**11** **Паттерны проектирования**

Задание 1. Реализация паттерна “Фабричный метод”

Листинг задачи:

namespace Task1

{

public class AdminFactory : UserFactory

{

public override IUser CreateUser()

{

return new AdminUser();

}

}

}namespace Task1

{

public class AdminUser : IUser

{

public string[] GetPermissions()

{

return new string[] { "чтение", "запись", "удаление", "управление пользователями" };

}

}

}namespace Task1

{

public interface IUser

{

string[] GetPermissions();

}

}namespace Task1

{

public class ModeratorFactory : UserFactory

{

public override IUser CreateUser()

{

return new ModeratorUser();

}

}

}namespace Task1

{

public class ModeratorUser : IUser

{

public string[] GetPermissions()

{

return new string[] { "чтение", "запись", "редактирование", "предупреждение пользователей" };

}

}

}namespace Task1

{

public class RegularFactory : UserFactory

{

public override IUser CreateUser()

{

return new RegularUser();

}

}

}namespace Task1

{

public class RegularUser : IUser

{

public string[] GetPermissions()

{

return new string[] { "чтение", "запись", "комментирование" };

}

}

}namespace Task1

{

public abstract class UserFactory

{

public abstract IUser CreateUser();

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Разрешения администратора, модератора, пользователя |

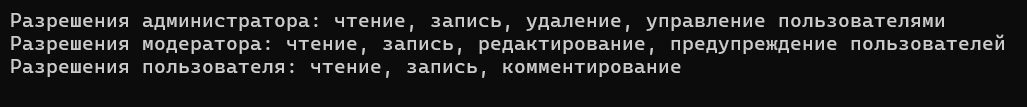
Анализ результатов:

Рисунок 1.1 – Результат работы программы