

Создать консольное приложение, удовлетворяющее следующим требованиям:

1. Использовать возможности ООП: классы, наследование, полиморфизм, инкапсуляция.
2. Каждый класс должен иметь исчерпывающие смысл, название и информативный состав.
3. Наследование должно применяться только тогда, когда это имеет смысл.
4. При кодировании должны быть использованы соглашения об оформлении кода java code convention.
5. Классы должны быть грамотно разложены по пакетам.
6. Предусмотреть проверку корректности обрабатываемых данных и операций над ними.
7. Объекты хранить в коллекциях (при необходимости – в картах).
8. Сведения для инициализации объектов должны храниться в файлах.
9. Реализовать возможность пользователю задать с консоли критерии поиска данных и просмотра результатов поиска на консоли.
10. Предусмотреть возможность сравнения 2-х объектов одного класса (переопределить метод equals).

1. Student: id, Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс, Группа.

Создать набор объектов. Вывести:

- a) список студентов заданного () факультета;
- b) списки студентов для каждого факультета и курса;
- c) список студентов, родившихся после заданного года;
- d) список учебной группы.

2. Customer: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета.

Создать набор объектов. Вывести:

- a) список покупателей в алфавитном порядке;
- b) список покупателей, у которых номер кредитной карточки находится в заданном интервале.

3. Patient: id, Фамилия, Имя, Отчество, Адрес, Телефон, Номер медицинской карты, Диагноз.

Создать набор объектов. Вывести:

- a) список пациентов, имеющих данный диагноз;
- b) список пациентов, номер медицинской карты у которых находится в заданном интервале.

4. Abiturient: id, Фамилия, Имя, Отчество, Адрес, Телефон, Оценки.

Создать набор объектов. Вывести:

- a) список абитуриентов, имеющих неудовлетворительные оценки;
- b) список абитуриентов, средний балл у которых выше заданного;
- c) выбрать заданное число n абитуриентов, имеющих самый высокий средний балл.

5. Book: id, Название, Автор(ы), Издательство, Год издания, Количество страниц, Цена, Переплет.

Создать набор объектов. Вывести:

- a) список книг заданного автора;
- b) список книг, выпущенных заданным издательством;
- c) список книг, выпущенных после заданного года.

6. House: id, Номер квартиры, Площадь, Этаж, Количество комнат, Улица, Тип здания, Срок эксплуатации.

Создать набор объектов. Вывести:

- а) список квартир, имеющих заданное число комнат;
- б) список квартир, имеющих заданное число комнат и расположенных на этаже, который находится в заданном промежутке;
- с) список квартир, имеющих площадь, превосходящую заданную.

7. Phone: id, Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Дебет, Кредит, Время городских и междугородных разговоров.

Создать набор объектов. Вывести:

- а) сведения об абонентах, у которых время внутригородских разговоров превышает заданное;
- б) сведения об абонентах, которые пользовались междугородной связью;
- с) сведения об абонентах в алфавитном порядке.

8. Car: id, Марка, Модель, Год выпуска, Цвет, Цена, Регистрационный номер.

Создать набор объектов. Вывести:

- а) список автомобилей заданной марки;
- б) список автомобилей заданной модели, которые эксплуатируются больше n лет;
- с) список автомобилей заданного года выпуска, цена которых больше указанной.

9. Product: id, Наименование, UPC, Производитель, Цена, Срок хранения, Количество.

Создать набор объектов. Вывести:

- а) список товаров для заданного наименования;
- б) список товаров для заданного наименования, цена которых не превосходит заданную;
- с) список товаров, срок хранения которых больше заданного.

10. Train: Пункт назначения, Номер поезда, Время отправления, Число мест (общих, купе, плацкарт, люкс).

Создать набор объектов. Вывести:

- а) список поездов, следующих до заданного пункта назначения;
- б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
- с) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

11. Bus: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег.

Создать набор объектов. Вывести:

- а) список автобусов для заданного номера маршрута;
- б) список автобусов, которые эксплуатируются больше 10 лет;
- с) список автобусов, пробег у которых больше 100000 км.

12. Airlines: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели.

Создать набор объектов. Вывести:

- а) список рейсов для заданного пункта назначения;
- б) список рейсов для заданного дня недели;
- с) список рейсов для заданного дня недели, время вылета для которых больше заданного.

13. Создать суперкласс Транспортное средство и подклассы Автомобиль, Велосипед, Повозка. Подсчитать время и стоимость перевозки пассажиров и грузов каждым транспортным средством.

14. Создать суперкласс Пассажироперевозчик и подклассы Самолет, Поезд, Автомобиль. Задать правила выбора транспорта в зависимости от расстояния и наличия путей сообщения.
15. Создать суперкласс Учащийся и подклассы Школьник и Студент. Определить способы обучения и возможности его продолжения.
16. Реализовать систему «калькулятор» для обработки арифметических операций ("+", "-", "*", "/", "(", ")", "0"-"9", ".") с точностью до 3-х знаков после запятой.
17. Реализовать систему «Операции с матрицами» (сложение, умножение, умножение на число, транспонирование, вычисление определителя).
18. Реализовать систему «Операции с векторами на плоскости, заданными в координатной форме» (Сложение двух векторов, Сложение нескольких векторов, Умножение вектора на число, определение угла между векторами, проверка 2-х векторов на коллинеарность).
20. Система «КВАДРАТНОЕ УРАВНЕНИЕ». Создать класс «Квадратное уравнение». Поля класса – коэффициенты уравнения, корни уравнения; методы класса – расчет корней. Предусмотреть конструктор для инициализации полей класса. В конструкторе аналитически определить отрезки, на котором существуют корни, в методах реализовать получение корней методами дихотомии, простых итераций, а также при помощи дискриминанта. Если корней не существует, после создания объекта класса должно отображаться соответствующее сообщение. При необходимости в класс можно добавлять поля, методы, конструкторы либо инициализаторы. Предусмотреть проверку корректности обрабатываемых данных и операций над ними.
21. Система «ТРАНСПОРТНОЕ СРЕДСТВО». Создать класс «транспортное средство». Определить поля «максимальная скорость», «расход топлива на 100 км» и «ёмкость бака». Предусмотреть конструктор для создания объекта, методы вычисления расхода топлива на заданное расстояние (если полученный расход превышает ёмкость бака, вывести значение ёмкости бака) и вычисления скорости по заданному расстоянию и времени в пути (если полученная скорость превышает максимальную, вывести значение максимальной скорости). Создать класс «легковой автомобиль», наследующий от класса «транспортное средство». Определить поля «модель», «цвет», «количество посадочных мест». Предусмотреть для их инициализации конструктор, а для работы с ними – соответствующие методы. Если количество посадочных мест превышает 9, объект класса не может быть создан. Создать класс «грузовой автомобиль», наследующий от класса «транспортное средство». Определить поля «тип автомобиля» (тягач, самосвал, для перевозки людей и пр.), «грузоподъёмность». Предусмотреть для их инициализации конструктор, а для работы с ними – соответствующие методы. При необходимости в классы можно добавлять поля, методы, конструкторы либо инициализаторы. Реализовать возможность добавления транспортных средств в список с сортировкой по максимальной скорости. Транспортные средства с одинаковой максимальной скоростью должны быть отсортированы по ёмкости бензобака.
22. Создать класс для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы индексов, возможность задания произвольных границ индексов при создании объекта и выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, печати (вывода на экран) элементов массива по индексам и всего массива. При необходимости в класс можно добавлять поля, методы, конструкторы либо инициализаторы.
23. Система ДОМАШНИЕ ЭЛЕКТРОПРИБОРЫ. Определить иерархию электроприборов. Создать коллекцию «Работающие приборы». Реализовать операции добавления/удаления в (из) неё

приборов (включить / выключить). Посчитать потребляемую мощность у включенных электроприборов. Провести сортировку приборов в квартире по их наименованию, затем - по мощности. Найти приборы в квартире, соответствующие каким-либо параметрам (задать самостоятельно).

24. Создать:

Класс "Точка", имеющий 2 координаты.

Реализовать:

- конструктор для инициализации координат;
- процедуру изменения координат точки на величину step.

Класс "Отрезок". Реализовать конструкторы:

- с параметрами 2 точки,
- с параметрами 2 пары координат.

Реализовать:

- функцию вычисления длины отрезка
- процедуру изменения начала и конца отрезка на величину step.

Интерфейс «Фигура» с абстрактными методами вычисления площади и периметра.

Класс "Прямоугольник", наследующий от класса "Отрезок" и реализующий интерфейс «Фигура».

Реализовать конструкторы:

- с параметрами 2 точки,
- с параметрами 2 пары координат.
- с параметрами 2 отрезка.

Реализовать:

- методы интерфейса;
- процедуру перемещения всех вершин прямоугольника на величину step

Создать класс "Круг", наследующий от класса "Отрезок" и реализующий интерфейс «Фигура».

Реализовать конструкторы:

- с параметром отрезок (радиус)
- с параметрами 2 точки: центр и точка на границе круга.

Реализовать:

- методы интерфейса;
- процедуру перемещения центра круга на величину step

Класс "Эллипс", наследующий от класса "Круг". Реализовать конструкторы:

- с параметрами 2 отрезка (радиусы)
- с параметрами 3 точки: центр и точки на границе эллипса (концы малого и большого радиусов).

Реализовать:

- методы интерфейса [площадь эллипса: $\pi \cdot R \cdot r$, периметр эллипса: $4 \cdot \pi \cdot D \cdot d + (D - d)^2 / (D + d)$];
- возможность добавления фигур в список с сортировкой по площади и периметру.

При необходимости в систему можно добавлять классы, поля, конструкторы либо инициализаторы.

25. Разработать иерархию цветов. Создать несколько объектов-цветов. Собрать букет (используя аксессуары) с определением его стоимости. Провести сортировку цветов в букете на основе уровня свежести; по их названиям. Найти цветы в букете, соответствующие заданному диапазону длин стеблей. Найти количество цветов заданного типа в букете (розы, гвоздики и пр. – вводится пользователем).

26. Реализовать класс "Расписание занятий", имеющий следующие поля: время, предмет, аудитория.

Составить расписание на неделю с использованием подходящих структур данных.

Вывести на консоль это расписание, отсортированное по дням недели, с занятиями, отсортированными внутри каждого дня недели по времени начала.

Реализовать поиск в расписании по различным критериям (задаётся пользователем – не менее 2-х).

При необходимости в класс можно добавлять дополнительные члены, конструкторы либо инициализаторы.

27. Напишите консольную программу с именем FlightPlanner, которая считывает файл, содержащий пункты назначения рейсов из разных городов, а затем позволяет пользователю планировать маршрут полета в оба конца.

Данные о рейсе поступают из файла с именем flight.txt, который имеет формат, как показано ниже:

Сан-Хосе -> Сан-Франциско

Сан-Хосе -> Анкоридж

Нью-Йорк -> Анкоридж

Нью-Йорк -> Сан-Хосе

Нью-Йорк -> Сан-Франциско

Нью-Йорк -> Гонолулу

Анкоридж -> Нью-Йорк

Анкоридж -> Сан-Хосе

Гонолулу -> Нью-Йорк

Гонолулу -> Сан-Франциско

Денвер -> Сан-Хосе

Сан-Франциско -> Нью-Йорк

Сан-Франциско -> Гонолулу

Сан-Франциско -> Денвер

Программа должна:

- Читать информацию о рейсе из файла flight.txt и сохранять её в соответствующей структуре данных.

- Показывать полный список городов, отсортированный сначала по месту отправления, затем – по месту назначения.

- Разрешить пользователю выбрать город вылета.

- Печатать все пункты назначения, которые пользователь может достичь непосредственно из текущего города, и предложить пользователю выбрать город прилёта.

- Как только пользователь выберет маршрут, распечатать его на консоли.

Приложение должно корректно работать с кириллицей.

28. В учреждении на балансе находятся следующие виды оргтехники: Планшеты, ПК, ноутбуки. У всех видов есть общие атрибуты: процессор (Intel, AMD), объем оперативной памяти, объем жесткого диска, размер экрана, тип (Планшет, ПК, ноутбук). У каждого типа техники есть собственные характерные атрибуты: Планшет - линейные размеры; ПК - видеокарта(g-force, radeon); Ноутбук - наличие CD-ROM (есть/нет);

Оргтехника находится в разных отделах: отдел техподдержки, снабжения, сбыта.

В каждом отделе не менее 5 различных видов оргтехники.

Написать программу, выводящую на консоль список отделов, отсортированный по именам, причём по каждому отделу выводить количество оргтехники, по видам, например:

отдел сбыта: Планшет - 4 шт, ПК - 2 шт, ноутбук - 1 шт.

отдел снабжения: ноутбук - 5 шт, ПК - 1 шт, Планшет - 1 шт.

отдел техподдержки: ПК - 4 шт, Планшет - 2 шт, ноутбук - 1 шт.

Списки оргтехники должны быть отсортированы по количеству оргтехники. В случае, если количество одинаковое - по названию.

При создании программы корректно пользоваться всеми принципами ООП; использовать абстракцию, именованные константы и т. д.

29. Напишите класс "игральная карта", имеющий 3 поля - ранг, масть и козырь. Значения ранга и масти реализовать в виде именованных констант (Enum).

В классе создать метод с принимающий другой объект этого класса и сравнивающий данную карту с ним. В методе вывести на консоль "карты равны" если ранг и масть карт совпадают. Если совпадают только масти карт, вывести "старшая карта: " с указанием старшей карты. Если масти карт не совпадают, карты не считаются равными (вывести «карты не равны»). Если один из аргументов метода - козырная карта, её автоматически считать старшей, с выводом на консоль соответствующей информации. Если обе козырные - сравнивать их по рангу.

Создать коллекцию карт. Отсортировать в ней карты по мастям и в порядке старшинства (козырная масть идёт в начале, затем – масти по алфавиту).

Дать возможность пользователю выбрать из колоды

- все карты заданной масти;
- все карты заданного ранга.

30. Написать программу, которая считывает текстовый файл в список. Имя файла вводится в консоли. Затем программа должна напечатать эти строки, отсортированные по длине. Если длина одинакова, использовать сортировку по алфавиту. Регистр букв не учитывается. Должна поддерживаться корректная сортировка кириллицы (буква ё идёт после е).