Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных Центр программной инженерии

ОТЧЕТ

к лабораторной работе №4 по дисциплине:

|  |  |
| --- | --- |
| «Технология программирования» | |
| Наследование | |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб-20-2 |  |  |  | Теленьков М.А. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Бахвалова З.А. |
| 16.11.21г.  **20 Баллов** | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2021 г.

Оглавление

[Лабораторное задание №4 3](#_Toc87961437)

[Общая постановка задачи 3](#_Toc87961438)

[Постановка задачи 3](#_Toc87961439)

[class Form1 – класс для работы с формой Windows Forms 3](#_Toc87961440)

[Поля класса Form1 3](#_Toc87961441)

[Методы класса Form1 3](#_Toc87961442)

[class Vehicle – класс, описывающий некий транспорт 5](#_Toc87961443)

[Поля класса Vehicle 5](#_Toc87961444)

[Методы класса Vehicle 5](#_Toc87961445)

[class Bicycle – класс–наследник Vehicle, описывающий некий велосипед 5](#_Toc87961446)

[Поля класса Bicycle 5](#_Toc87961447)

[Методы класса Bicycle 5](#_Toc87961448)

[class Car – класс–наследник Vehicle, описывающий некий автомобиль 6](#_Toc87961449)

[Поля класса Car 6](#_Toc87961450)

[Методы класса Car 7](#_Toc87961451)

[class Plane – класс–наследник Vehicle, описывающий некий самолёт 7](#_Toc87961452)

[Поля класса Plane 7](#_Toc87961453)

[Методы класса Plane 8](#_Toc87961454)

[Таблица тестов 8](#_Toc87961455)

[Код программы 9](#_Toc87961456)

[Тесты 9](#_Toc87961457)

**Выполнил:** Теленьков Максим Алексеевич, группа ИСТб-20-2

# Лабораторное задание №4

## Общая постановка задачи

Придумать иерархию классов.

* 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.
* Реализовать эмулятор торгового автомата по образу и подобию как описано тут
* 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)
* Если чувствуете склонность к красным, добавьте картинки для каждого типа объектов, которые будут видны при выводе очереди

## Постановка задачи

**Задача №4.**

Для раздачи транспортных средств (количество колес)

* Велосипеды (тип (горный, городской, детский), радиус колес)
* Автомобили (тип (автобус, грузовик, внедорожник, легковая), объем двигателя, количество дверей),
* Самолеты (тип двигателя, максимальная высота полета)

## class Form1 – класс для работы с формой Windows Forms

### Поля класса Form1

Класс Form1 не имеет полей.

### Методы класса Form1

|  |  |  |
| --- | --- | --- |
| Название метода | Назначение метода | Тип данных |
| listReCreateBtn\_Click | Создание и пересоздание элементов списка | void |
| ShowInfo | Выводит информацию о состоянии списка элементов | void |
| getBtn\_Click | Вывод информации о первом элементе списка и его последующее удаление. | void |
| infBtn\_Click | Вывод сообщения, содержащего информацию о задании. | void |

#### Словесное описание метода listReCreateBtn\_Click

1. Очистка существующего списка
2. Создание объекта класса Random
3. Цикл for до 10
   1. Заполнение i-элемента списка новым объектом
4. Обновление информации о списке

#### Таблица спецификаций метода listReCreateBtn\_Click

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| rnd | Генерация случайных чисел, для заполнения списка. | Random | [0, 2] |

#### Словесное описание метода ShowInfo

1. Создание и инициализация счётчиков для каждого типа объекта списка
2. Цикл for по всему списку
   1. Вывод информации о состоянии очереди
   2. Подсчёт количества объектов каждого типа в списке
3. Вывод информации о количестве объектов каждого типа в списке

#### Таблица спецификаций метода ShowInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| bCounter | Счётчик объектов класса Bicycle | int | [-2147483648, 2147483647] |
| cCounter | Счётчик объектов класса Car | int | [-2147483648, 2147483647] |
| pCounter | Счётчик объектов класса Plane | int | [-2147483648, 2147483647] |
| vehicle | Временное хранение данных об объекте из списка | Vehicle | Объекты класса Vehicle |

#### Словесное описание метода getBtn\_Click

1. Если список пуст
   1. Вывод сообщения о пустоте автомата
   2. Сброс поля для изображения к стандартному значению.
2. Если список не пуст
   1. Вывод всех полей первого объекта списка
   2. Удаление первого элемента из списка
3. Обновление информации о списке

#### Таблица спецификаций метода getBtn\_Click

У метода нет переменных.

#### Словесное описание метода infBtn\_Click

1. Вызов сообщения, содержащего информацию о задании

#### Таблица спецификаций метода infBtn\_Click

У метода нет переменных

## class Vehicle – класс, описывающий некий транспорт

### Поля класса Vehicle

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| nOfWheels | Количество колёс | int | [-2147483648, 2147483647] |
| rnd | Генератор случайных чисел, для случайного заполнения полей | Random | Зависит от поля. |
| img | Хранение изображения | System.Drawing.Bitmap | Изображение |

### Методы класса Vehicle

|  |  |  |
| --- | --- | --- |
| Название метода | Назначение метода | Тип данных |
| GetInfo | Вывод информации об объекте | virtual string |

#### Словесное описание метода GetInfo

1. Вывести: «Транспорт»

#### Таблица спецификаций метода GetInfo

Метод не имеет переменных

## class Bicycle – класс–наследник Vehicle, описывающий некий велосипед

### Поля класса Bicycle

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| Поля класса Vehicle | | | |
| type | Тип велосипеда | BicyleType | Горный,  Городской,  Детский, |
| wRad | Радиус колёс | int | [10, 30] |

### Методы класса Bicycle

|  |  |  |
| --- | --- | --- |
| Название метода | Назначение метода | Тип данных |
| Bicycle | Конструктор по умолчанию, заполняющий все поля случайными значениями. | Конструктор |
| GetInfo | Вывод информации об объекте | String |

#### Словесное описание конструктора Bicycle

1. Случайное определение типа
2. Определение изображения
   1. Если тип «Горный», то изображение горного велосипеда
   2. Если тип «Городской», то изображение городского велосипеда
   3. Если тип «Детский», то изображение детского велосипеда
3. Случайное определение радиуса колёс
4. Случайное определение количества колёс

#### Таблица спецификаций конструктора Bicycle

Конструктор не имеет переменных

#### Словесное описание метода GetInfo

1. Формирование строки, содержащей информацию об объекте
2. Возвращение этой строки

#### Таблица спецификаций метода GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| outMessage | Cтрока, содержащая информацию об объекте | string | Текст |

## class Car – класс–наследник Vehicle, описывающий некий автомобиль

### Поля класса Car

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| Поля класса Vehicle | | | |
| type | Тип автомобиля | CarType | Автобус,  Грузовик,  Внедорожник,  Легковая, |
| engVol | Объём двигателя | int | [1, 29] |
| dAmount | Количество дверей | int | [2, 4] |

### Методы класса Car

|  |  |  |
| --- | --- | --- |
| Название метода | Назначение метода | Тип данных |
| Car | Конструктор по умолчанию, заполняющий все поля случайными значениями. | Конструктор |
| GetInfo | Вывод информации об объекте | String |

#### Словесное описание конструктора Car

1. Случайное определение типа
2. Определение изображения
   1. Если тип «Легковая» то изображение легкового автомобиля
   2. Если тип «Грузовик» то изображение грузового автомобиля
   3. Если тип «Внедорожник» то изображение внедорожного автомобиля
   4. Если тип «Автобус» то изображение автобуса
3. Случайное определение объёма двигателя
4. Случайное определение количества дверей
5. Случайное определение количества колёс

#### Таблица спецификаций конструктора Car

Конструктор не имеет переменных

#### Словесное описание метода GetInfo

1. Формирование строки, содержащей информацию об объекте
2. Возвращение этой строки

#### Таблица спецификаций метода GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| outMessage | Cтрока, содержащая информацию об объекте | string | Текст |

## class Plane – класс–наследник Vehicle, описывающий некий самолёт

### Поля класса Plane

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| Поля класса Vehicle | | | |
| type | Тип двигателя самолёта | EngineType | Винтовой,  Реактивный, |
| maxHeight | Максимальная высота полёта | int | [1, 118000] |

### Методы класса Plane

|  |  |  |
| --- | --- | --- |
| Название метода | Назначение метода | Тип данных |
| Plane | Конструктор по умолчанию, заполняющий все поля случайными значениями. | Конструктор |
| GetInfo | Вывод информации об объекте | String |

#### Словесное описание конструктора Plane

1. Случайное определение типа
2. Определение изображения
   1. Если тип «Винтовой» то изображение винтового самолёта
   2. Если тип «Реактивный» то изображение реактивного самолёта
3. Случайное определение максимальной высоты полёта
4. Случайное определение количества колёс

#### Таблица спецификаций конструктора Plane

Конструктор не имеет переменных

#### Словесное описание метода GetInfo

1. Формирование строки, содержащей информацию об объекте
2. Возвращение этой строки

#### Таблица спецификаций метода GetInfo

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | Назначение | Тип данных | Диапазон допустимых значений |
| outMessage | Cтрока, содержащая информацию об объекте | string | Текст |

## Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Создание списка | Первое нажатие на кнопку «Создать список» | Создание списка из десяти случайных элементов |
| 2 | Пересоздание списка | Повторное нажатие на кнопку «Создать список» | Пересоздание списка на новый список из десяти новых случайных элементов |
| 3 | «Вытаскивание» объекта | Нажатие на кнопку «Взять талон», пока в списке есть элементы | Отображение информации о первом элементе списка, удаление этого элемента из списка. |
| 4 | «Вытаскивание» объекта из пустого списка | Нажатие на кнопку «Взять талон», когда в списке нет элементов | Отображение надписи «Автомат пуст!» |

## Код программы

Репозиторий проекта на GitHub: <https://github.com/ValletOr/Lab4-Task4.git>

## Тесты

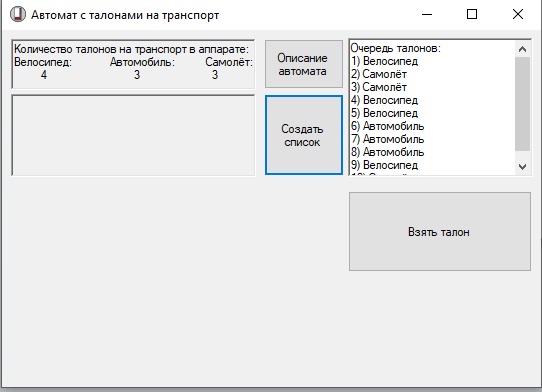


Рисунок 1 – Тест 1

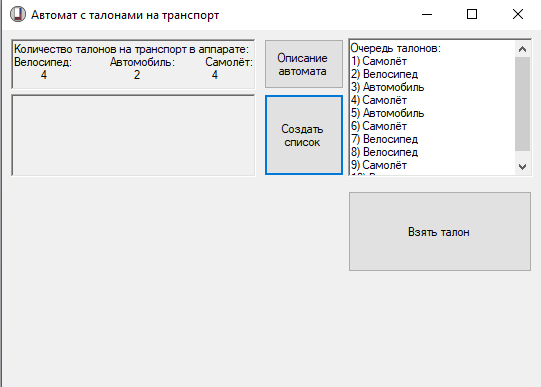


Рисунок 2 – Тест 2

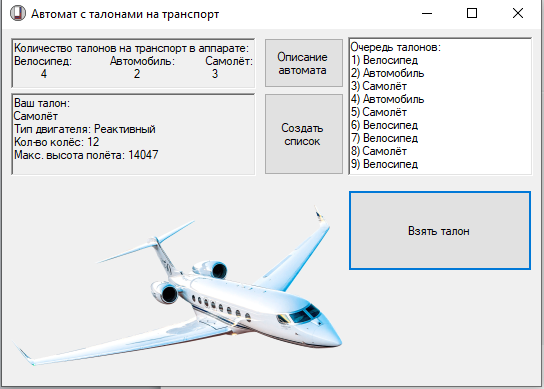


Рисунок 3 – Тест 3

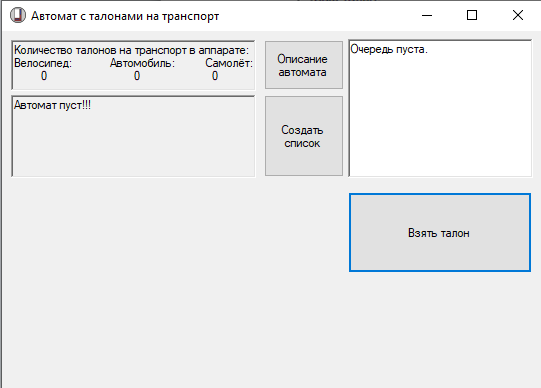


Рисунок 4 – Тест 4