|  |
| --- |
| Министерство науки и высшего образования Российской Федерации |
| федеральное государственное бюджетное образовательное учреждение высшего образования |
| "Российский экономический университет имени Г.В. Плеханова" |
| **МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ** |

МДК 01.04 Системное программирование

ПМ 01 Разработка программного обеспечения для компьютерных систем

**Специальность: 09.02.07 «Информационные системы и программирование»**

Квалификация: программист

|  |  |
| --- | --- |
| Выполнил студент: | Проверил преподаватель: |
| Группы П50-2-21 | Пахомов Д. А. |
| Дубовик Д.А. | «\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2022 года |
| « » \_\_\_\_\_\_ 2023 год | Оценка: \_\_\_\_\_\_\_\_\_ |
|  |  |

**СОДЕРЖАНИЕ**

[**ПРАКТИЧЕСКАЯ РАБОТА №1** 3](#_Toc130033785)

[Калькулятор 3](#_Toc130033786)

[Создание функции ввода 3](#_Toc130033787)

[Основные математические операции 4](#_Toc130033788)

[Числа Фиббоначи 8](#_Toc130033789)

[**ПРАКТИЧЕСКАЯ РАБОТА №2** 10](#_Toc130033790)

[«Указатели, Функции, Массивы» 10](#_Toc130033791)

[**ПРАКТИЧЕСКАЯ РАБОТА №3** 13](#_Toc130033792)

[Структуры, Классы 13](#_Toc130033793)

# **ПРАКТИЧЕСКАЯ РАБОТА №1**

## Калькулятор

Цель работы: разработать калькулятор, поддерживающий следующие операции: сложение, вычитание, умножение, деление, остаток от деления, а так же находить числа Фиббоначи. Для последних необходима возможность задать номер находимого числа и вывести каждое число последовательности, а также их сумму. Необходимо обеспечить «защиту от дураков».

Ход работы:

### Создание функции ввода

Так как по условию поставленной задачи необходимо не допустить ситуации, когда пользователь вводит не числовое значение и программа не сообщает ему об ошибке, то напишем функцию, которая при ошибке приведения типов данных будет уведомлять пользователя о ошибке и очищать буфер ввода.

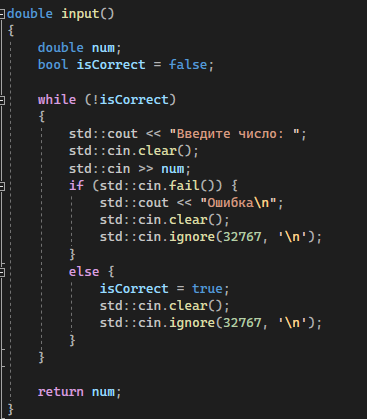


Рисунок – код функции ввода чисел input()

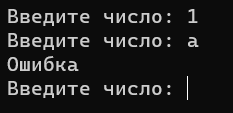


Рисунок – результат работы функции input()

### Основные математические операции

Далее была написана основная часть программы, которая отвечает за вывод меню, выбор операции, сами операции. Непосредственно основная часть программы заключена в цикл с пост условием do…while(). В самом начале объявляются 3 переменные: 2 числовые с типом double и 1 логическая, отвечающая за продолжение выполнения программы и по умолчанию имеющая значение true.

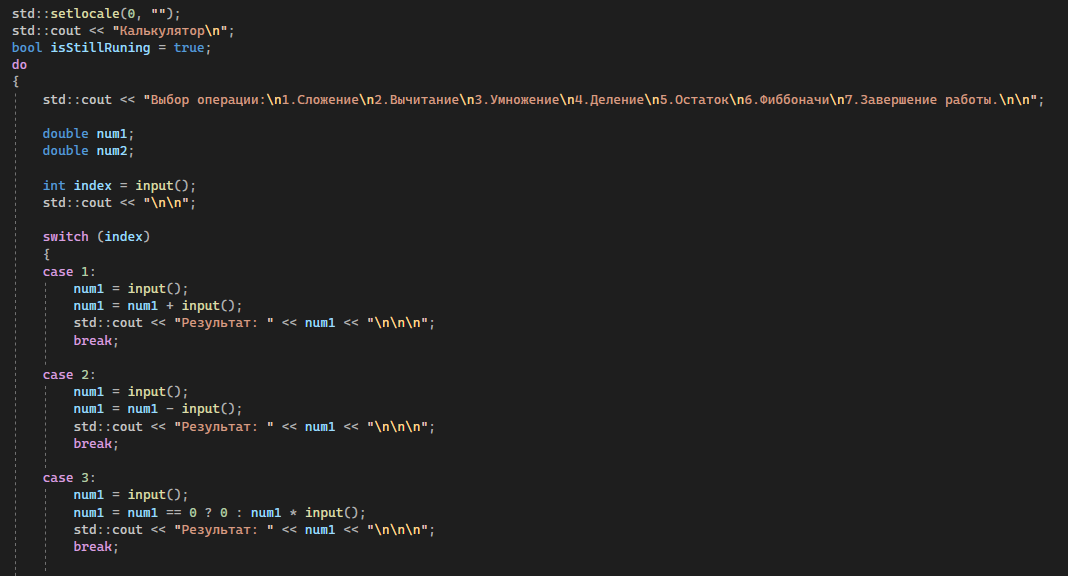


Рисунок – объявление переменных, операции сложения, вычитания и умножения в main

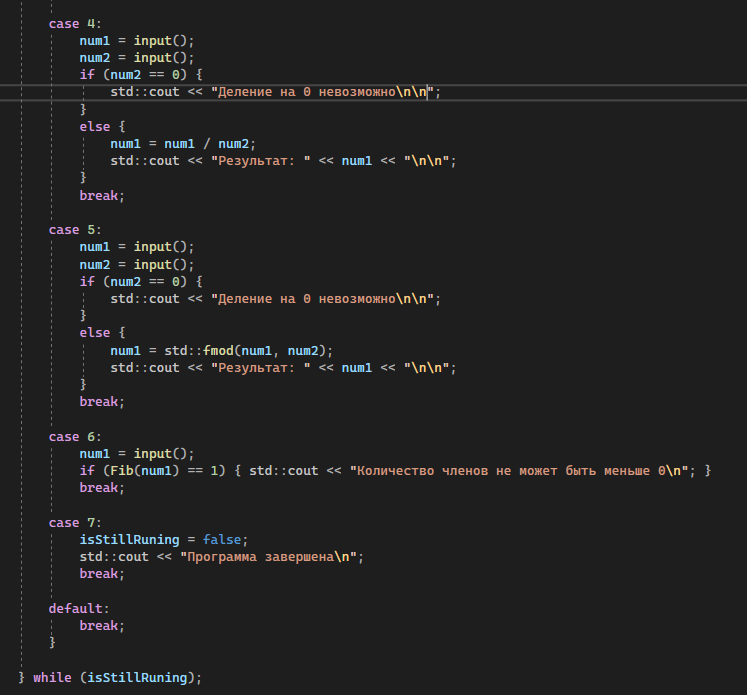


Рисунок – деление, нахождение остатка, числа Фиббоначи и выход из программы в main

Для деления и умножения реализованы проверки на нулевые значения. Для чисел Фиббоначи была создана функция заглушка.

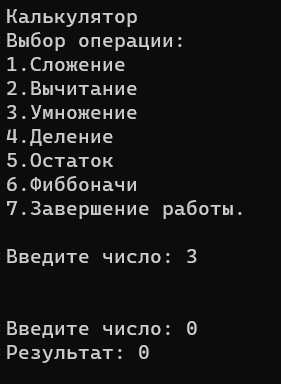


Рисунок – результат выполнения умножения на 0

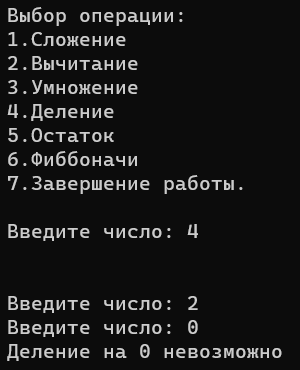


Рисунок – результат попытки делить на 0

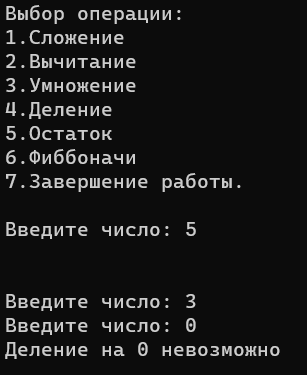


Рисунок – результат попытки найти остаток от деления на 0

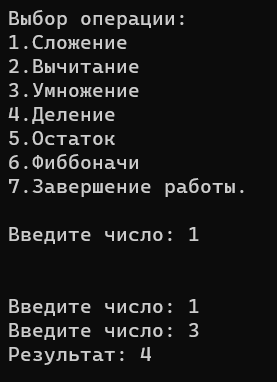


Рисунок – результат выполнения сложения

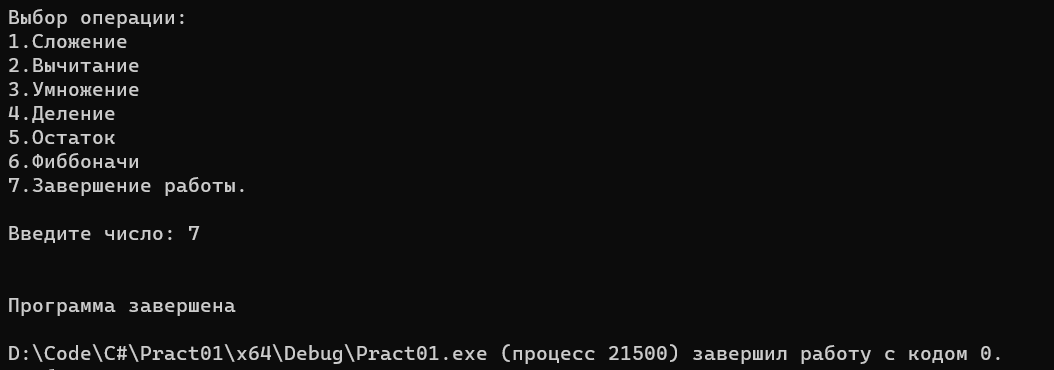


Рисунок – завершение работы программы

### Числа Фиббоначи

Для подсчета чисел Фиббоначи было решено использовать цикл с счетчиком. В случае если номер числа менее 1, то пользователю сообщает об ошибке, а если он равен единице, то вывит заранее сформированное сообщение. Для работы с особо большими числами был использован тип данных long без знака.

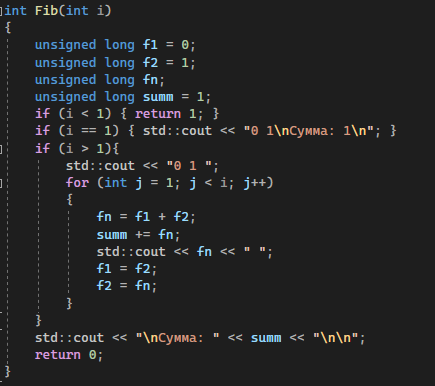


Рисунок – функция вычисления чисел Фиббоначи

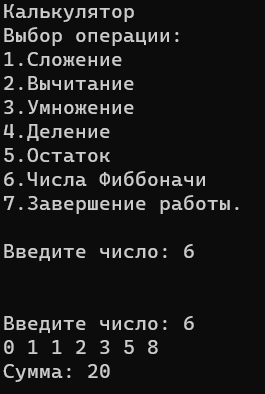


Рисунок – результат вычисления чисел Фиббоначи

Вывод: разработал калькулятор, поддерживающий следующие операции: сложение, вычитание, умножение, деление, остаток от деления, а так же находить числа Фиббоначи. Для последних была реализована возможность задать номер находимого числа и вывести каждое число последовательности, а также их сумму. Была обеспечена «защита от дураков».

# **ПРАКТИЧЕСКАЯ РАБОТА №2**

## «Указатели, Функции, Массивы»

Цель работы: научиться работать с указателями, функциями, а также массивами на языке программирования C++.

Ход работы:

Для начала напишем функцию для вычисления факториала при помощи рекурсии.

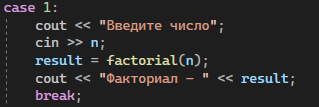


Рисунок – функция вычисления факториала №1

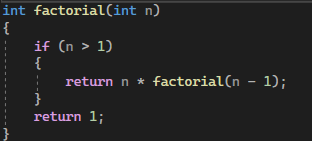


Рисунок – функция вычисления факториала №2

Далее реализуем при помощи указателей перестановку переменных.

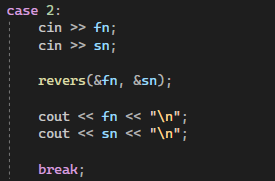


Рисунок – функция замены переменных №1

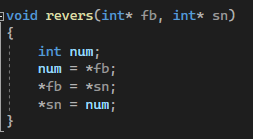


Рисунок – функция замены переменных №2

После этого можно начать работу с массивами. Для начала сделаем создание массива с заявленной пользователем длинной и элементами. Также добавим подсчёт суммы значений массива.

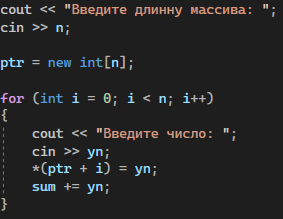


Рисунок - создание массива

После этого можно уже производить операции над массивом. Сделаем поиск среднего значения всех элементов массива.

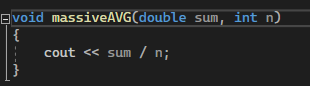


Рисунок - поиск среднего значения массива

После можно сделать вывод всех значений массива.

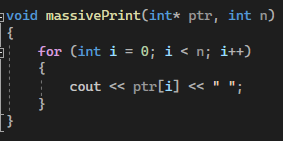


Рисунок - вывод всех значений

Итоговый вывод в массиве будет выглядеть так:

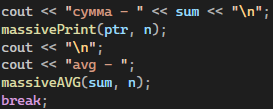


Рисунок - код вывода

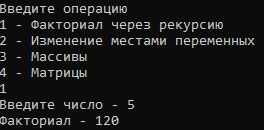


Рисунок - вывод в консоль факториала

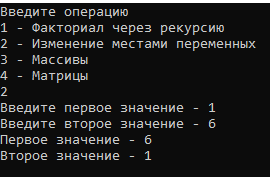


Рисунок - вывод в консоль изменения значений

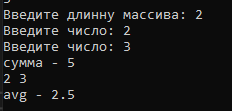


Рисунок - общий вывод массива

Вывод: научились работать с указателями, функциями и массивами на языке программирования C++.

# **ПРАКТИЧЕСКАЯ РАБОТА №3**

## Структуры, Классы

Цель работы: научиться работать с классами, создавать из экземпляры, а также обращаться ких полям и методам.

Ход работы:

Создадим класс "Автомобиль" (Car), который имеет следующие свойства и методы:

Свойства:

Марка автомобиля (brand), Модель автомобиля (model), Год выпуска автомобиля (year), Скорость автомобиля (speed).

Методы:

Конструктор класса (constructor), который принимает марку, модель, год выпуска и скорость автомобиля. Метод "увеличить скорость" (increaseSpeed), который принимает значение скорости в км/ч и увеличивает скорость автомобиля на это значение. Метод "уменьшить скорость" (decreaseSpeed), который принимает значение скорости в км/ч и уменьшает скорость автомобиля на это значение. Метод "получить марку автомобиля" (getBrand), который возвращает марку автомобиля. Метод "получить модель автомобиля" (getModel), который возвращает модель автомобиля. Метод "получить год выпуска автомобиля" (getYear), который возвращает год выпуска автомобиля. Метод "получить скорость автомобиля" (getSpeed), который возвращает скорость автомобиля.

Создадим класс "Грузовик" (Truck), который наследует свойства и методы класса "Автомобиль", и имеет дополнительные свойства и методы: Дополнительные свойства:

Грузоподъемность (payloadCapacity) Дополнительные методы:

Конструктор класса (constructor), который принимает марку, модель, год выпуска, скорость и грузоподъемность грузовика. Метод "получить грузоподъемность" (getPayloadCapacity), который возвращает грузоподъемность грузовика. Создать список по классу Автомобиль и Грузовик с 10 данными и вывести его.

Ход работы:

Создадим поля класса автомобиль и его функции:

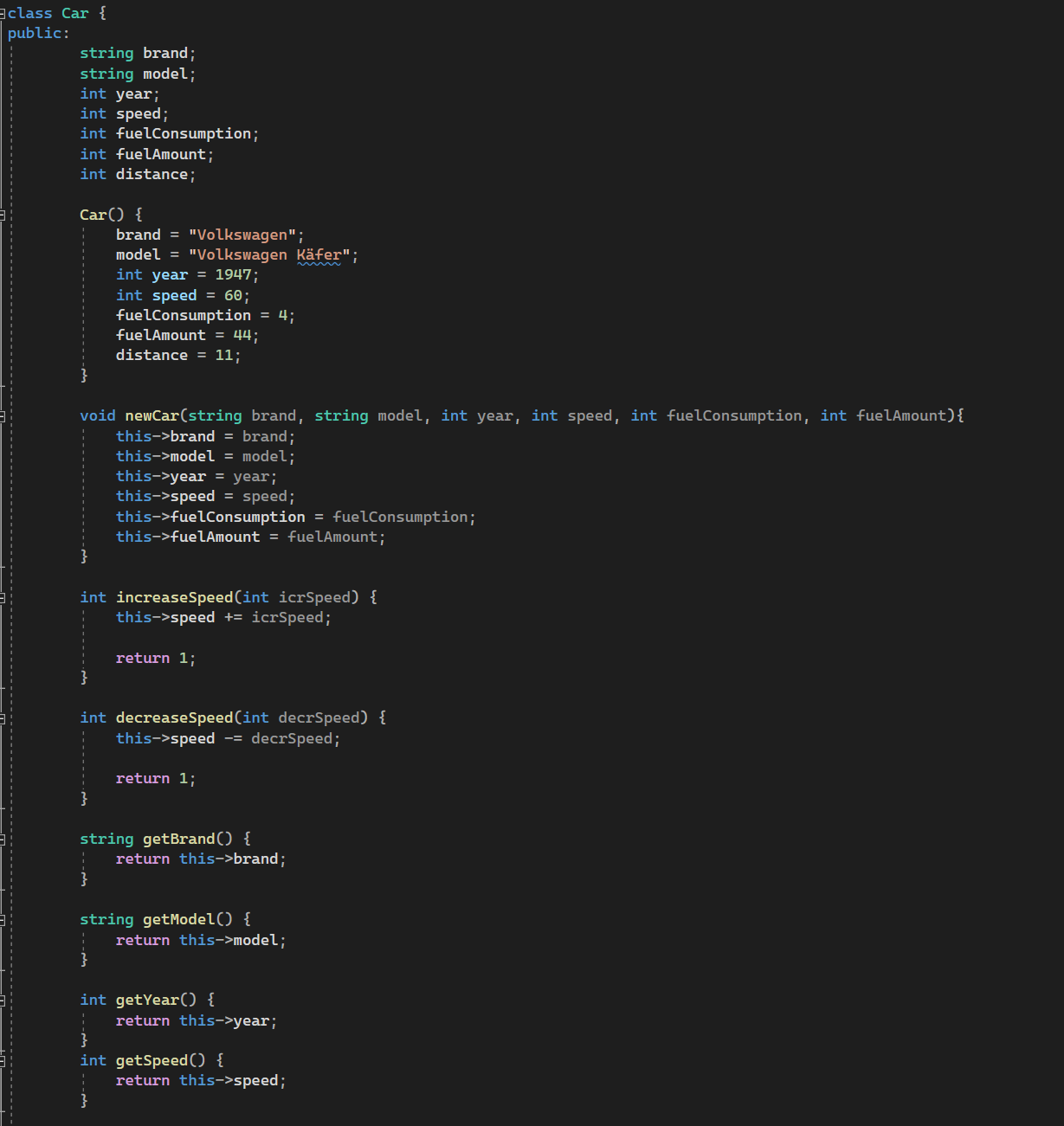


Рисунок – код класса car

Так же создадим и класс Truck:

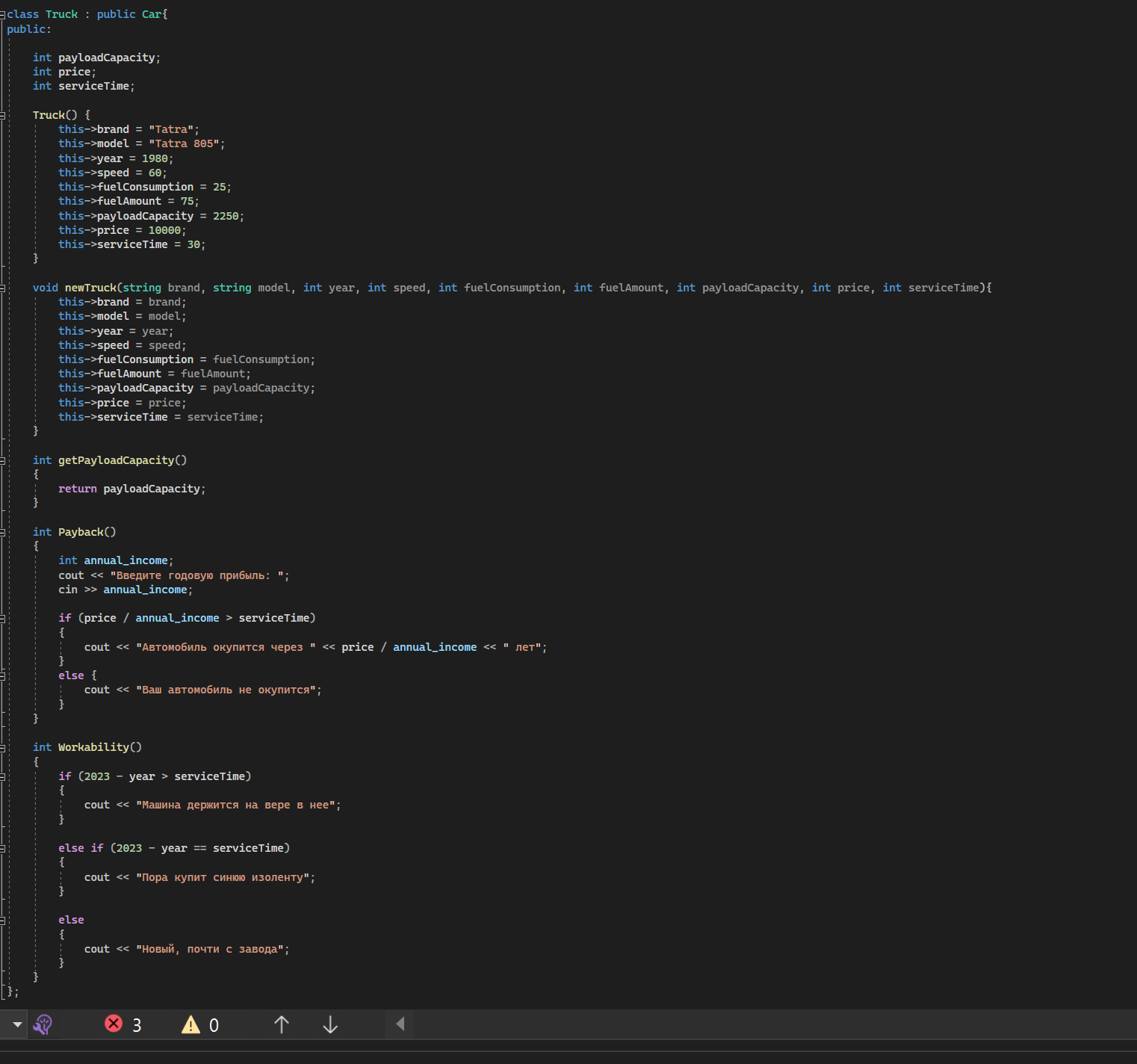


Рисунок – класс Truck

Теперь создадим 2 связных списка на 10 машин и 10 грузовиков соответственно, а затем выведем все их содержимое.

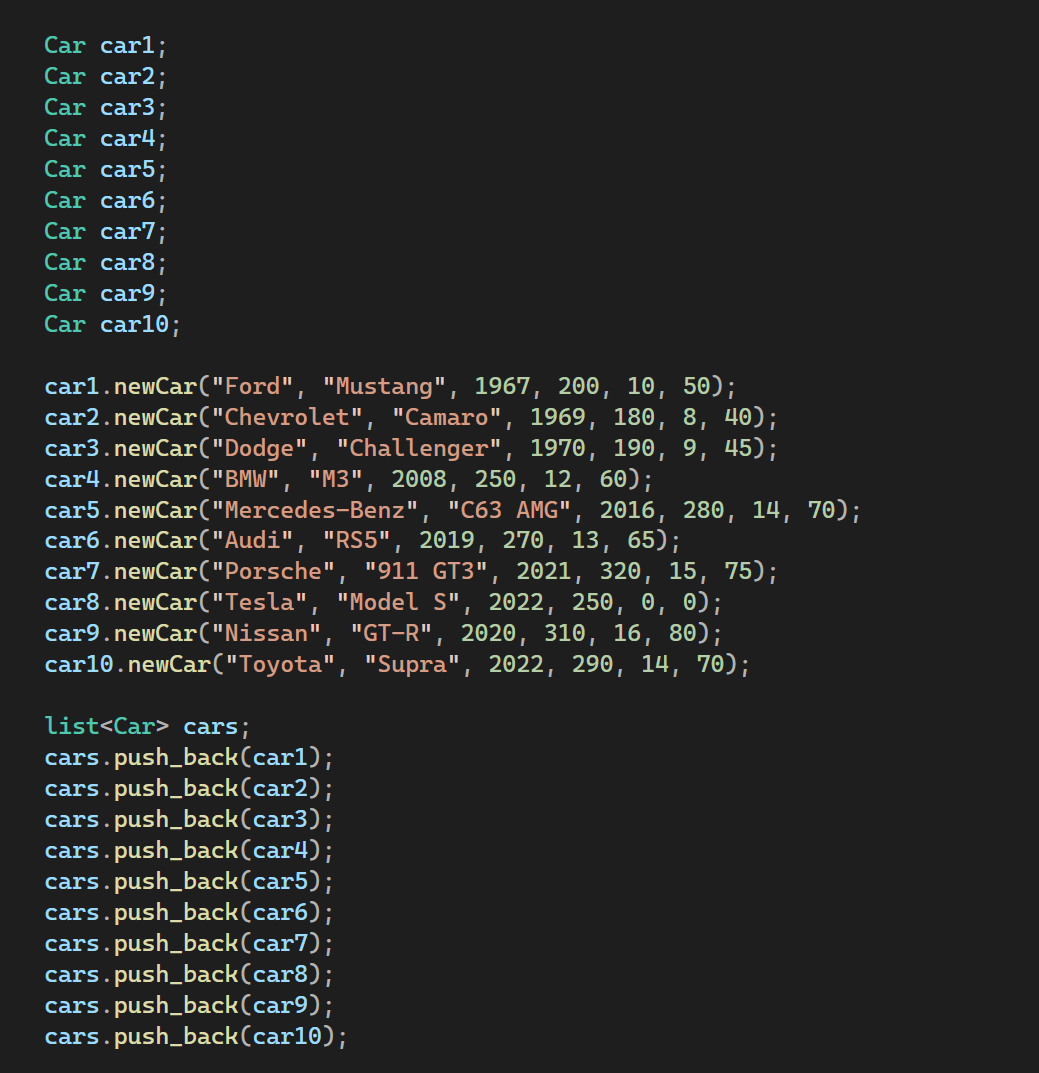


Рисунок – машины

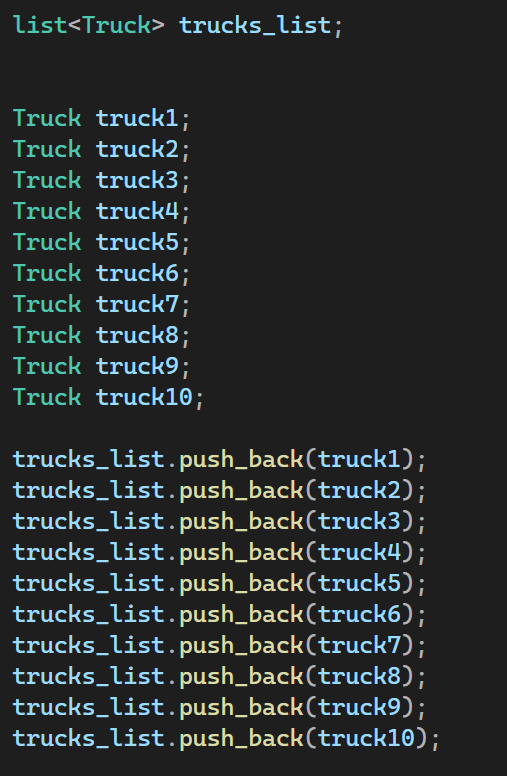


Рисунок – грузовики

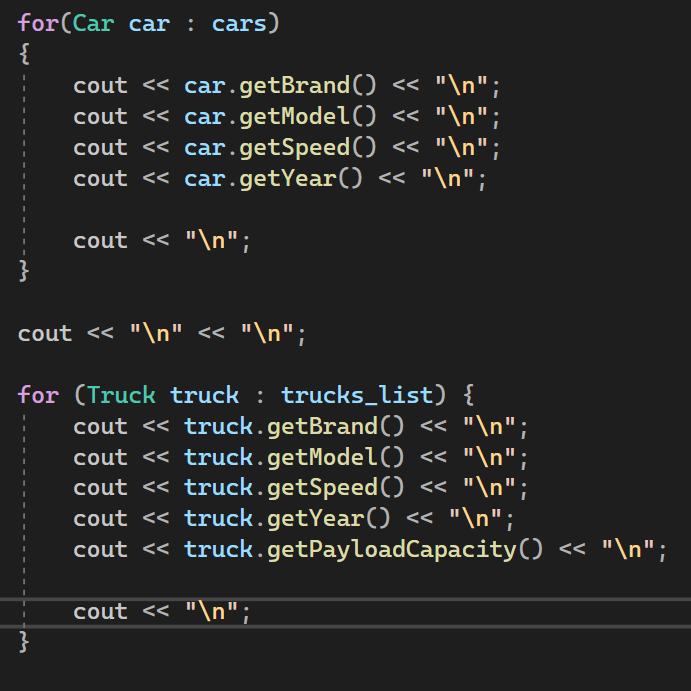


Рисунок – вывод элементов

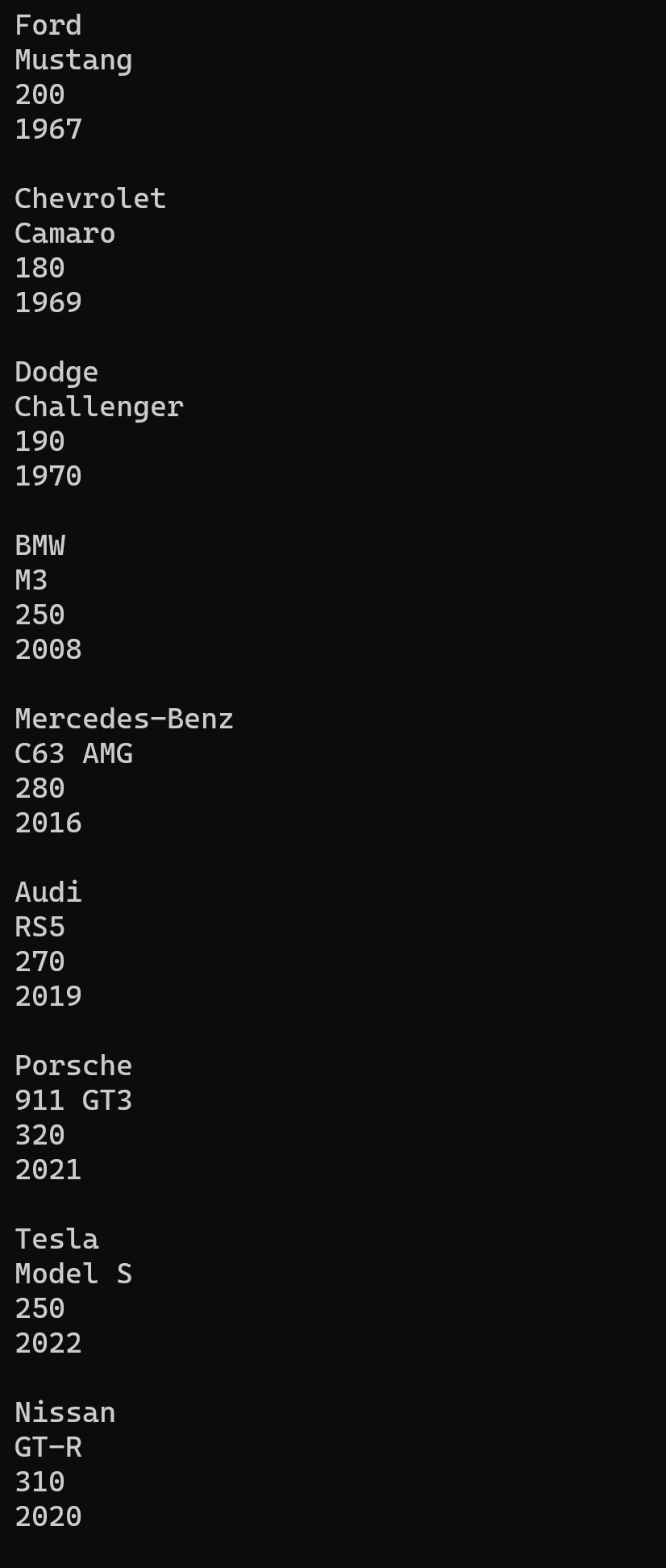


Рисунок – результат работы программы

Свойства: научились работать с классами, создавать из экземпляры, а также обращаться ких полям и методам.

[Рисунок 1 – код функции ввода чисел input() 3](#_Toc130033747)

[Рисунок 2 – результат работы функции input() 3](#_Toc130033748)

[Рисунок 3 – объявление переменных, операции сложения, вычитания и умножения в main 4](#_Toc130033749)

[Рисунок 4 – деление, нахождение остатка, числа Фиббоначи и выход из программы в main 5](#_Toc130033750)

[Рисунок 5 – результат выполнения умножения на 0 6](#_Toc130033751)

[Рисунок 6 – результат попытки делить на 0 6](#_Toc130033752)

[Рисунок 7 – результат попытки найти остаток от деления на 0 7](#_Toc130033753)

[Рисунок 8 – результат выполнения сложения 7](#_Toc130033754)

[Рисунок 9 – завершение работы программы 7](#_Toc130033755)

[Рисунок 10 – функция вычисления чисел Фиббоначи 8](#_Toc130033756)

[Рисунок 11 – результат вычисления чисел Фиббоначи 8](#_Toc130033757)

[Рисунок 12 – функция вычисления факториала №1 10](#_Toc130033758)

[Рисунок 13 – функция вычисления факториала №2 10](#_Toc130033759)

[Рисунок 14 – функция замены переменных №1 10](#_Toc130033760)

[Рисунок 15 – функция замены переменных №2 10](#_Toc130033761)

[Рисунок 16 - создание массива 11](#_Toc130033762)

[Рисунок 17 - поиск среднего значения массива 11](#_Toc130033763)

[Рисунок 18 - вывод всех значений 11](#_Toc130033764)

[Рисунок 19 - код вывода 11](#_Toc130033765)

[Рисунок 20 - вывод в консоль факториала 12](#_Toc130033766)

[Рисунок 21 - вывод в консоль изменения значений 12](#_Toc130033767)

[Рисунок 22 - общий вывод массива 12](#_Toc130033768)

[Рисунок 23 – код класса car 14](#_Toc130033769)

[Рисунок 24 – класс Truck 15](#_Toc130033770)

[Рисунок 25 – машины 16](#_Toc130033771)

[Рисунок 26 – грузовики 16](#_Toc130033772)

[Рисунок 27 – вывод элементов 17](#_Toc130033773)