key:
            field_text.setText(note[1])
            list_tags.clear()
            list_tags.addItem(note[2])
```

# Итоги опроса



70%

**Минималистичный дизайн**  
Более 70% людей понравился интерфейс программы



50%

**Работа программы**  
Более 50% людей понравилась функционал приложения



80%

**Быстродействие**  
Достаточно быстро справляется со всеми задачами





# В ходе работы



# Почему вам стоит попробовать **Умные**

**заметки** 3 важных фактора после которого вам стоит попробовать умные заметки

01

## Сохранен ия

Возможность  
написание и  
сохранения в  
любой момент

02

## Редактиру й

Отредактируй,  
если что то не  
понравилось

03

## Память

Никогда не  
забудешь важную  
информацию

**спасибо**  
**за внимание**

