Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Кафедра «Компьютерная безопасность»

Лабораторная работа №5

Вариант 25

по дисциплине

«Методы программирования»

Работу выполнил
студент группы СКБ-201 _____ А.Н. Ушаков
подпись, дата

Работу проверил _____ С.А. Сластников
подпись, дата

Москва, 2023

Hierarchical Index

Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Class Index

Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

File Index

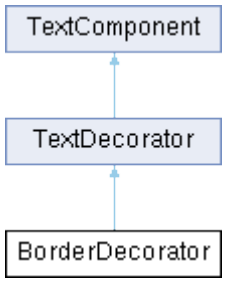
File List

Here is a list of all files with brief descriptions:

Class Documentation

BorderDecorator Class Reference

Inheritance diagram for BorderDecorator:



Public Member Functions

- **BorderDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override

Public Member Functions inherited from TextDecorator

- **TextDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override
- virtual string **getText** (const std::string &filename)=0

Additional Inherited Members

Protected Attributes inherited from TextDecorator

- **TextComponent * component_**

Detailed Description

Definition at line **41** of file **ConsoleApplication11.cpp**.

Constructor & Destructor Documentation

BorderDecorator::BorderDecorator (TextComponent * *component*) `[inline]`

Definition at line **43** of file **ConsoleApplication11.cpp**.

Member Function Documentation

string BorderDecorator::getText (const std::string & *filename*) `[inline]`, `[override]`, `[virtual]`

Reimplemented from **TextDecorator** (*p.11*).

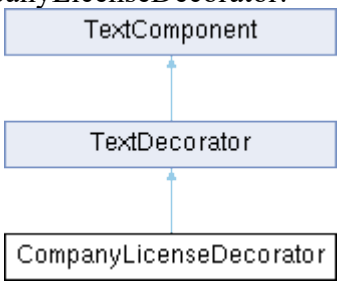
Definition at line **44** of file **ConsoleApplication11.cpp**.

The documentation for this class was generated from the following file:

ConsoleApplication11.cpp

CompanyLicenseDecorator Class Reference

Inheritance diagram for CompanyLicenseDecorator:



Public Member Functions

- **CompanyLicenseDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override

Public Member Functions inherited from TextDecorator

- **TextDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override
- virtual string **getText** (const std::string &filename)=0

Additional Inherited Members

Protected Attributes inherited from TextDecorator

- **TextComponent * component_**

Detailed Description

Definition at line **79** of file **ConsoleApplication11.cpp**.

Constructor & Destructor Documentation

CompanyLicenseDecorator::CompanyLicenseDecorator (TextComponent * *component*) `[inline]`

Definition at line **81** of file **ConsoleApplication11.cpp**.

Member Function Documentation

string CompanyLicenseDecorator::getText (const std::string & *filename*) `[inline]`, `[override]`, `[virtual]`

Reimplemented from **TextDecorator** (*p.11*).

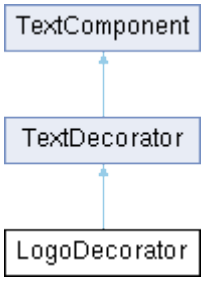
Definition at line **82** of file **ConsoleApplication11.cpp**.

The documentation for this class was generated from the following file:

ConsoleApplication11.cpp

LogoDecorator Class Reference

Inheritance diagram for LogoDecorator:



Public Member Functions

- **LogoDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override

Public Member Functions inherited from TextDecorator

- **TextDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override
- virtual string **getText** (const std::string &filename)=0

Additional Inherited Members

Protected Attributes inherited from TextDecorator

- **TextComponent * component_**

Detailed Description

Definition at line **98** of file **ConsoleApplication11.cpp**.

Constructor & Destructor Documentation

LogoDecorator::LogoDecorator (TextComponent * *component*) [inline]

Definition at line **100** of file **ConsoleApplication11.cpp**.

Member Function Documentation

string LogoDecorator::getText (const std::string & *filename*) [inline], [override], [virtual]

Reimplemented from **TextDecorator** (*p.11*).

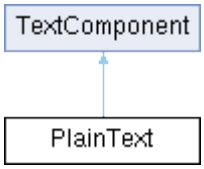
Definition at line **101** of file **ConsoleApplication11.cpp**.

The documentation for this class was generated from the following file:

ConsoleApplication11.cpp

PlainText Class Reference

Inheritance diagram for PlainText:



Public Member Functions

- string **getText** (const std::string &filename) override
- virtual string **getText** (const std::string &filename)=0

Detailed Description

Definition at line **14** of file **ConsoleApplication11.cpp**.

Member Function Documentation

string PlainText::getText (const std::string & ***filename*****)** `[inline], [override], [virtual]`

Implements **TextComponent** (*p.10*).

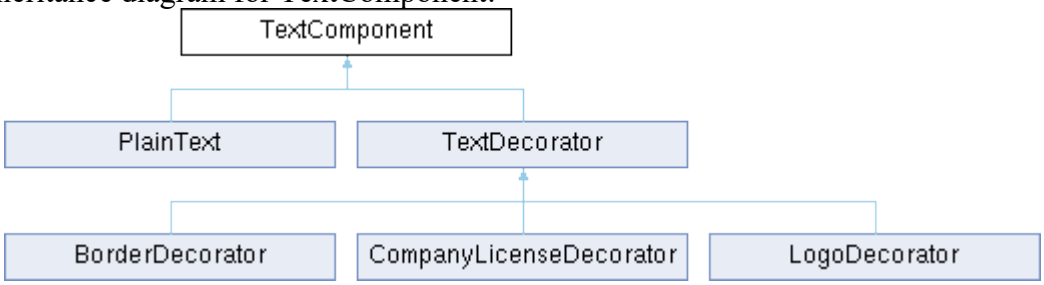
Definition at line **16** of file **ConsoleApplication11.cpp**.

The documentation for this class was generated from the following file:

ConsoleApplication11.cpp

TextComponent Class Reference

Inheritance diagram for TextComponent:



Public Member Functions

- virtual string **getText** (const std::string &filename)=0

Detailed Description

Definition at line **9** of file **ConsoleApplication11.cpp**.

Member Function Documentation

virtual string TextComponent::getText (const std::string & *filename*) `[pure virtual]`

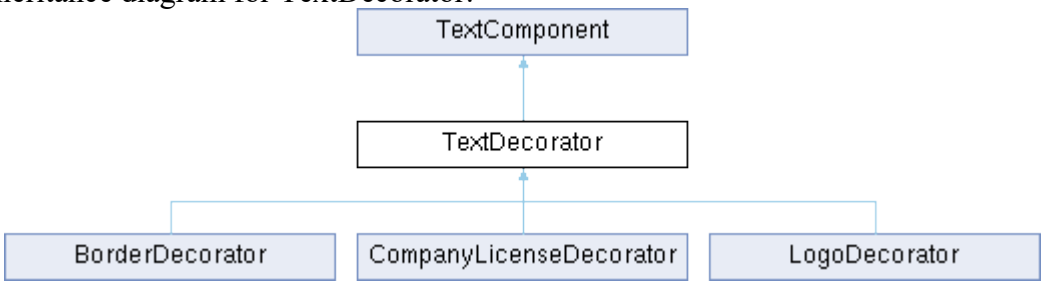
Implemented in **PlainText** (*p.9*), **TextDecorator** (*p.11*), **BorderDecorator** (*p.4*), **CompanyLicenseDecorator** (*p.6*), and **LogoDecorator** (*p.8*).

The documentation for this class was generated from the following file:

ConsoleApplication11.cpp

TextDecorator Class Reference

Inheritance diagram for TextDecorator:



Public Member Functions

- **TextDecorator** (**TextComponent** *component)
- string **getText** (const std::string &filename) override
- virtual string **getText** (const std::string &filename)=0

Protected Attributes

- **TextComponent * component_**

Detailed Description

Definition at line **30** of file **ConsoleApplication11.cpp**.

Constructor & Destructor Documentation

TextDecorator::TextDecorator (TextComponent * *component*) `[inline]`

Definition at line **34** of file **ConsoleApplication11.cpp**.

Member Function Documentation

string TextDecorator::getText (const std::string & *filename*) `[inline]`, `[override]`, `[virtual]`

Implements **TextComponent** (*p.10*).

Reimplemented in **BorderDecorator** (*p.4*), **CompanyLicenseDecorator** (*p.6*), and **LogoDecorator** (*p.8*).

Definition at line **36** of file **ConsoleApplication11.cpp**.

Member Data Documentation

TextComponent* TextDecorator::component_ `[protected]`

Definition at line **32** of file **ConsoleApplication11.cpp**.

The documentation for this class was generated from the following file:
ConsoleApplication11.cpp

File Documentation

ConsoleApplication11.cpp File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
#include <sstream>
```

Classes

- class **TextComponent**class **PlainText**
- class **TextDecorator**
- class **BorderDecorator**
- class **CompanyLicenseDecorator**
- class **LogoDecorator**

Functions

- int **main** ()

Function Documentation

int main ()

Definition at line **110** of file **ConsoleApplication11.cpp**.

ConsoleApplication11.cpp

Go to the documentation of this file.

```
00001 #include <iostream>
00002 #include <string>
00003 #include <fstream>
00004 #include <vector>
00005 #include <sstream>
00006
00007 using namespace std;
00008
00009 class TextComponent {
00010 public:
00011     virtual string getText(const std::string& filename) = 0;
00012 };
00013
00014 class PlainText : public TextComponent {
00015 public:
00016     string getText(const std::string& filename) override {
00017         std::ifstream file(filename);
00018         if (file) {
00019             std::string content((std::istreambuf_iterator<char>(file)),
00020                 (std::istreambuf_iterator<char>()));
00021             return content;
00022         }
00023         else {
00024             std::cerr << "Error opening file: " << filename << std::endl;
00025             return "";
00026         }
00027     }
00028 };
00029
00030 class TextDecorator : public TextComponent {
00031 protected:
00032     TextComponent* component_;
00033 public:
00034     TextDecorator(TextComponent* component) : component_(component) {}
00035
00036     string getText(const std::string& filename) override {
00037         return component_->getText(filename);
00038     }
00039 };
00040
00041 class BorderDecorator : public TextDecorator {
00042 public:
00043     BorderDecorator(TextComponent* component) : TextDecorator(component) {}
00044     string getText(const std::string& filename) override {
00045         string str = component_->getText(filename);
00046         istringstream iss(str);
00047         vector<std::string> lines;
00048         string line;
00049         while (std::getline(iss, line)) {
00050             lines.push_back(line);
00051         }
00052         size_t max_length = 0;
00053         for (const auto& line : lines) {
00054             if (line.size() > max_length) {
00055                 max_length = line.size();
00056             }
00057         }
00058         string framed_content = "+";
00059         for (size_t i = 0; i < max_length + 2; i++) {
00060             framed_content += "-";
00061         }
00062         framed_content += "+\n";
00063         for (const auto& line : lines) {
00064             framed_content += "| " + line;
00065             for (size_t i = line.size(); i < max_length; i++) {
00066                 framed_content += " ";
00067             }
00068             framed_content += " |\n";
00069         }
00070         framed_content += "+";
00071         for (size_t i = 0; i < max_length + 2; i++) {
00072             framed_content += "-";
00073         }
```

```
00074         framed_content += "+\n";
00075         return framed_content;
00076     }
00077 };
00078
00079 class CompanyLicenseDecorator : public TextDecorator {
00080 public:
00081     CompanyLicenseDecorator(TextComponent* component) : TextDecorator(component)
{}
00082     string getText(const std::string& filename) override {
00083         string utopia_license =
00084
"//===----------------------------------------------------------------------===//\n"
00085             "//\n"
00086             "// Part of the Utopia EDA Project, under the Apache License v2.0\n"
00087             "// SPDX-License-Identifier: Apache-2.0\n"
00088             "// Copyright 2023 ISP RAS (http://www.ispras.ru)\n"
00089             "//\n"
00090
"//===----------------------------------------------------------------------===//\n";
00091
00092         string text = component_->getText(filename);
00093         text =utopia_license+"\n"+text;
00094         return text;
00095     }
00096 };
00097
00098 class LogoDecorator : public TextDecorator {
00099 public:
00100     LogoDecorator(TextComponent* component) : TextDecorator(component) {}
00101     string getText(const std::string& filename) override {
00102         string logo = R"";
00103
00104         string text = component_->getText(filename);
00105         text =logo+"\n"+text;
00106         return text;
00107     }
00108 };
00109
00110 int main() {
00111     TextComponent* text = new PlainText();
00112     cout << "Enter namefile:\n";
00113     string filename;
00114     cin >> filename;
00115
00116     cout << "\nCurrent text:\n" << text->getText(filename) << "\n";
00117
00118     for (int i = 0; i < 3; ++i) {
00119         std::cout << "\nEnter mode: 1 - logo, 2 - license, 3 - border\n";
00120         string tmp;
00121         cin >> tmp;
00122         if (tmp == "1") {
00123             text = new LogoDecorator(text);
00124             cout << "\n" << text->getText(filename) << "\n";
00125         }
00126         else if (tmp == "2") {
00127             text = new CompanyLicenseDecorator(text);
00128             cout << "\n" << text->getText(filename) << "\n";
00129         }
00130         else if (tmp == "3") {
00131             text = new BorderDecorator(text);
00132             cout << "\n" << text->getText(filename) << "\n";
00133         }
00134         else {
00135             cout << "Repeat input\n";
00136             --i;
00137         }
00138     }
00139     delete text;
00140     return 0;
00141 }
```

https://github.com/WhiteWLf-dev/Lab5

```
Enter namefile:
txt.txt

Current text:
#include <iostream>
int main() {
std::cout <<"Hello world\n";
}

Enter mode: 1 - logo, 2 - license, 3 - border
1
```

```
#include <iostream>
int main() {
std::cout <<"Hello world\n";
}
```

```
Enter mode: 1 - logo, 2 - license, 3 - border
2

//===-------------------------------------------------------------------------===//
//
// Part of the Utopia EDA Project, under the Apache License v2.0
// SPDX-License-Identifier: Apache-2.0
// Copyright 2023 ISP RAS (http://www.ispras.ru)
//
//===-------------------------------------------------------------------------===//
```

```
#include <iostream>
int main() {
std::cout <<"Hello world\n";
}
```

```
Enter mode: 1 - logo, 2 - license, 3 - border
3

+-------------------------------------------------------------------------------+
| //===-------------------------------------------------------------------===// |
| //                                                                           |
| // Part of the Utopia EDA Project, under the Apache License v2.0             |
| // SPDX-License-Identifier: Apache-2.0                                       |
| // Copyright 2023 ISP RAS (http://www.ispras.ru)                             |
| //                                                                           |
| //===-------------------------------------------------------------------===// |
|                                                                               |
|                                                                               |
|                                 /T /I                                         |
|                                / |/ | .-~/                                    |
|                           T\ Y  I  |/  /  _                                   |
|          /T               | \I  |  I  Y.-~/                                  |
|         I l   /I       T\ |  |  l  |  T  /                                   |
|      T\ |  \ Y l  /T   | \I  l   \ `  l Y                                    |
|   __  | \l   \l  \I l __l  l   \    `   _. |                                 |
|   \ ~-l  `\   `\  \  \\ ~\  \    `. .-~   |                                 |
|    \   ~-. "-.  `  \  ^._ ^. "-.  /  \   |                                 |
| .--~~-._  ~-  `  _  ~-_.-"-." ._ /._ ." ./                                    |
|  >--.  ~-.   ._  ~>-"    "\\   7   7   ]                                     |
| ^.___~"--._    ~-{  .-~ .  `\ Y . /    |                                     |
|  <__ ~"-.  ~       /_/   \   \I  Y   : |                                     |
|    ^-.__           ~(_/   \   >._:   | l_____                              |
|        ^--.,___.-~"  /_/   !  `-.~"--l_ /     ~"-.                          |
|               (_/ .  ~(   /'     "~"--,Y   -=b-. _)                         |
|                (_/ .  \  :           / l      c"~o \                        |
|                 \ /    `.    .     .^   \_.-~"~--.  )                       |
|                  (_/ .   `  /     /       !       )/                        |
|                   / / _.   '.   .':      /        '                         |
|                   ~(_/ .   /    _  `  .-<_                                  |
|                     /_/ . ' .-~" `.  / \  \          ,z=.                   |
|                     ~( /   '  :   | K   "-.~-._____//                      |
|                       "-,.    l   I/ \_    __{--->._(==.                    |
|                        //(     \  <    ~"~"     //                          |
|                       /' /\     \  \     ,v=.  ((                           |
|                     .^. / /\     "  }__ //===-  `                           |
|                    / / ' '  "-.,__ {---(==-                                   |
|                  .^ '       :  T  ~"   ll                                   |
|                 / .  .  . : | :!        \\                                   |
|                (_/  /   | | j-"          ~^                                   |
|                  ~-<_(_.^-~"                                                  |
|                                                                               |
| #include <iostream>                                                           |
| int main() {                                                                  |
| std::cout <<"Hello world\n";                                                  |
| }                                                                             |
+-------------------------------------------------------------------------------+
```