Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

**ОТЧЁТ**

К лабораторной работе №4 по дисциплине:

|  |
| --- |
| Технология программирования |
| Наследование |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: |  |  |  |  |  |
|  | шифр группы |  | подпись |  | Фамилия И.О. |
| Проверил: |  |  |  |  | З.А. Бахвалова |
|  | должность |  | подпись |  | Фамилия И.О. |

Иркутск 2021 г.

**Оглавление**

[Наследование 1](#_Toc88337709)

[Постановка задачи 3](#_Toc88337710)

[Словесное описание алгоритма 3](#_Toc88337711)

[Таблицы спецификаций 4](#_Toc88337712)

[Класс Movie 4](#_Toc88337713)

[Класс Film 4](#_Toc88337714)

[Класс Cartoon 5](#_Toc88337715)

[Класс Series 5](#_Toc88337716)

[Класс MathModel 6](#_Toc88337717)

[Класс ViewModel 7](#_Toc88337718)

[Тесты 8](#_Toc88337719)

[Код программы 8](#_Toc88337720)

[Результаты 9](#_Toc88337721)

### Постановка задачи

1. Должен быть 1 базовый класс, и 3 класса наследника. У базового класса должно быть, как минимум одно свойство передающиеся по наследству остальным, у каждого из классов наследников должно быть как минимум два уникальных свойства.
2. Реализовать эмулятор торгового автомата
3. Если вы обычно делаете желтые задачки, то в эмуляторе надо реализовать отображение состояния очереди (чтобы было видно какие объекты в каком порядке сейчас находятся в автомате)
4. Если чувствуете склонность к красным, добавьте картинки для каждого типа объектов, которые будут видны при выводе очереди

Для раздачи погоды (температура)

* Солнце (высота солнца над горизонтом, наличия свежего ветерка)
* Дождь (величина осадков, наличие радуги, наличие грозы)
* Снег (тип снега (мелкий, хлопьями и т.п.), высота сугробов)

### Словесное описание алгоритма

Метод добавления объекта в список

1. Добавление в список Кино нового объекта класса Кино
2. Добавление в список с типами кино тип текущего объекта
3. Если тип кино
   1. Если это фильм, то увеличить количество фильмов на 1
   2. Если это сериал, то увеличить количество сериалов на 1
   3. Если это мультфильм, то увеличить количество мультфильмов на 1

Метод взятия объекта

1. Если список не пуст
   1. Выбрать объект из начала списка
   2. Удалить начальный элемент из списка Кино
   3. Удалить начальный элемент из списка типов кино
   4. Переменной информация присвоить значение информации выбранного объекта
   5. Переменной тип кино присвоить значение типа кино выбранного объекта
   6. Переменной наименование кино присвоить значение наименование кино выбранного объекта
   7. Переменной изображение присвоить изображение выбранного объекта
   8. Если тип кино
      1. Если это фильм, то уменьшить количество фильмов на 1
      2. Если это сериал, то уменьшить количество сериалов на 1
      3. Если это мультфильм, то уменьшить количество мультфильмов на 1
2. Иначе
   1. Выбранный элемент присвоить значение null
   2. Переменной информация о кино присвоить значение “Автомат пуст”
   3. Переменной тип кино присвоить пустое значение
   4. Переменной наименование кино присвоить значение “Отсутствует”
   5. Переменной изображение присвоить пустой картинке

### Таблицы спецификаций

### Класс Weathers

Таблица 1 – Таблица спецификация класса Weathers

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Диапазон |
| abstract class Weathers – абстрактный родительский класс Погоды | | | |
| \_temperature | Тип кино | protected float | [-25;35] |

Таблица 2 – Таблица методов класса Weathers

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Параметры |
| GetInformation | Абстрактный метод получения информации об объекте | public abstract string | — |

### Класс Sun

Таблица 3 – Таблица спецификация класса Sun

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | | Назначение | | Тип | | Диапазон |
| class Sun : Weathers – дочерний класс Солнца | | | | | | |
| \_freshBreeze | Наличие свежего ветра | | private readonly string | | — | |
| \_heightHorizon | Высота над горизонтом | | private readonly ushor | | [0;90] | |
| \_count | Количество объектов класса | | private static ushort | | [0;30] | |

Таблица 4 – Таблица методов класса Sun

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Параметры |
| GetInformation | Метод получения информации об объекте | public override string | — |

### Класс Snow

Таблица 5 – Таблица спецификация класса Snow

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | | Назначение | | Тип | | Диапазон |
| class Snow : Weathers - дочерний класс Снега | | | | | | |
| \_type | Тип мультфильма | | private readonly string | | — | |
| \_heighSnowdrifts | Высота снежных сугробов | | private readonly ushort | | [0, 120] | |
| \_count | Количество объектов класса | | private static ushort | | [0;30] | |

Таблица 6 – Таблица методов класса Snow

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Параметры |
| GetInformation | Метод получения информации об объекте | public override string | — |

### Класс Rain

Таблица 7 – Таблица спецификация класса Rain

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | | Назначение | | Тип | | Диапазон |
| class Rain : Weathers - дочерний класс Дождя | | | | | | |
| \_episodes | Количество эпизодов | | private readonly ushort | | 0≤\_episodes ≤65535 | |
| \_seasons | Количество сезонов | | private readonly byte | | 0≤\_seasons ≤255 | |
| \_complited | Статус завершён | | private readonly string | | — | |

Таблица 8 – Таблица методов класса Rain

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Параметры |
| GetInformation | Метод получения информации об объекте | public override string | — |

### Класс MathModel

Таблица 9 – Таблица спецификация класса MathModel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | | Назначение | | Тип | | Диапазон |
| class MathModel : BindableBase – математические модель | | | | | | |
| \_namesMovie | Коллекция типов кино | | private readonly ObservableCollection<string> | | — | |
| NamesMovie | Публичная коллекция типов кино | | public ReadOnlyObservableCollection  <string> | | — | |
| \_listMovies | Список объектов класса Кино | | private readonly List<Movie> | | — | |
| \_selektedMovie | Выбранный объект | | private Movie? | | — | |
| \_filmCount | Количество фильмов | | private byte | | 0≤\_filmCount ≤255 | |
| \_seriesCount | Количество сериалов | | private byte | | 0≤\_seriesCount ≤255 | |
| \_cartoonCount | Количество мультфильмов | | private byte | | 0≤\_cartoonCount ≤255 | |
| \_typeMovie | Тип кино | | private string? | | — | |
| \_nameMovie | Наименование кино | | private string? | | — | |
| \_image | Изображение | | private string? | | — | |
| \_information | Информация | | private string? | | — | |

Таблица 10 – Таблица методов класса MathModel

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Параметры |
| PatternList | Заранее созданный список | private static List<Movie> | — |
| CreateListMovies | Случайное заполнение списка из заранее созданного | private void | — |
| AddMovies | Добавление объекта в список | private void | Movie movie |
| TakeMovie | Взятие объекта из списка | public void | — |

Таблица 11 - Таблица спецификаций методов класса MathModel

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Назначение | Тип | Диапазон |
| PatternList | | | |
| list | Заготовленный список | List<Movie> | — |
| CreateListMovies | | | |
| list | Копия заготовленного списка | List<Movie> | — |
| random | Объект класса Random | Random | — |
| lenght | Длинна будущего списка | int | 0≤ lenght ≤20 |
| AddMovies | | | |
| movie | Новый объект | Movie | — |
| TakeMovie | | | |
| — | | | |

### Класс ViewModel

Таблица 12 – Таблица спецификация класса ViewModel

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Имя | | Назначение | | Тип | | Диапазон |
| class ViewModel : BindableBase – ViewModel | | | | | | |
| \_listMovie | Экземпляр математической модели | | readonly MathModel | | — | |

### Тесты

Таблица 13 – Таблица тестов

|  |  |
| --- | --- |
| Номер теста | Назначение |
| 1 | Соответствие общего количества элементов размерам списка объектов |
| 2 | Соответствие количества объектов каждого типов количеству объектов в списке объектов |
| 3 | Соответствие списку очереди и списку объектов |
| 4 | Тестирование взятие объекта из списка |

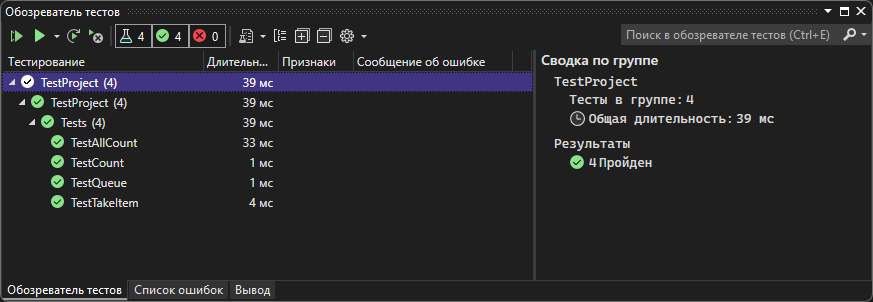


Рисунок 1 – результаты автотестов

### Код программы

Программа написанаc помощью системы для визуализации клиентских приложений WPF

Ссылка на код в GitHub: <https://github.com/imysko/Movies>

### Результаты

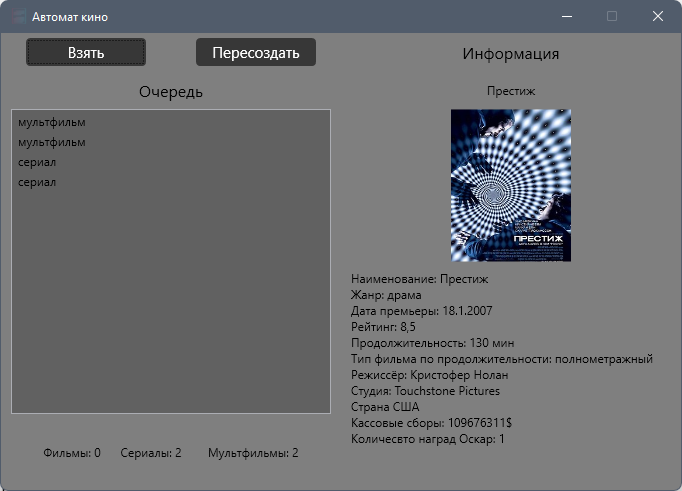


Рисунок 2 – Внешний вид программы