

18. Какие ограничения на класс накладывает модификатор `base`? Приведите примеры.
19. Какие ограничения на класс накладывает модификатор `interface`? Приведите примеры.
20. Какие ограничения на класс накладывает модификатор `final`? Приведите примеры.
21. Какие ограничения на класс накладывает модификатор `sealed`? Приведите примеры.
22. Какие ограничения на класс накладывает модификатор `mixin`? Приведите примеры.
23. Что такое `Private field promotion` и когда он не работает?
24. Что такое `mixins` (примеси)? Для чего они используются?
25. Как в Dart реализованы `generics` (Обобщения)? Для чего они нужны?
26. Перечислите ограничения, которые имеются у обычного и расширенного `enum`.
27. Что такое исключение? Для чего они используются?
28. Какие два базовых класса для работы с исключениями существуют?
29. В чем смысл приведения к базовому классу или интерфейсу? Приведите примеры.
30. Для каких целей и как используется каждая конструкция `try...catch...finally`?
31. Какое ключевое слово следует использовать для того, чтобы сгенерировать исключение? Посредством какого ключевого слова можно распространить перехваченное исключение дальше?
32. Что такое трассировка стека? Как использовать данный механизм?
33. Для чего используются утверждения? Всегда ли выполняются утверждения в коде?

Лабораторная работа № 7. Объектно-ориентированное программирование

Цель работы: познакомиться с основными способами написания объектно-ориентированного кода 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. Реализуйте класс «Комната общежития». В нем должна храниться следующая информация: количество проживающих студентов, максимальный размер комнаты (количество кроватей) и текущий размер комнаты. При добавлении экземпляра класса «Студент», проверяется, если количество проживающих студентов превышает максимальный размер комнаты, должна выводиться ошибка. На коменданта, даже в случае переполнения комнаты, такие ограничения не должны действовать. Также реализуйте возможность вывода текущих состояний объектов в терминал.

2. Реализуйте класс «Студент», который должен содержать ФИО студента, дату поступления и балл GPA. Балл GPA должен быть в диапазоне от 2.0 до 4.0. Если балл находится вне этого диапазона, то студенту следует отказать в добавлении в группу (класс «Группа студентов»). Группа должна предоставлять метод поиска студента по его ФИО и выводить список студентов ориентируясь на заданный диапазон баллов. Также реализуйте возможность вывода текущих состояний объектов в терминал.

3. Реализуйте класс «Книжный шкаф», который содержит информацию о максимальном весе хранимых книг и их количестве. Класс «Книга» должен содержать поля: название, автор, год издания, вес и стоимость книги. При добавлении книги в шкаф осуществляется проверка по весу и количеству книг, которые может вместить шкаф в данный момент. Если один из этих параметров больше разрешенного, то книга в шкаф не добавляется. Дополнительно для книжного шкафа реализуйте методы поиска книги по автору, расчет полной стоимости и веса хранимых книг, получения списка книг, чья стоимость \geq указанной. Также реализуйте возможность вывода текущих состояний объектов в терминал.

4. Реализуйте класс «Кошелек» куда могут добавляться экземпляры классов денежных валют (рубль и юань) различного номинала. Кошелек должен иметь возможность возвращать полную сумму денег всех валют в задаваемой валюте, количество денег конкретной валюты и уметь конвертировать с задаваемым коэффициентом деньги из одной валюты в другую. Требуется реализовать возможность вывода текущих состояний объектов в терминал.

5. Создайте класс Parking, который может хранить объекты класса Car. В конструктор Parking следует передать максимальное количество автомобилей, которые могут быть припаркованы. У класса Car должны быть

следующие поля: производитель, модель и гос. номер автомобиля. Класс `Parking` имеет следующие методы: добавление автомобиля, удаление автомобиля, поиск автомобиля по гос. номеру и подсчет количества автомобилей на парковке. Если парковка заполнена, то автомобиль не может быть добавлен, о чем предупредите пользователя. Также обеспечьте вывод текущего состояния объектов в терминал.

6. Реализуйте класс `Barbell` (Гриф), способный удерживать задаваемый предельный вес и класс `Plate` (Блин) (10, 15, 20 кг). Экземпляры класса `Plate` должны навешиваться на `Barbell` как с левой, так и с правой стороны. Класс `Barbell` должен предоставлять методы для получения общего веса блинов и разницы между левой и правой стороной, которая не должна превышать 15 кг. Если разница превышает этот вес, то добавление блина на штангу запрещено и об этом должно быть выведено сообщение. Также обеспечьте вывод текущего состояния объектов в терминал.

7. Реализуйте класс `Product` и `Order`. `Order` может состоять из любого количества `Products`, которые составляют его итоговую стоимость. Предусмотрите возможность предоставления скидок на определенные товары в заказе и на общую стоимость заказа. Обеспечьте возможность получения информации о стоимости товаров определенного типа в заказе и вывода текущего состояния объектов на терминал.

8. Реализуйте класс `Scale` (весы) и `Product`. В конструктор класса `Scale` передайте максимальный вес, который они могут выдержать. Также данный класс должен предоставлять пользователю следующие методы: добавление и удаление товаров, сортировку товаров по стоимости и весу, нахождение товара с минимальной стоимостью, товара с максимальным весом, определение текущего веса всех товаров на весах и общего веса товаров определенного типа. Если общий вес товаров равен или превышает максимальный, новые товары не должны добавляться на весы (вывод оповещения). Также обеспечьте вывод текущего состояния объектов в терминал.

9. Создайте класс `DecimalCounter`. В конструкторе этого класса укажите диапазон счета (например, от 3 до 87). Класс должен увеличивать или уменьшать свое значение на 1 в указанном диапазоне. Вы можете инициализировать этот класс значениями по умолчанию или произвольными значениями. `DecimalCounter` должен иметь следующие методы: увеличить значение, уменьшить значение и получить текущее значение. Если счетчик достигает минимального значения, и вы вызываете метод уменьшения, то значение счетчика становится равным максимальному значению и наоборот. Каждый такой переход должен сопровождаться оповещением пользователя. Также обеспечьте вывод текущего состояния объектов в терминал.

10. Реализуйте класс «Акции», который будет содержать информацию о компании, эмитенте акций, количестве акций в портфеле, текущей котировке и курсе заданной валюты, а также класс «Инвестиционный портфель». Портфель инвестора должен содержать методы для получения полной суммарной стоимости всех акций в портфеле; по определенной отрасли; возвращающий список акций компании заданной отрасли. Также обеспечьте вывод текущего состояния объектов в терминал.

11. Реализуйте класс Library (Библиотека), который содержит информацию о книгах (Book). Класс Library должен иметь методы для добавления новой книги, удаления книги, поиска книги по названию и получения общей стоимости всех книг в библиотеке. Также предусмотрите возможность предоставления скидки на определенные книги, на общую стоимость всех книг в библиотеке и обеспечьте вывод текущего состояния объектов в терминал.

12. Реализуйте класс Delivery (Доставка), который представляет собой систему доставки товаров. Класс должен иметь методы для добавления нового заказа (класс Order), отслеживания статуса заказа и расчета общей стоимости доставки. В системе должны быть предусмотрены различные виды доставки (например, курьерская доставка, почтовая доставка) с разными ставками и условиями. Класс Delivery должен также предоставлять методы для вывода текущего состояния заказов и доставок в терминал.

13. Реализуйте класс JewelryStore (Ювелирный магазин), который представляет собой систему управления продажей ювелирных изделий. Класс должен иметь методы для добавления нового изделия, удаления изделия, поиска изделия по названию, списка изделий где цена \geq заданной и расчета общей стоимости всех изделий в магазине. В системе должны быть предусмотрены различные типы ювелирных изделий (например, кольца, ожерелья, серьги) с разными ценами и свойствами. Класс JewelryStore должен также предоставлять методы для вывода текущего состояния изделий и информации о продажах в терминал.

14. Реализуйте класс Machine (Машина), который представляет собой систему управления производством механических устройств. Класс должен иметь методы для добавления нового устройства, удаления устройства, поиска устройства по модели, инвентарному номеру и расчета общей стоимости всех устройств в системе. В системе должны быть предусмотрены различные типы механических устройств (например, двигатели, насосы, турбины) с разными параметрами и ценами. Класс Machine должен также предоставлять методы для вывода текущего состояния устройств и информации о производстве в терминал. Предусмотрите случаи, когда может генерироваться исключение, например, при попытке добавить устройство с некорректными параметрами или при поиске устройства, которого нет в системе.

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: