|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
|  |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт кибербезопасности безопасности и цифровых технологий |
| Кафедра КБ-2 «Прикладные информационные технологии» |

**Практическая работа № 4**

**Тема «Итератор, посетитель»**

по дисциплине: **«Технологии и методы программирования»**.

Подготовил студент 3 курса группы БИСО-01-21 Бондаренко Сергей Александрович

Студенческий билет : 21Б0411

Проверил преподаватель Лесько Сергей Александрович.

**Итератор (Iterator)**

Листинг программы

#include <iostream>

#include <vector>

#include <string>

// Объявление класса, представляющего итератор

class TextEditorIterator {

public:

TextEditorIterator(std::vector<std::string> text) : text\_(text), index\_(0) {}

void first() {

index\_ = 0;

}

void next() {

index\_++;

}

bool is\_done() {

return index\_ >= text\_.size();

}

std::string current\_item() {

return text\_[index\_];

}

private:

std::vector<std::string> text\_;

size\_t index\_;

};

// Класс, представляющий текстовый редактор

class TextEditor {

public:

TextEditor(std::vector<std::string> text) : text\_(text) {}

TextEditorIterator\* create\_iterator() {

return new TextEditorIterator(text\_);

}

void print\_text() {

for (std::string line : text\_) {

std::cout << line << std::endl;

}

}

private:

std::vector<std::string> text\_;

};

int main() {

std::vector<std::string> lines = { "This is a sample text",

"for demonstrating the Text Editor",

"with iterator implementation in C++" };

TextEditor text\_editor(lines);

TextEditorIterator\* iterator = text\_editor.create\_iterator();

for (iterator->first(); !iterator->is\_done(); iterator->next()) {

std::cout << iterator->current\_item() << std::endl;

}

delete iterator;

return 0;

}

Скриншоты работы программы

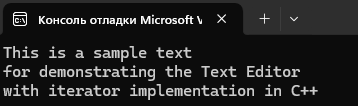


Рисунок 1 – Результат работы программы

Этот код реализует текстовый редактор с помощью класса TextEditor и создает итератор для обхода текста. При запуске программы текстовый редактор выводит каждую строку текста из вектора в консоль.

**Посетитель (Visitor)**

Листинг программы

#include <iostream>

#include <string>

#include <vector>

class TextFile;

class HighlightVisitor;

class Visitor {

public:

virtual void visitTextFile(TextFile\* file) = 0;

virtual void visitHighlightVisitor(HighlightVisitor\* visitor) = 0;

};

class Element {

public:

virtual void accept(Visitor\* visitor) = 0;

};

class TextFile : public Element {

public:

void accept(Visitor\* visitor) override {

visitor->visitTextFile(this);

}

void open() {

std::cout << "Opening text file..." << std::endl;

}

void edit() {

std::cout << "Editing text file..." << std::endl;

}

void print() {

std::cout << "Printing text file..." << std::endl;

}

};

class HighlightVisitor : public Visitor {

public:

void visitTextFile(TextFile\* file) override {

file->open();

file->edit();

file->print();

}

void visitHighlightVisitor(HighlightVisitor\* visitor) override {

std::cout << "Highlighting text file..." << std::endl;

}

};

int main() {

TextFile textFile;

HighlightVisitor highlightVisitor;

textFile.accept(&highlightVisitor);

return 0;

}

Скриншоты работы программы

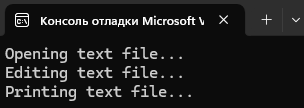


Рисунок 2 – Результат работы программы

В данном коде созданы классы TextFile, HighlightVisitor, Visitor и Element. Класс TextFile является элементом, который может принять посетителя (Visitor) для выполнения операций над ним. Класс HighlightVisitor реализует методы visitTextFile и visitHighlightVisitor для выполнения операций над текстовым файлом, такие как открытие, редактирование, печать и подсветка ключевых слов. В функции main() создается объект textFile и highlightVisitor, после чего вызывается метод accept для выполнения операции подсветки ключевых слов над текстовым файлом.

Вывод: Изучил и реализовал предложенные методы.