**21.** **Паттерны проектирования**

Задание №1

Реализовать паттерны по примерам из теоретического материала согласно вариантам. Паттерн Состояние.

Листинг программы:

Program:

using z1.Model;

Console.WriteLine(doc);

Console.WriteLine();

doc.ToPublish();

Console.WriteLine();

Console.WriteLine(doc);

Console.WriteLine();

doc.LowerState();

Console.WriteLine();

Console.WriteLine(doc);

Console.WriteLine();

doc.LowerState();

Console.WriteLine();

Console.WriteLine(doc);

Console.WriteLine();

doc.LowerState();

Console.WriteLine();

Document:

namespace z1.Model

{

internal class Document

{

private StateDocument \_state;

private string \_name;

private string \_text;

public StateDocument State { get => \_state; set => \_state = value; }

public Document(string name, string text)

{

\_state = new DraftState();

\_text = text;

}

public void ToPublish()

{

\_state.NextState(this);

}

public void LowerState()

{

\_state.BackState(this);

}

public override string ToString()

{

return $"Название: {\_name}\n" +

$"Статус: {\_state}\n" +

$"{\_text}\n";

}

}

}

StateDocument:

namespace z1.Model

{

internal abstract class StateDocument

{

public abstract void NextState(Document doc);

public abstract void BackState(Document doc);

public abstract override string ToString();

}

}

DraftState:

namespace z1.Model

{

internal class DraftState : StateDocument

{

public override void NextState(Document doc)

{

doc.State = new ModerationState();

Console.WriteLine("Документ сменил состаяние с черновика на модрерацию");

}

public override void BackState(Document doc)

{

Console.WriteLine("Документ является черновиком. Это минимальное состояние для документа");

}

public override string ToString()

{

return "Черновик";

}

}

}

ModerationState:

namespace z1.Model

{

internal class ModerationState : StateDocument

{

public override void NextState(Document doc)

{

doc.State = new PublishedState();

Console.WriteLine("Документ сменил состаяние с модрерации на публиковаеый ");

}

public override void BackState(Document doc)

{

doc.State = new DraftState();

Console.WriteLine("Документ сменил состаяние с модрерации на черновик");

}

public override string ToString()

{

return "На модерации";

}

}

}

PublishedState:

namespace z1.Model

{

internal class PublishedState : StateDocument

{

public override void NextState(Document doc)

{

Console.WriteLine("Документ является публикованным, следующего состояния нет");

}

public override void BackState(Document doc)

{

doc.State = new ModerationState();

Console.WriteLine("Документ сменил состаяние с публиковоный на модрерацию");

}

public override string ToString()

{

return "Опубликован";

}

}

}

Таблица 21.1 – Входные и выходные данные программы задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

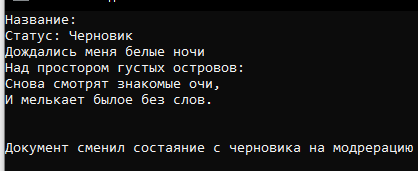


Рисунок 21.1 – Результат выполнения программы задание №1