

# Задания по основам объектно-ориентированного программирования

Для решения задачи необходимо разработать подходящий класс или классы. Все элементы классов должны иметь минимально возможную область доступа. Каждый класс должен содержать конструкторы, методы для изменения значений полей, методы для работы с полями, в зависимости от задачи. Каждый метод класса должен выполнять только одну операцию. Во всех случаях, когда возможно использование библиотечных классов Java API, они должны быть использованы. При чтении и записи данных имя входного файла можно задавать в тексте программы.

Все ошибочные ситуации должны корректно обрабатываться программой! Если во входном файле встречается строка с ошибкой, соответствующие данные должны быть проигнорированы с сообщением в консоли о найденной ошибке, программа должна продолжить работу со всеми корректными данными. Если без этих данных продолжение работы невозможно, то необходимо завершить работу программы с соответствующим сообщением в консоли.

Файл с данными может быть в формате CSV (Comma-Separated Values). Это текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит значения ячеек таблицы, разделенные запятыми. Если исходные данные ячейки содержат запятые, то эти данные заключаются в двойные кавычки. Дополнительно об обработке файлов в формате CSV можно прочитать в статье Как читать и анализировать CSV-файл в Java.

## Оценка 3

Для получения оценки 3 задача должна быть решена полностью.

**Задача 1.** Создайте класс для хранения и обработки информации о треугольнике. Треугольник задаётся длинами трёх сторон. Создайте методы для изменения полей класса, проверяйте неравенство треугольника. Создайте методы для определения периметра, площади, углов треугольника. Создайте методы для проверки, является ли треугольник прямоугольным, равнобедренным или равносторонним.

В главном классе создайте два треугольника, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом треугольнике, в том числе периметр, площадь и углы. Проверьте, являются ли треугольники прямоугольными, равнобедренными или равносторонними. Результат выведите на консоль.

**Задача 2.** Создайте класс для хранения информации о прямоугольном треугольнике. Полями класса являются длины катетов. Создайте методы, которые вычисляют гипотенузу, градусную меру, синус, косинус, тангенс, и котангенс для каждого острого угла, радиусы вписанной и описанной окружности. Создайте метод для сравнения треугольников на равенство.

В главном классе создайте два треугольника, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом треугольнике и результаты работы методов. Проверьте, являются ли треугольники равными. Результат выведите на консоль.

**Задача 3.** Создайте класс для хранения и обработки информации о комплексных числах. Комплексное число задаётся парой чисел: действительной и мнимой частью. Создайте методы для опре-

деления модуля числа, суммы, разности, произведения, частного комплексных чисел. Информация о комплексном числе должна выводиться в виде  $5 + 3i$ , кроме того, должен быть метод выводющий на консоль комплексное число в тригонометрической форме.

В главном классе создайте два комплексных числа, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом числе, в том числе модуль, найдите сумму, разность, произведение, частное комплексных чисел. Результат выведите на консоль.

**Задача 4.** Создайте класс для хранения и обработки информации о дробях. Дробь задается числителем и знаменателем (целые числа), знаменатель не должен быть равен нулю. Создайте методы для определения суммы, разности, произведения, частного дробей. После выполнения операций сокращайте дробь, полученную в результате, если это возможно.

В главном классе создайте две дроби, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждой дроби, найдите сумму, разность, произведение, частное дробей. Результат выведите на консоль.

Реализуйте сравнение дробей и выведите результат сравнения для двух созданных объектов.

**Задача 5.** Создайте класс для хранения и обработки информации о многоугольниках. Многоугольник задаётся количеством углов и координатами этих углов. Создайте методы для определения периметра многоугольника, является ли многоугольник правильным.

В главном классе создайте два многоугольника, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом многоугольнике. Проверьте, является ли каждый из многоугольников правильным треугольником или квадратом.

**Задача 6.** Создайте класс или классы для хранения и обработки информации о конкурсе. Конкурс содержит информацию о премиальном фонде и массив заявок. У заявки есть номер и экспертная оценка от 0 до 100 баллов. Создайте метод для определения победителей по следующему правилу: победителями считаются все, у кого экспертная оценка больше средней арифметической оценки всех заявок. Создайте метод для распределения премиального фонда среди победителей пропорционально экспертным оценкам.

В главном классе создайте два конкурса, информацию о каждом прочитайте из файла или с консоли: сначала размер премиального фонда, затем количество заявок и их экспертные оценки. Выведите на консоль информацию о победителях и премиях каждого конкурса.

**Задача 7.** Создайте класс или классы для хранения и обработки информации о книге в библиотеке. Информация о книге содержит автора, название, количество книг в библиотеке. Создайте методы для вывода информации о книге и изменения количества книг в библиотеке.

В главном классе создайте массив или список книг, количество объектов и параметры прочитайте из файла или с консоли. Организуйте цикл выдачи книг читателям. Читатель вводит название или автора, программа сообщает о наличии книги, если подходящих несколько выводит их пронумерованный список и предлагает дополнительно выбрать книгу по номеру. После выбора, поле с количеством книг соответствующего объекта уменьшается. Если подходящей книги нет или количество равно нулю, выдаётся сообщение.

**Задача 8.** В файле `data_comets.csv` содержится информация о кометах, о каждой комете в отдельной строке. Формат строки следующий: название в формате номеримя, эксцентриситет (степень вытянутости орбиты), угол наклона плоскости орбиты кометы к плоскости эклиптики, размер кометы, перигелий (минимальное расстояние от Солнца), афелий (максимальное расстояние от Солнца), период обращения в годах.

Создайте класс для хранения данных о комете. Из данных файла сформируйте список комет. По выбору пользователя выведите список отсортированный по имени, размеру или периоду обращения.

## Оценка 3 — 4

Для получения оценки 3 нужно решить всё, кроме дополнительного задания. Для получения оценки 4 задача должна быть решена полностью.

**Задача 9.** Создайте класс для хранения и обработки информации об  $n$ -мерном векторе. Создайте методы для определения модуля вектора, умножения вектора на число, суммы, разности, скалярного произведения векторов, угла между векторами.

В главном классе создайте два вектора, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом векторе и его модуль. Найдите сумму, разность, скалярное

произведение векторов, угла между векторами. Результат выведите на консоль.

Дополнительно реализуйте методы для определения ортогональности векторов, нормирования вектора и выведите результат работы для двух созданных объектов. Реализуйте сравнение векторов по длине и выведите результат сравнения для двух созданных объектов.

**Задача 10.** Создайте классы для хранения данных об автобусных маршрутах и автобусах. Первый входной файл содержит количество и список маршрутов автобусов. Сведения о каждом маршруте включают: номер, длину в километрах (не обязательно целое количество), количество автобусов, необходимых для обеспечения движения по маршруту. Второй входной файл содержит количество и список автобусов. Сведения о каждом автобусе содержат: номер автобуса, расход бензина (в литрах на километр), номер маршрута.

Программа должна читать данные из входного файла и выводить на консоль для каждого маршрута список автобусов, сообщение, достаточно ли у маршрута автобусов (“не хватает”, “достаточно” или “больше, чем нужно”), сколько литров бензина суммарно тратят все автобусы, проехав по маршруту один раз.

Дополнительно реализуйте возможность перераспределения автобусов с маршрутов, где их больше, чем нужно на те, где автобусов не хватает. Автобус перемещается на другой маршрут максимально близкий по длине к первоначально указанному, при этом соответствующее поле с номером маршрута изменяет своё значение. Результат перераспределения выводится на консоль.

**Задача 11.** Входной файл содержит информацию о людях: имя, возраст, тип (формалист, реалист или неформал). Любой человек способен выполнять два действия: здороваться с другим человеком; рассказывать о себе. Формат входного файла: в первой строке количество людей, далее в каждой строке задаются данные об одном человеке: имя, возраст, тип, разделённые произвольным количеством пробелов.

Формалисты здороваются со всеми так: «Здравствуй, <имя>», где <имя>— имя человека, с которым он здоровается. Неформалы со всеми здороваются: «Привет, <имя>!». Реалисты, если возраст собеседника меньше или равен возрасту самого реалиста или больше своего возраста не более чем на 5 лет, говорит «Привет, <имя>!», иначе «Здравствуй, <имя>». Рассказ о человеке является строкой вида «Меня зовут <имя>, мой возраст <число> лет, я <тип>».

Создайте класс для хранения данных о человеке и содержащий методы, реализующие описанные действия. Программа должна читать данные из входного файла и выводить на консоль рассказы о всех людях. Затем все люди должны поздороваться друг с другом в таком порядке: первый здоровается со вторым, потом второй с первым, потом первый с третьим, третий с первым, и так далее первый со всеми и все с первым, потом второй с третьим, третий со вторым и т.д. Нужно записать в отдельных строках имя человека, который здоровается, двоеточие, приветствие, например:

Петя: Привет, Вася!

Вася: Здравствуй, Петя!

Дополнительно реализуйте сравнение людей тремя способами: по имени, по возрасту, по типу и имени вместе. Выведите списки людей отсортированные всеми тремя способами.

**Задача 12.** Входной файл содержит информацию о стрелках: имя, возраст, стаж обучения стрельбе (в годах). Формат входного файла: файл содержит в первой строке количество стрелков, далее в каждой строке задаются данные об одном стрелке: имя, возраст, стаж обучения, разделённые произвольным количеством пробелов.

Люди делятся на новичков (стаж обучения не более трех лет), опытных (стаж обучения от четырех до восьми лет) и ветеранов (стаж обучения более восьми лет). Попадание при совершении выстрела определяется случайным образом, причем для новичка вероятность попасть равна  $0,01 \cdot \text{стаж обучения}$ ; для опытного  $= 0,05 \cdot \text{стаж обучения}$ ; для ветерана  $= 0,9 - 0,01 \cdot \text{возраст}$ .

Создайте класс для хранения данных о стрелке. Методы класса определяют попадание при выстреле с помощью датчика случайных чисел и на основе значений полей.

Программа должна читать данные из входного файла и выводить на консоль результаты стрельб, где каждый человек делает по 10 выстрелов, в виде таблицы с информацией о попадании каждого человека на каждом выстреле и итоговым количеством попаданий, а также информацию о самом метком стрелке.

Дополнительно выведите информацию о трёх самых лучших стрелках каждой группы: новичков, опытных и ветеранов. Реализуйте сравнение людей по имени и возрасту и выведите список людей отсортированный по этому способу.

**Задача 13.** Входной файл содержит информацию о студентах: имя, тип (добросовестный, сообразительный, гений), количество посещенных занятий. Формат входного файла: файл содержит в первой строке количество студентов, далее в каждой строке задаются данные об одном студенте: имя, тип, количество посещенных занятий, разделённые произвольным количеством пробелов. Студенты сдают зачет по предмету по которому было проведено 36 занятий. Добросовестные студенты точно сдают зачет, если посетили все занятия, если были более чем на половине занятий, то сдают с вероятностью 0,5; иначе не сдают. Сообразительные студенты тоже точно сдают зачет, если посетили все занятия, если были более чем на половине занятий, то сдают с вероятностью 0,7, иначе не сдают зачет. Гении точно сдают зачет, если были хотя бы на одном занятии, иначе не сдают.

Создайте класс для хранения данных о студенте. Методы класса определяют результат зачёта с помощью датчика случайных чисел и на основе значений полей.

Программа должна читать данные из входного файла и выводить на консоль информацию о всех студентах, затем информацию о сдавших зачет и отдельно информацию о несдавших зачет,

Дополнительно реализуйте сравнение студентов двумя способами: по имени и по количеству посещенных занятий. Выведите список студентов отсортированный по имени. Выведите имя студента, посетившего максимальное количество занятий, но несдавшего зачет. По запросу пользователя выведите информацию о студентах заданного типа несдавших зачет.

**Задача 14.** В файле `data_bestsellers.txt` содержится информация о книгах. Формат файла следующий: в каждых семи строках записаны: название, автор, рейтинг, количество отзывов, цена, год издания, жанр (Fiction или Non Fiction).

Создайте класс для хранения данных о книге. Создайте массив или список книг и определите диапазон значений года издания. По заданному пользователем году издания и жанру выведите список книг, отсортированный по рейтингу.

Дополнительно создайте класс для хранения информации о книгах писателя. Из данных файла сформируйте список писателей и выведите о каждом из них информацию в отдельный файл.

**Задача 15.** В файле `data_country.csv` содержится информация о странах, о каждой стране в отдельной строке. Формат строки следующий: название, показатель расходов на здравоохранение, показатель доходов, показатель инфляции, показатель продолжительности жизни.

Создайте класс для хранения данных о стране. Выведите на консоль список стран со всеми данными, отсортированный по параметру, указанному пользователем.

Дополнительно сформируйте и выведите в отдельные файлы матрицы из названий стран. Первая: два столбца, в одном список стран, отсортированный по убыванию расходов на здравоохранение, в другом — список тех же стран, отсортированный по убыванию продолжительности жизни. Вторая: три столбца, каждый со списком стран, упорядоченных по возрастанию доходов, инфляции, продолжительности жизни.

**Задача 16.** В файле `data_human_development.csv` содержится информация о показателях человеческого развития в странах, о каждой стране в отдельной строке. Формат строки следующий: название, индекс человеческого развития, средняя продолжительности жизни, ожидаемое количество лет обучения, реальное среднее количество лет обучения, валовой национальный доход на душу населения в тысячах долларов.

Создайте класс для хранения данных о стране. Выведите на консоль список самых богатых стран (количество стран, которое надо выбрать вводит пользователь), отсортированный по возрастанию разности между ожидаемым количеством лет обучения и реальным средним.

Дополнительно сформируйте и выведите в отдельный файл матрицу из названий стран. В матрице должно быть два столбца, в каждом список всех стран, отсортированных по возрастанию параметра. Два параметра для сортировки задаёт пользователь.

**Задача 17.** В файле `data_marathon.csv` содержится информация о победителях марафонов, о каждом победителе в отдельной строке. Формат строки следующий: год забега, имя победителя, пол (Male — мужской, Female — женский), страна — родина победителя, время пробега марафонской дистанции (часы:минуты:секунды), город, где проводился марафон.

Создайте класс для хранения данных о марафонцах-победителях страны. Данные должны содержать название страны, список победителей. Для хранения данных о победителе создайте отдельный класс. Выведите на консоль список стран, отсортированный по возрастанию количества победителей. По году, заданному пользователем, выведите на консоль список победителей с указанием родной страны и пола.

Дополнительно выведите в текстовый файл список стран, упорядоченный по алфавиту. Для каждой страны укажите наименьшее время пробега марафона у женщин и мужчин, среднее время пробега всех победителей страны. Создайте класс для хранения информации о победителях в каждом году. Класс должен содержать год, список городов, где проходили марафоны. Город хранит информацию о его названии и победителях.

**Задача 18.** В файле `data_medicalDoctors.csv` содержится информация о количестве врачей в странах мира. Формат строки следующий: название страны, год, среднее количество врачей на 10 000 человек.

Создайте класс для хранения данных о стране: название и информация о количестве врачей в каждом известном году. Из данных файла сформируйте список стран. Выведите на консоль список стран, отсортированных по возрастанию количества врачей в году, заданному пользователем. В список включите только те страны, о которых есть информация по заданному году.

Дополнительно выведите в текстовый файл (можно в формате `csv`) информацию о количестве врачей в странах в виде таблицы: по строкам название страны, по столбцам — годы (1990 — 2018), в ячейке таблицы, соответствующее количество врачей. Если информация отсутствует, то ставится прочерк.

**Задача 19.** В файле `dataPainter.txt` содержится информация о художниках. Формат файла следующий: в каждых семи строках записаны: индекс, имя, годы жизни через пробелы и тире, стиль картин или стили через запятую, родная страна, страница о художнике в википедии, количество картин.

Создайте класс для хранения данных о художнике. Создайте список художников и выведите его сначала отсортировав по количеству картин, затем по продолжительности жизни. По заданному пользователем стилю выведите список художников, относящихся к этому стилю.

Дополнительно создайте класс для хранения информации о стране со списком художников, относящихся к этой стране. По заданному пользователем диапазону лет, выведите на консоль список художников, живших в этот период. В текстовый файл выведите таблицу с именами художников по странам и стилям.

## Оценка 4

Для получения оценки 4 задача должна быть решена полностью.

**Задача 20.** В файле `data_AUS_state.csv` содержится информация о городах Австралии, о каждом городе в отдельной строке. Формат строки следующий: название города, название штата или территории, к которым относится город, количество жителей в 2017 году, количество жителей в 2011 году (целые числа, в которых разряды разделены запятыми), сколько аборигенов проживает в городе (доля в процентах от общей численности аборигенов Австралии — число со знаком процента). Аборигены составляют 2,5 % от всех жителей Австралии.

Создайте классы для хранения данных о городе и о штате (территории). Штат должен содержать список городов. Выведите на консоль список городов каждого штата. Для каждого города укажите прирост населения в 2017 году в процентах по сравнению с 2011 и примерное число аборигенов, живущих в этом городе.

**Задача 21.** В файле `data_austin_weather.csv` содержится информация о погоде на морском курорте за три года 2014, 2015, 2016, о каждом дне года в отдельной строке. Формат строки следующий: дата (год-месяц-день), температура по Фаренгейту, влажность в процентах, видимость в милях, скорость ветра в милях в час, количество осадков в дюймах, атмосферные явления (Rain — дождь, Thunderstorm — гроза, Fog — туман, Snow — снег).

Эквивалентно-эффективная температура (ЭЭТ) учитывает комплексное влияние на человека температуры, влажности воздуха и скорости ветра. Расчеты ЭЭТ производятся по формуле А. Миссенарда и оцениваются по уровням комфорта.

Создайте класс или классы для хранения данных о погоде, например, отдельный класс для данных ежедневной погоды и отдельный класс для хранения данных за год. Определите в каждом из трёх лет месяц с наибольшим количеством комфортных для человека дней. Для заданного пользователем месяца и года выведите количество дней для каждого уровня комфорта.

**Задача 22.** В файле `data_books.csv` содержится информация о книгах, о каждой книге в отдельной строке. Формат строки следующий: название, автор, рейтинг, код языка, количество страниц,

дата публикации, издательство.

Создайте класс для хранения данных о книге: название, автор, рейтинг, код языка, количество страниц, дата публикации. Создайте класс издательство с названием и списком книг. Сформируйте из файла список издательств. Выведите информацию об издательстве с самым длинным списком книг, отсортированных по дате издания в отдельный файл. По заданному пользователем году выведите название издательства с самым большим средним рейтингом книг, изданных в этом году.

**Задача 23.** В файле `data_books.csv` содержится информация о книгах, о каждой книге в отдельной строке. Формат строки следующий: название, автор, рейтинг, код языка, количество страниц, дата публикации, издательство.

Создайте класс для хранения данных о книге: название, рейтинг, количество страниц, издательство, дата публикации. Создайте класс писатель с именем и списком книг. Сформируйте из файла список писателей и выведите информацию о писателе с самой толстой книгой в отдельный файл, список книг отсортируйте по рейтингу. Для заданного пользователем имени писателя выведите список названий издательств, в которых были опубликованы его книги.

**Задача 24.** В файле `data_comets.csv` содержится информация о кометах, о каждой комете в отдельной строке. Формат строки следующий: название в формате номеримя, эксцентриситет (степень вытянутости орбиты), угол наклона плоскости орбиты кометы к плоскости эклиптики, размер кометы, перигелий (минимальное расстояние от Солнца), афелий (максимальное расстояние от Солнца), период обращения в годах.

Создайте класс для хранения данных о комете. Из данных файла сформируйте список комет.

В файле `data_comets_date.txt` содержится информация о дате прохождения кометой перигелия, о каждой комете в отдельной строке. Формат строки следующий: номер, имя на русском языке, дата (год-месяц-день), разделенные пробелами. Найдите в списке кометы, для которых известна дата прохождения перигелия, выведите о них информацию на консоль: название и примерную дату следующего прохождения перигелия.

**Задача 25.** В файле `data_women_chess_players.csv` содержится информация о женщинах шахматистках, о каждой шахматистке в отдельной строке. Формат строки следующий: фамилия и имя через запятую и пробел, код страны, год рождения, звание (FIDE titles), рейтинг.

Создайте класс для хранения данных о шахматистках каждой страны. Для страны надо хранить код и список шахматисток. По данным файла сформируйте список стран. По заданному пользователем званию выведите на консоль список шахматисток каждой страны, упорядоченный по рейтингу, затем список, упорядоченный по возрасту.

**Задача 26.** Создайте классы для хранения и обработки информации о слушателе курсов английского языка. Слушатель может записаться на один из трёх уровней курсов: базовый, продвинутый, углублённый. Занятия оплачиваются ежемесячно по следующим правилам. В каждом месяце проходит 8 занятий. Стоимость занятия продвинутого уровня на 25 % больше, чем базового, а углублённого на 20 % больше, чем продвинутого. Слушатель может оплатить занятия за месяц вперёд и получить скидку 7 % или оплатить все посещённые занятия в конце месяца по полной стоимости. При оплате заранее, если слушатель пропустил более половины занятий, он может вернуть часть денег: 80 % от стоимости пропущенных занятий. Но для углублённого уровня оплата производится только в конце месяца и только за посещённые занятия. Создайте класс для хранения данных об уровне с его названием и расчётом стоимости на основе информации о посещаемости занятий за месяц, например, в виде массива из 0 и 1. Класс слушатель должен хранить имя, уровень курсов, вид оплаты и посещаемость за месяц.

В главном классе прочитайте с консоли или из файла информацию о стоимости занятия базового уровня. Затем, введите информацию о пользователе: имя, уровень. Узнайте количество месяцев, которое слушатель будет посещать курсы (от 1 до 5), для каждого месяца запросите информацию о способе оплаты, в начале или в конце, и данные о посещаемости. Выведите на консоль стоимость курсов за каждый месяц, если часть денег возвращается, то выводите фактическую стоимость, полную стоимость и сумму возврата.

**Задача 27.** Создайте класс для хранения данных об абитуриенте и класс приёмной комиссии, содержащий массив абитуриентов и реализующий методы заполнения массива и определения списка поступивших.

Входной файл содержит информацию об абитуриентах, поступающих на некоторое направление: имя, тип конкурса (целевой, бюджетный, внебюджетный), количество баллов по трём предметам (от 0 до 100): математика, русский язык, физика. Формат входного файла: файл содержит в первой

строке количество абитуриентов, далее в каждой строке задаются данные об одном абитуриенте: имя, тип, три числа (баллы).

Абитуриент поступает на направление по следующим правилам:

1. для всех типов конкурсов баллы по предметам не ниже минимальных (математика — 39, русский язык — 40, физика — 37),
2. на целевые места проходят все, кроме одного набравшего минимальную сумму баллов,
3. определяется максимальная сумма баллов среди всех абитуриентов и на бюджетные места проходят все, у кого сумма баллов больше 75 % от этой максимальной суммы баллов,
4. на внебюджетные места проходит количество абитуриентов равное количеству поступивших на бюджетные места, если количество поступающих на внебюджет больше, выбираются те, у кого сумма баллов больше, если сумма баллов одинаковая, в виде исключения берут всех

Программа должна читать данные из входного файла и выводить на консоль информацию о всех абитуриентах, затем информацию о поступивших.

**Задача 28.** Создайте классы для организации музыкальных радиопрограмм. В файле хранится информация о десяти классических музыкальных произведениях, десяти джазовых, десяти популярных российских песнях, десяти популярных зарубежных песнях, десяти в стиле рок. О каждом произведении известно название, длительность в минутах (от 2 до 6) и рейтинг - целое неотрицательное число. Из этих произведений необходимо создавать два типа передач: хит-парад и по заявкам. Каждая передача длится 30 минут и включает музыкальные произведения и рекламу. Реклама вставляется в начало и середину программы, дополняя ее до нужных 30 минут, но не должна занимать более 6 минут суммарно. Хит-парад формируется из самых популярных произведений жанров в случайном порядке. Для формирования передачи по заявкам случайно генерируются заявки на музыкальные произведения, при этом к рейтингу, выбранного произведения добавляется единица.

В главном классе прочитайте из файла информацию о музыкальных произведениях и сформируйте по очереди заданное количество программ передач двух описанных типов. Результат выведите на консоль.

**Задача 29.** Создайте классы для подбора вакансий на бирже труда. Информация о вакансии содержит название должности, размер заработной платы, требования к возрасту (диапазон) и к образованию (высшее, среднее профессиональное, школьное среднее общее образование). Заявка работника содержит информацию об имени, возрасте, образовании, должности, минимальной заработной плате.

В главном классе прочитайте из файла информацию о количестве вакансий и самих вакансиях, из другого файла информацию о заявках работников. Подберите подходящие пары вакансия-заявка двумя способами: 1) в пользу работника, если ему подходят несколько вакансий, выбирается с наибольшей заработной платой, 2) в пользу работодателя, если ему подходят несколько работников, то самого молодого. Результат выведите на консоль.

**Задача 30.** Создайте классы для организации концертов. Первый входной файл содержит список исполнителей, которые могут приехать в КЗЦ (концертно-зрелищный центр) с концертом. Информация о каждом исполнителе включает его имя и процент от сборов с билетов, который исполнитель хочет получить за концерт. Второй входной файл содержит статистику о предыдущих концертах в КЗЦ. Для каждого концерта указывается имя исполнителя, и сумма сборов с билетов.

Администрация КЗЦ хочет провести 12 концертов и получить максимальную прибыль. Собирав деньги за билеты, она отдаёт процент исполнителю, остальное оставляет КЗЦ.

Напишите программу, которая считывает данные из входных файлов и выводит на консоль для каждого исполнителя среднюю сумму сборов с билетов за предыдущие выступления; сумму, которую он теоретически получит за концерт (исходя из его желаемого процента и средней суммы сборов) и сумму, которую КЗЦ теоретически получит за его концерт. На основании этих данных выведите список из 12 самых выгодных для КЗЦ концертов, упорядоченный по прибыли КЗЦ.

**Задача 31.** Создайте классы для подбора квартиры для покупки. Квартиры расположены в двух районах города. Для каждого района известен список улиц, который можно прочитать из файла в следующем формате: в первой строке улицы первого района через пробел, во второй — второго района. Информация о квартире содержит название улицы, количество комнат, цену. Заявка на покупку содержит информацию о количестве комнат, районе, максимальной цене.

В главном классе прочитайте из файла информацию о количестве квартир и самих квартирах, из другого файла информацию о заявках на покупку квартиры. Подберите подходящие пары квартира-заявка, если заявке подходят несколько квартир, выбирается с наименьшей ценой. Результат выведите на консоль.

**Задача 32.** Входной файл содержит список оценок школьника, полученных в течение четверти. Каждая запись об оценке содержит предмет, тип задания (самостоятельная, домашняя, контрольная) и балл (2, 3, 4, 5). Создайте классы для хранения и обработки информации о школьнике и его оценках. Методы должны выдавать списки оценок и среднюю оценку за каждый тип задания и формировать оценки за четверть по всем предметам. Список предметов может быть фиксированным и задаваться в программе или быть прочитан из файла. Оценка за четверть вычисляется как медиана всех оценок. Медиана — это такое число, что половина из элементов списка больше него, а другая половина меньше. Т.е., если в списке нечётное число элементов, то медиана — это число в середине упорядоченного по возрастанию списка. Если в списке чётное число элементов, то медиана — это среднее арифметическое двух элементов, стоящих в середине упорядоченного по возрастанию списка. Так как оценка за четверть должна быть целым числом, то среднее арифметическое нужно округлить по математическим правилам.

Программа должна читать данные из входного файла и выводить на консоль список оценок за четверть по предметам и оценки за задания.

**Задача 33.** Создайте класс для хранения и обработки информации об окружности. Окружность задаётся двумя координатами центра и величиной радиуса. Создайте методы для изменения полей класса. Создайте метод для определения взаимного расположения двух окружностей: внешнее и внутреннее касание и точку касания, две точки пересечения, отсутствие общих точек, метод для определения, попадает ли точка внутрь окружности. Реализуйте сравнение окружностей по длине радиуса. Информация об окружности должна выводиться в виде уравнения  $(x - x_0)^2 + (y - y_0)^2 = r^2$ .

В главном классе создайте две окружности, параметры прочитайте из файла или с консоли, и выведите на консоль информацию об окружностях, их взаимное расположение, результат сравнения. Введите координаты точки на плоскости и определите, находится ли она внутри одной из окружностей.

**Задача 34.** Создайте класс «трехмерная точка». Данные класса должны быть представлены тремя полями: координатами точки в трехмерном пространстве. Класс должен содержать конструктор по умолчанию, конструктор, инициализирующий координаты точки заданными значениями. Обязательными являются методы, возвращающие значения полей, изменяющие эти значения, метод выводющий точку на консоль в виде (1,-2,3). Создайте класс «трехмерный вектор», задаваемый двумя точками: начало и конец вектора. Класс должен содержать конструктор по умолчанию, конструктор, инициализирующий координаты точек заданными значениями. Обязательными являются методы, возвращающие значения полей, изменяющие эти значения, метод выводющий вектор на консоль в виде (1,0,3). Другие методы класса должны определять длину вектора, сумму, разность, скалярное произведение двух векторов, угол между векторами. Реализуйте сравнение векторов по длине.

В главном классе создайте два вектора, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом векторе и его модуль. Найдите сумму, разность, скалярное произведение векторов, угол между векторами, сравните вектора по длине. Результат выведите на консоль.

**Задача 35.** Создайте класс для хранения и обработки информации о точках и треугольниках на плоскости. Точка задаётся координатами, треугольник задаётся тремя точками, не лежащими на одной прямой. Создайте методы для изменения полей классов. Создайте методы для определения длин сторон, углов треугольника, сравнения треугольников по площади. Создайте методы для параллельного переноса треугольника со сдвигом по двум осям.

В главном классе создайте два треугольника, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом треугольнике, в том числе длины сторон. Проверьте, являются ли треугольники прямоугольными, тупоугольными или остроугольными. Результат выведите на консоль. Определите, какой из треугольников больше. Введите сдвиг по оси x и оси y и выведите информацию о треугольниках, получающихся из заданных соответствующим параллельным переносом.



## Оценка 4 — 5

Для получения оценки 4 нужно решить всё, кроме дополнительного задания. Для получения оценки 5 задача должна быть решена полностью.

**Задача 36.** Создайте класс для хранения и обработки информации о многочлене вида  $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ . Многочлен задается набором действительных коэффициентов. Создайте методы для изменения полей класса. Создайте методы для определения суммы и разности многочленов, вывода на консоль таблицы значений многочлена для аргументов в заданном диапазоне от  $a$  до  $b$  с шагом  $e$ .

В главном классе создайте два многочлена, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждом многочлене, их сумме, разности. Введите диапазон значений аргументов и шаг. Выведите таблицу значений каждого многочлена.

Дополнительно создайте методы для нахождения производной многочлена и произведения двух многочленов, результатом работы методов должен быть тоже многочлен. Выведите на консоль производную для каждого из двух многочленов, произведение этих многочленов.

**Задача 37.** Создайте классы для хранения и обработки информации о точках и прямых на плоскости. Информация о прямых хранится в виде двух точек, определяющих прямую. Создайте методы для изменения полей класса. Создайте методы для вывода уравнения прямой в виде  $y = kx + b$  или  $x = a$ , для определения, попадает ли точка на прямую. Создайте методы, определяющие являются ли две прямые перпендикулярными или параллельными.

В главном классе создайте две прямые, параметры прочитайте из файла или с консоли. Выведите на консоль информацию о каждой прямой. Определите, являются ли прямые параллельными или перпендикулярными. Результат выведите на консоль. Введите координаты точки и определите, принадлежит ли точка какой-нибудь прямой.

Дополнительно создайте методы для построения прямой параллельной данной и проходящей через заданную точку (проверяйте, не лежит ли заданная точка на прямой), прямой перпендикулярной данной и проходящей через заданную точку, для нахождения точки пересечения двух прямых.

**Задача 38.** Создайте класс для работы с квадратными матрицами. Информация о матрице хранится в виде двумерного массива действительных чисел. Создайте методы для изменения этого массива. Создайте методы для сложения и умножения матриц, вычисления определителя, вывода элементов матрицы на консоль.

В главном классе создайте две матрицы, данные о размере и элементах прочитайте из файла или с консоли. Выведите на консоль определитель, квадрат каждой матрицы, сумму, разность, произведение матриц.

Дополнительно создайте методы для построения обратной матрицы, если это возможно и для умножения матрицы на число и вектор. Продемонстрируйте работу этих методов.

**Задача 39.** Создайте класс для работы с квадратными матрицами. Информация о матрице хранится в виде двумерного массива однозначных целых чисел. Создайте методы для изменения отдельных элементов этого массива по индексам и заполнения массива случайными числами. Создайте методы для построения матрицы, полученной из исходной путём поворота на 90, 180 и 270 градусов против часовой стрелки, вывода элементов матрицы на консоль.

В главном классе создайте две матрицы, данные о размере и элементах первой прочитайте из файла или с консоли, вторую сформируйте из случайных чисел. Выведите на консоль исходные матрицы и матрицы, полученные поворотом на 90, 180 и 270 градусов.

Дополнительно создайте методы для определения, является ли исходная матрица магическим (суммы чисел по строкам, столбцам и диагоналям равны) и латинским квадратом (матрица состоит из чисел от 1 до  $n$ , все числа в строках и столбцах различны). Введите размер и элементы матрицы и проверьте, является ли она магическим и латинским квадратом, если нет, то предложите внести изменения и после этого снова проверьте.

**Задача 40.** Создайте класс или классы для работы с банковскими операциями. Входной файл содержит информацию о движении денежных средств между двумя счетами. Формат входного файла: файл содержит произвольное количество строк двух видов. Первый вид — строка, содержащая информацию о начальном состоянии счета: номер и сумма. Второй вид — информация о переводе денег с первого счета на второй: дата (три числа — день, месяц, год), счёт1, счёт2, сумма. В начале

файла расположены строки только первого типа, затем только второго, данные в каждой строке разделены произвольным количеством пробелов. Строки по датам неупорядочены.

Программа должна читать данные из входного файла и выводить на консоль информацию о состоянии всех счетов на текущий момент, упорядоченную по номеру счёта.

Дополнительно для заданного счёта и заданного периода времени (две даты) вычислите входящий остаток (сумму на счёте до начала периода), приход (все поступления на счёт в периоде), расход (все списания со счёта в периоде) и исходящий остаток (входящий + приход - расход). Проконтролируйте корректность операций по переводу денег со счёта на счёт, если переводится сумма больше текущего остатка, сообщите о невозможности операции и не учитывайте её в вычислениях. Для работы с датами используйте класс `LocalDate`.

**Задача 41.** Создайте классы для подбора авиарейса по заявке. Первый входной файл содержит список авиарейсов. Сведения о каждом рейсе содержат: номер, пункт отправления, пункт назначения, дату (день, месяц, год) и время вылета, стоимость билета. Второй входной файл содержит список заявок на рейсы. Сведения о каждой заявке содержат: пункт отправления, пункт назначения, дату (день, месяц, год), фамилию и имя пассажира.

Программа должна читать данные из входного файла и выводить на консоль список всех рейсов, отсортированных по пункту отправления, затем по пункту назначения (если пункты отправления совпадают). Для каждой заявки нужно вывести список всех подходящих рейсов. Рейс подходит по дате, если его дата совпадает с датой заявки или является более поздней датой по отношению к заявке. Если рейсов нет, об этом нужно вывести сообщение.

Дополнительно для заявки добавьте пожелание пассажира. Пожелание может быть двух типов: “хочу самый дешёвый рейс”, “хочу ближайший рейс”. Реализуйте подбор рейсов для заявки с учётом пожеланий. Для хранения даты и времени используйте класс `LocalDateTime`.

**Задача 42.** Создайте классы для организации работы ипподрома. В классе, описывающем ипподром, хранится информация о лошадях, например, в виде массива имён, информация о текущем забеге и игроках, сделавших ставки в этом забеге. Для забега случайным образом выбираются семь лошадей и случайным же образом определяется результат забега: в каком порядке лошади пришли к финишу. Информация об игроке содержит идентификационный номер, денежный размер ставки и имя лошади, которая должна придти первой. Из общей суммы ставок 20% забирает ипподром, остальные деньги распределяются между игроками, угадавшими победителя заезда, пропорционально их ставкам. Если победителя никто не угадал, все деньги достаются ипподрому.

В главном классе прочитайте из файла информацию о количестве и именах лошадей, сформируйте заезд и выведите о нём информацию. Введите с консоли информацию о ставках игроков, идентификационный номер каждого игрока назначается в программе. Результат заезда и выигрыш игроков выведите на консоль.

Дополнительно реализуйте возможность делать разные типы ставок: 1) имя лошади, пришедшей первой, 2) имя лошади, пришедшей последней, 3) последовательность из двух лошадей, занявших первое и второе место, где важен порядок лошадей, 4) последовательность из двух лошадей, занявших первое и второе место, где не важен порядок лошадей, 5) последовательность из трёх лошадей, занявших первое, второе и третье место, где важен порядок лошадей. При распределении выигрыша надо учесть правила: при одинаковых размерах ставки игрок угадавший двух первых лошадей в правильном порядке получает в 1,5 раза больше того, кто угадал победителя, игрок, угадавший трёх первых лошадей получает в 2 раза больше того, кто угадал победителя, игрок, угадавший двух первых лошадей без учёта порядка получает в 1,1 раза больше того, кто угадал победителя, игрок, угадавший лошадь, пришедшую последней получает 0,8 от соответствующего выигрыша того, кто угадал победителя. Если, выбирая две или три лошади, игрок угадал только победителя, то он получает 0,5 от выигрыша того, кто ставил и угадал одного победителя. Во всех остальных случаях выигрыш не начисляется. Реализуйте также возможность одному игроку делать разные виды ставок, но по одной каждого вида. В этом случае выигрыш игрока суммируется по всем ставкам.

**Задача 43.** В файле `data_austin_weather.csv` содержится информация о погоде на морском курорте за три года 2014, 2015, 2016, о каждом дне года в отдельной строке. Формат строки следующий: дата (год-месяц-день), температура по Фаренгейту, влажность в процентах, видимость в милях, скорость ветра в милях в час, количество осадков в дюймах, атмосферные явления (Rain — дождь, Thunderstorm — гроза, Fog — туман, Snow — снег).

Создайте класс или классы для хранения данных о погоде, например, отдельный класс для данных ежедневной погоды и отдельный класс для хранения данных за год. По заданной поль-

зователем дате и периоду времени (день, неделя, декада, месяц) выведите полную информацию о погоде в формате для русского пользователя (градусы Цельсия, видимость в километрах, ветер — в метрах в секунду, осадки — в мм.)

Дополнительно, по заданному пользователем виду атмосферного явления для каждого сезона (зима, весна, лето, осень), вычислите статистические характеристики температуры, влажности, видимости, скорости ветра: среднее значение, среднее отклонение от этого значения, диапазон изменений, моду, медиану. Описание характеристик и формулы их расчёта.

**Задача 44.** В файле `data_countries_world.csv` содержится информация о странах, о каждой стране в отдельной строке. Формат строки следующий: название, регион, численность населения, площадь в квадратных милях, береговая линия (отношение длины береговой линии к площади), ВВП на душу населения в долларах, грамотность в процентах, уровень рождаемости на 1000 человек, уровень смертности на 1000 человек, доля сельского хозяйства в ВВП, доля промышленности в ВВП, доля сферы обслуживания в ВВП.

Создайте класс для хранения данных о стране с методами для вычисления: плотности населения (человек на квадратный километр); длины береговой линии; абсолютного значения ВВП в долларах и в валюте, заданной пользователем (курс по отношению к доллару тоже указывает пользователь); примерного количества неграмотных людей в тысячах, сферу деятельности, дающую наибольший вклад в ВВП. В полях класса храните только те данные, которые нужны для работы указанных методов.

Выведите в отдельный файл для каждого региона список стран, упорядоченный по площади, с названиями и результатами работы, перечисленных выше методов. Если страна не граничит с морем или океаном, то длину береговой линии не выводите.

Дополнительно создайте класс или классы для хранения информации о частях света: европе, азии, африке, америке, австралии. Каждая часть света должна содержать список регионов, регион — список стран с полной информацией о них. Выведите на консоль информацию о стране, заданной пользователем. Выведите в отдельный файл для каждой части света список стран, упорядоченный по параметру, который указывает пользователь, в том числе, который может быть рассчитан с помощью описанных в задаче методов.

**Задача 45.** В файле `data_coursea.txt` содержится информация об учебных курсах: идентификационный номер, название, организация, которая проводит курс, тип сертификата, рейтинг, уровень курса, количество студентов, зачисленных на курс ( $1k = 1000$ ). Каждый из семи параметров расположен в отдельной строке, в следующих семи строках — информация о следующем объекте.

Создайте класс для хранения данных о курсе. Подберите подходящие курсы по параметрам, задаваемым пользователем: уровню, типу сертификата и ключевым словам. Ключевые слова вводятся пользователем через пробел и ищутся в названии курса без учёта регистра. Список подходящих кусков выведите на консоль, отсортированный по убыванию рейтинга и количества учащихся.

Дополнительно создайте класс для хранения информации об организации, проводящей курсы. Организация содержит список, проводимых курсов. Выведите в текстовый файл в формате csv информацию об организации: название, количество курсов по уровням, средний рейтинг курсов каждого уровня, общее количество учащихся.

**Задача 46.** В файле `data_olympic_summer.csv` содержится информация о чемпионах летних олимпийских игр, о каждом победителе в отдельной строке. Формат строки следующий: год, город проведения игр, вид спорта, имя победителя (фамилия, имя — разделены запятой и пробелом), код страны, пол, соревнование, полученная медаль.

Создайте класс для хранения информации о победителях отдельной страны. Данные должны содержать код страны, список победителей. По данным файла создайте список стран. Выведите в текстовый файл для каждой страны список имён золотых медалистов с указанием года и города проведения олимпиады, соревнование, такой же список серебряных и золотых медалистов, количество мужчин-чемпионов и женщин-чемпионов.

Дополнительно выведите в файл список видов спорта с указанием страны с наибольшим количеством чемпионов в этом виде спорта. Для заданной пользователем страны выведите олимпийские игры, в которых эта страна не участвовала. Выведите на консоль списки стран, упорядоченные по следующим параметрам

1. по убыванию количества завоёванных медалей в заданном году, тип медалей указывает пользователь (один из трёх или все вместе);

2. по убыванию доли женщин-чемпионов за всё время проведения игр;
3. по возрастанию количества многократных чемпионов.

**Задача 47.** В файле `data_person_year.csv` содержится информация о человеке или персоне года (до 2016 включительно), о каждом в отдельной строке. Формат строки следующий: год, номинация, имя, страна, год рождения, год смерти, звание, категория, комментарий. Некоторые данные могут отсутствовать.

Создайте класс для хранения данных только о людях, объявленных человеком года. Выведите на консоль и в текстовый файл список людей года для каждой страны отдельно с именем и званием. По выбору пользователем года или человека выведите полную информацию о человеке года (если в соответствующем году был выбран именно человек). Выведите на консоль список людей года, отсортированный по убыванию возраста на момент получения титула человека года.

Дополнительно создайте классы для хранения всей информации, возможно иерархию наследования. Создайте класс для хранения информации о людях или персонах года для категории. В классе храните название категории и списки людей и персон. По данным из файла создайте список категорий. Выведите на консоль список категорий, упорядоченный по убыванию общего количества людей и персон года, с указанием стран, к которым они относятся. Для каждой категории выведите в отдельный файл общий список людей и персон года, упорядоченный по году присвоения.

## Оценка 5

Для получения оценки 5 задача должна быть решена полностью.

**Задача 48.** В файлах `data_olimpic_summer.csv` и `data_olimpic_winter.csv` содержится информация о победителях летних и зимних олимпийских игр, о каждом победителе в отдельной строке. Формат строки следующий: год, город проведения игр, вид спорта, имя победителя (фамилия, имя — разделены запятой и пробелом), код страны, пол, соревнование, полученная медаль.

Создайте классы для хранения информации о победителях олимпийских игр для решения следующих задач:

1. по заданному году выведите город проведения олимпиады, вид (зимняя или летняя), список и количество стран-участниц, список золотых медалистов, серебряных, бронзовых;
2. по заданному виду спорта выведите года и города проведения олимпиад, включавших этот вид спорта, общий список соревнований в этом виде спорта, отдельно для каждого года список соревнований;
3. списки двукратных, трёхкратных и т. д. победителей с указанием имени, соревнования, года олимпиады и завоёванных медалей;
4. по заданному соревнованию укажите вид спорта, к которому он относится, и определите диапазон лет, когда он присутствовал на олимпийских играх.

Результаты решения задач выведите в текстовые файлы.

**Задача 49.** Информация о 12 знаках содержит диапазоны дат для каждого знака, краткое описание, и таблицу совместимости, которую можно найти по ссылке <https://horoscopes.rambler.ru/sovместimost-znakov-zodiaka/>. Для работы с датами надо использовать объекты класса `LocalDate` (пакет `java.time`).

Пользователь вводит дату рождения. Программа должна сообщать информацию о соответствующем знаке зодиака, затем о трёх наиболее подходящих знаках. Далее по запросу пользователя выдаётся информация об интересующем его знаке зодиака с датами и описанием. Затем выдаётся информация о следующих трёх подходящих знаках и о трёх самых неподходящих. Для тех дат, которые попадают на границу двух знаков (три дня до и три дня после границы), выдаётся об этом сообщение, после чего пользователь должен выбрать один из этих знаков и информация предоставляется о нём. Пользователь может отказаться от выбора предпочтительного знака, тогда информация выдаётся о каждом знаке отдельно.

**Задача 50.** Напишите программу, моделирующую работу с расписанием занятий студентов. Входной файл содержит список элементов расписания. Сведения о каждом элементе расписания

содержат: день недели, номер пары, название предмета, номер аудитории (от 1 до 20), название группы студентов. Список групп фиксированный и может быть задан в программе или отдельном файле.

Программа должна читать данные из входного файла и формировать расписание по правилам. В один день у группы не должно быть больше пяти пар, в один и тот же день недели на одной и той же паре в одной аудитории не должно оказаться двух разных занятий, в один и тот же день недели на одной и той же паре для одной и той же группы не должно оказаться двух разных занятий. Если очередной элемент расписания нарушает одно из условий, необходимо сообщить об этом и предложить вариант исправления расписания: переместить пару (название предмета и номер группы не изменяется) на подходящий день недели в свободную аудиторию и подходящее время. Выведите общее для всех групп расписание на каждый день недели, а также расписание для каждой группы.

**Задача 51.** Первый входной файл содержит информацию о плане расхода денежных средств. Формат входного файла: файл содержит пять строк следующего вида: название статьи расходов (продукты, квартплата, транспорт, медицина, развлечения) и запланированная сумма денег, разделенные произвольным количеством пробелов.

Второй входной файл содержит информацию о реальных расходах. Формат входного файла: файл содержит произвольное количество строк следующего вида: название статьи расходов, название продукта или услуги и потраченная сумма денег, данные разделены произвольным количеством пробелов.

Программа должна читать данные из входных файлов и выводить на консоль информацию о расходах в виде таблицы со столбцами: название статьи расходов, запланированная сумма денег, реально потраченная сумма денег, процент выполнения плана. В отдельных строках надо вывести подробную информацию о тех статьях, где план расходов был превышен, информация включает название статьи и набор продуктов и услуг, на которые были потрачены деньги в рамках данной статьи расходов, упорядоченный по убыванию потраченной суммы денег.

**Задача 52.** Входной файл содержит информацию о мероприятиях: название, дата и время проведения, длительность в часах, возможно не целое количество часов. Для хранения даты и времени используйте класс `LocalDateTime`.

Программа должна читать данные из входного файла и формировать календарь событий, упорядоченных по дате и времени. Для каждого события необходимо запросить у пользователя определение типа события: рабочее или личное. Рабочие события должны начинаться и заканчиваться в рабочее время: пять первых дней недели с 9 часов до 18. Аналогично личные события должны укладываться в остальное время, но в дни с воскресенья до четверга не позднее 23 часов. Если время события не подходит к этим условиям, оно не включается в календарь с сообщением об этом. Если очередное событие пересекается по времени с уже занесёнными, необходимо предложить пользователю либо отказаться от участия в нём, либо заменить им другие события, которые с ним пересекаются.

После формирования календаря список событий выводится на консоль. По запросу пользователя на консоль выводятся все события заданной даты.

**Задача 53.** Создайте программу для проведения и записи шахматной партии между двумя людьми. Игроки делают ходы по очереди. Доска и фигуры схематично отображаются на консоли. Запись партии осуществляется по правилам шахматной нотации в текстовый файл. При перезапуске программы восстанавливается прерванная партия или иницируется новая.

**Задача 54.** Создайте программу для проведения и записи игры в русские шашки между двумя людьми. Игроки делают ходы по очереди. Доска и фигуры схематично отображаются на консоли. Запись партии осуществляется по правилам записи ходов в текстовый файл. При перезапуске программы восстанавливается прерванная партия или иницируется новая.

**Задача 55.** Создайте классы для хранения и обработки информации о точках и плоскостях в трёхмерном пространстве. Информацию о плоскости храните в виде коэффициентов уравнения  $Ax + By + Cz + D = 0$ . Один из конструкторов должен создавать объект по трём точкам, если они не лежат на одной прямой. Создайте методы определяющие, лежит ли заданная точка в плоскости, расстояние от точки до плоскости, параллельность или перпендикулярность плоскостей, угол между плоскостями. Создайте метод, строящий плоскость параллельную данной и проходящую через заданную точку.

В главном классе создайте две плоскости, параметры прочитайте из файла или с консоли. Вы-

ведите на консоль информацию о каждой плоскости. Определите, являются ли плоскости параллельными или перпендикулярными, если нет, определите угол между ними. Результат выведите на консоль. Введите координаты точки и определите, принадлежит ли точка какой-нибудь плоскости. Если точка не принадлежит, то найдите расстояние от точки до плоскости и выведите уравнение плоскости параллельной заданной и проходящей через эту точку.

**Задача 56.** Создайте класс для работы с неориентированным графом. Про графы можно найти информацию <https://kvodo.ru/data-structures/graph-data-structure>. Определите поля и методы класса для решения описанных ниже задач.

Неориентированный граф задан матрицей смежности: если в графе существует ребро, соединяющее  $i$ -ю и  $j$ -ю вершины, то элементы  $i$ -й строки  $j$ -го столбца и  $j$ -й строки  $i$ -го столбца матрицы содержат положительное число — вес ребра, в противном случае — «ноль».

Программа должна читать описание графа из входного файла и выводить на консоль количество рёбер графа, все рёбра наибольшего и наименьшего веса, а также вектор, содержащий номера компонент связности, которым принадлежат соответствующие вершины графа (алгоритм можно посмотреть в статье [https://e-maxx.ru/algo/connected\\_components](https://e-maxx.ru/algo/connected_components)).

Формат входного файла: в первой строке задаётся количество вершин в графе, в последующих — элементы матрицы смежности (одна строка матрицы соответствует одной строке файла), разделённые произвольным количеством пробелов.

**Задача 57.** Создайте класс для работы с ориентированным графом. Про графы можно найти информацию <https://kvodo.ru/data-structures/graph-data-structure>. Определите поля и методы класса для решения описанных ниже задач.

Ориентированный граф задан матрицей смежности: если в графе существует дуга, исходящая из  $i$ -й и входящая в  $j$ -ю вершину, то элемент  $i$ -й строки  $j$ -го столбца матрицы содержит положительное число — длину дуги, в противном случае — «ноль».

Программа должна читать описание графа из входного файла и выводить на консоль количество дуг графа, количество дуг, входящих в каждую вершину графа, кратчайшие пути от каждой вершины до всех остальных достижимых из неё (алгоритм можно посмотреть в статье <https://kvodo.ru/dijkstra-algorithm.html>).

Формат входного файла: в первой строке задаётся количество вершин в графе, в последующих — элементы матрицы смежности (одна строка матрицы соответствует одной строке файла), разделённые произвольным количеством пробелов.

**Задача 58.** Создайте класс для работы с неориентированным графом. Про графы можно найти информацию <https://kvodo.ru/data-structures/graph-data-structure>. Определите поля и методы класса для решения описанных ниже задач.

Неориентированный граф задан набором специальных команд, его формирующих.

Программа должна читать описание графа из входного файла и выводить на консоль матрицу смежности графа.

Формат входного файла: в первой строке задаётся количество вершин в графе, в последующих — команды. Символы команд и аргументы отделяются друг от друга произвольным количеством пробелов, табуляций и символов перевода строки. Если команда не может быть выполнена или не является допустимой, она игнорируется с записью соответствующего сообщения в выходной файл. Допустимые команды:

A <номер вершины> <номер вершины> — добавить ребро, соединяющее вершины;  
R <номер вершины> <номер вершины> — удалить ребро, соединяющее вершины;  
F <номер вершины> — добавить рёбра, соединяющие заданную вершину со всеми остальными;  
I <номер вершины> — удалить все рёбра, инцидентные данной вершине;  
C <номер вершины> <номер вершины> — поменять местами номера двух вершин;