15. Реализуйте класс TaxiCompany (Таксопарк), который представляет собой систему управления такси. Класс должен иметь методы для добавления нового такси, удаления такси, поиска такси по номеру и расчета общей доходности всего таксопарка. В системе должны быть предусмотрены различные типы такси (например: легковые автомобили, минивэны, электромобили) с разными характеристиками и тарифами. Класс TaxiCompany должен также предоставлять методы для вывода текущего состояния таксопарка и информации о доходах в терминал. Предусмотрите случаи, когда может генерироваться исключение, например, при попытке добавить такси с некорректными характеристиками или при поиске такси, которого нет в таксопарке.

Таблица 4.2

Варианты работ

№ варианта	Номера заданий к варианту
1	1, 8
2	2, 5
3	3, 7
4	4, 10
5	5, 11
6	6, 12
7	7, 14
8	8, 13
9	9, 15
10	3, 11
11	4, 7
12	1, 12
13	2, 9
14	3, 11
15	4, 13
16	5, 8
17	6, 15
18	7, 11
19	8, 15
20	10, 13

Лабораторная работа № 8. Перегрузка операторов

Цель работы: познакомиться с основными способами перегрузки операторов в Dart.

Требования к формату защиты лабораторной работы:

- **Отчет** (титульный лист, текст задания с кодом по его выполнению);
- **Готовность внести исправления**, в присутствии преподавателя, в код любого из выполненных заданий лабораторной работы и **ответить на вопросы**;
- **Каждое задание должно сопровождаться минимум пятью тестами и содержать хотя бы одно исключение.**

Выберете вариант, соответствующий вашему порядковому номеру в журнале группы. В том случае, если ваш порядковый номер больше последнего номера варианта, используйте следующую формулу: N = n % f + 1, где n – ваш порядковый номер, f – номер последнего варианта, N – вариант для выполнения.

Задания:

1. Создайте класс Matrix, который будет представлять двумерную матрицу и перегружать операции умножения и сложения для работы с матрицами любых размеров (например, вы можете написать методы, которые будут проверять размеры матриц перед выполнением операций). Также добавите метод для транспонирования матрицы.

2. Создайте класс Date, который будет представлять дату (год, месяц и день) и перегружать операторы для сравнения дат.

3. Реализуйте класс Time, представляющий время в формате часы:минуты:секунды. Перегрузите операторы + и - для выполнения арифметических операций со временем. Например, оператор + должен позволять складывать два времени и получать новое время как результат.

4. Создайте класс Student с атрибутами name и grade. Перегрузите операции сравнения, для их использования при сортировке списка студентов по их оценкам, а если у двух студентов оценки совпадают, то сортировка должна производиться по алфавиту (по имени студента).

5. Реализуйте класс Color, представляющий цвет в формате RGB. Перегрузите операторы [] для доступа к компонентам цвета (красный, зеленый, синий) по индексу. Также перегрузите операторы &, | и ^ для выполнения побитовых операций над цветами.

6. Создайте класс Fraction, который имеет два атрибута: числитель и знаменатель. Для класса необходимо перегрузить операции сложения, вычитания, умножения и деления, чтобы можно было производить арифметические операции с дробями. При этом методы сложения и вычитания должны возвращать новый объект класса Fraction, а методы умножения и деления - результат в виде числителя и знаменателя.

7. Создайте класс Color, имеющий три атрибута: red, green и blue, каждый из которых должен быть целым числом в диапазоне от 0 до 255. Перегрузите у класса методы сложения и умножения. Результатом сложения должен

быть новый объект класса Color, у которого каждый из атрибутов будет равен сумме соответствующих атрибутов исходных цветов (если сумма больше 255, то атрибут равен 255). Аргументом метода умножения является число в диапазоне от 0 до 1, а его результатом должен быть новый объект класса Color, у которого каждый из атрибутов будет умножен на данный аргумент и округлен до целого числа.

8. Реализуйте класс Vector, представляющий вектор в трехмерном пространстве. Перегрузите операторы +, -, * для выполнения арифметических операций с векторами. Также перегрузите оператор == для сравнения векторов. Реализуйте доступ к элементам вектора по индексу.

9. Реализуйте класс MyString, представляющий строку, но хранящий ее элементы в списке. Перегрузите операторы + для конкатенации строк. Также перегрузите операторы ==, >, < для лексикографического сравнения строк. Реализуйте доступ к символам строки по индексу.

10. Реализуйте класс BitArray, представляющий массив битов. Перегрузите операторы [] для доступа к элементам массива по индексу. Также перегрузите операторы &, |, ^, >> и << для выполнения побитовых операций над битами массива.

11. Реализуйте класс Rectangle, представляющий прямоугольник. Перегрузите оператор * для выполнения операции масштабирования прямоугольника на коэффициент. Например, умножение прямоугольника на 2 должно увеличивать его размеры вдвое. Также перегрузите операторы сравнения, где экземпляры класса будут сравниваться по их площади.

12. Реализуйте класс Point, представляющий точку в двумерном пространстве. Перегрузите операторы + и - для выполнения арифметических операций с точками. Например, оператор + должен позволять складывать две точки и получать новую точку как результат. Предусмотрите метод, возвращающий расстояние между точками.

13. Реализуйте класс Employee, представляющий сотрудника компании. Перегрузите операторы + и - для выполнения операций увеличения и уменьшения зарплаты сотрудника соответственно. Также перегрузите операторы равенства и неравенства для сравнения сотрудников по зарплате.

14. Реализуйте класс IPAddress, представляющий IP-адрес. Перегрузите операторы [] для доступа к октетам IP-адреса по индексу. Также перегрузите операторы &, | и ^ для выполнения побитовых операций над IP-адресами.

15. Реализуйте класс TemperatureSensor, представляющий датчик температуры. Перегрузите операторы [] для доступа к значениям температуры, полученной от датчика (класс Sensor), по индексу (например, 0 для значения в Цельсиях, 1 для значения в Фаренгейтах и т.д.). Также

перегрузите операторы &, | и ^ для выполнения побитовых операций над значениями датчика температуры.

Таблица 4.3

Варианты работ

№ варианта	Номера заданий к варианту
1.	2, 9
2.	3, 11
3.	4, 13
4.	5, 8
5.	6, 15
6.	1, 12
7.	3, 11
8.	4, 7
9.	7, 14
10.	8, 13
11.	9, 15
12.	4, 10
13.	5, 11
14.	6, 12
15.	7, 11
16.	8, 15
17.	10, 13
18.	1, 8
19.	2, 5
20.	3, 7