Listings

Listing 1: 这是个生成器

```python
from pathlib import Path
import json
import argparse

parser = argparse.ArgumentParser(description="Convert␣code␣in␣current␣directory␣to␣a␣latex␣document.")
parser.add_argument('output', help="output␣path")

args = parser.parse_args()

output_path = args.output

out_file = open(output_path, 'w', encoding='utf-8')

template_str = """
\\lstinputlisting[language={0}{2}]{{{1}}}
"""

template_param_str = ",{0}={1}"

begin_code = """\
\\documentclass{report}
\\usepackage{geometry}
\\geometry{a4paper, scale=0.8}
\\usepackage{xeCJK}
\\usepackage{listings}
\\lstset{
    numbers=left,
    frame=single,
    caption=\\lstname,
    breaklines=true
}
\\begin{document}
\\lstlistoflistings

\\newpage

"""
end_code = """\
\end{document}
"""

ext_filters = {'.c': 'c', '.cpp': 'c++', '.h': 'c++', '.hpp': 'c++', '.java': 'java', '.py': 'python'}

out_file.write(begin_code)

p = Path('.')
filenames = list(p.rglob('*'))
for f in filenames:
    fs = str(f)
    ext = fs[fs.rfind('.'):]
    if (ext not in ext_filters):
        continue
    param_str = ""
    if (Path(fs + '.json').is_file()):
        with open(fs + '.json', 'r', encoding='utf-8') as config_file:
            file_content = config_file.read()
            params = json.loads(file_content)
            for k in params:
                param_str += template_param_str.format(k, params[k])
    fs = fs.replace('\\', '/')
    out_file.write(template_str.format(ext_filters[ext], fs, param_str))

out_file.write(end_code)

out_file.close()
```

Listing 2: cpp–code/huffman copy.cpp

```
#include <iostream>

// just for test

struct Huffman
{
    // ...
    int a = 0, b = 0, c = 0;
};

int main()
{
    std::cout << Huffman().a;
    return 0;
}
```

Listing 3: Huffman Tree

```
#include <iostream>

// just for test

struct Huffman
{
    // ...
    int a = 0, b = 0, c = 0;
};

int main()
{
    std::cout << Huffman().a;
    return 0;
}
```