|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| ***Заданий для практических занятий.*** | | | | |
| **Задание на практическую работу №1**  1. Создать проект в IntelliJ IDEA 2. Cоздать собственный Git репозитарий 3. Написать программу, в результате которой считается сумма элементов целочисленного массива с помощью циклов for, while, do while, результат выводится на экран. 4. Написать программу, в результате которой выводятся на экран аргументы командной строки в цикле for. 5. Написать программу, в результате работы которой выводятся на экран первые 10 чисел гармонического ряда (форматировать вывод). 6. Написать программу, в результате которой генерируется массив целых чисел случайным образом, вывести его на экран, отсортировать его, и снова вывести на экран (использовать два подхода к генерации случайных чисел – метод random() класса Math и класс Random). 7. Написать программу, которая с помощью метода, вычисляет факториал числа (использовать управляющую конструкцию цикла), проверить работу метода. 8. Результаты выполнения практической работы залить через IDE в свой репозитарий и продемонстрировать преподавателю.  **Задание на практическую работу №2**  1.Необходимо реализовать простейший класс Shape (Фигура) на языке программирования Java. Добавить метод toString(). Создать класс-тестер для вывода информации об объекте. 2. Реализуйте простейший класс «Мяч» 3. Реализуйте простейший класс «Книга»  4. Разработайте и реализуйте класс Dog (Собака), поля класса описывают кличку и возраст собаки. Необходимо выполнить следующие действия: определить конструктор собаки, чтобы принять и инициализировать данные экземпляра., включить стандартные методы (аксессоры) для получения и установки для имени и возраста, включить метод для перевода возраста собаки в "человеческий " возраст (возраст семь раз собаки), включите метод ToString, который возвращает описание экземпляра собаки в виде строки. Создание класса тестера под названием ПитомникСобак, реализует массив собак и основной метод этого класса позволяет добавить в него несколько объектов собаки.  **Задание на практическую работу №3**  1.Создать класс, описывающий модель окружности (Circle). В классе должны быть описаны нужные свойства окружности и методы для получения, изменения этих свойств. Протестировать работу класса в классе CircleTest, содержащим метод статический main(String[] args). 2.Создать класс, описывающий тело человека(Human). Для описания каждой части тела создать отдельные классы(Head, Leg, Hand). Описать необходимые свойства и методы для каждого класса. Протестировать работу класса Human. 3.Создать класс, описывающий книгу (Book). В классе должны быть описаны нужные свойства книги(автор, название, год написания и т. д.)и методы для получения, изменения этих свойств. Протестировать работу класса в классе BookTest, содержащим метод статический main(String[] args).  **Задание на практическую работу №4**  1.Прочитать UML диаграмму: на диаграмме представлены Абстрактный суперкласс Shape и его подклассы Circle, Rectangle and Square. 2.Перепишите суперкласс Shape и его подклассы так как это представлено на UML диаграмме Circle, Rectangle and Square  **Задание на практическую работу №5**  Вам нужно написать два класса MovablePoint и MovableCircle - которые реализуют интерфейс Movable на основе классов, разработанных в практической работе № 4. Изучите UML диаграмму и представьте реализацию класса  **Задание на практическую работу №6**  1.Создайте в draw.io UML диаграмму, а затем напишите по ней реализацию.  2.Добавьте в проект ранее разработанный класс прямоугольник (Rectangle). На основе него напишите новый класс MovableRectangle (движущийся прямоугольник). Ваш класс должен реализовывать интерфейс Movable  3.Прямоугольник можно также представить как две движущиеся точки MovablePoints (представляющих верхняя левая и нижняя правая точки) и реализующие интерфейс Movable.  4.Убедитесь, что две точки имеет одну и ту же скорость (нужно добавить метод, который бы это проверял).  **Задание на практическую работу №7**  1. Создать абстрактный класс, описывающий посуду(Dish). С помощью наследования реализовать различные виды посуды, имеющие свои свойства и методы. Протестировать работу классов. 2. Создать абстрактный класс, описывающий собак(Dog). С помощью наследования реализовать различные породы собак. Протестировать работу классов. 3. Создать абстрактный класс, описывающий мебель. С помощью наследования реализовать различные виды мебели. Также создать класс FurnitureShop, моделирующий магазин мебели.  4.Протестировать работу всех классов.  **Задание на практическую работу №8**  1.Создать окно, нарисовать в нем 20 случайных фигур, случайного цвета. Классы фигур должны наследоваться от абстрактного класса Shape, в котором описаны свойства фигуры: цвет, позиция. 2. Создать окно, отобразить в нем картинку, путь к которой указан в аргументах командной строки. 3. Создать окно, реализовать анимацию, с помощью картинки, состоящей из нескольких кадров.  **Задание на практическую работу №9**  В задание из практической работы №8 добавить кнопку, в результате обработки нажатия события нажатия на которую, генерируются случайным образом различные фигуры из задания на эту практическую работу  **Задание на практическую работу №10**  1. Треугольная последовательность. Дана монотонная последовательность, в которой каждое натуральное число k встречается ровно k раз: 1, 2, 2, 3, 3, 3, 4, 4, 4, 4,… По данному натуральному n выведите первые n членов этой последовательности. Попробуйте обойтись только одним циклом for. 2. От 1 до n. Дано натуральное число n. Выведите все числа от 1 до n. 3. От A до B. Даны два целых числа A и В (каждое в отдельной строке). Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае. 4. Заданная сумма цифр. Даны натуральные числа k и s. Определите, сколько существует k-значных натуральных чисел, сумма цифр которых равна d. Запись натурального числа не может начинаться с цифры 0. В этой задаче можно использовать цикл для перебора всех цифр, стоящих на какой-либо позиции. 5. Сумма цифр числа. Дано натуральное число N. Вычислите сумму его цифр. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). 6. Проверка числа на простоту. Дано натуральное число n>1. Проверьте, является ли оно простым. Программа должна вывести слово YES, если число простое и NO, если число составное. Алгоритм должен иметь сложность O(logn). Указание: понятно, что задача сама по себе является нерекурсивной, т.к. проверка числа n на простоту никак не сводится к проверке на простоту меньших чисел. Поэтому нужно сделать еще один параметр рекурсии: делитель числа, и именно по этому параметру и делать рекурсию. 7. Разложение на множители. Дано натуральное число n>1. Выведите все простые множители этого числа в порядке неубывания с учетом кратности. Алгоритм должен иметь сложность O(logn) 8. Палиндром. Дано слово, состоящее только из строчных латинских букв. Проверьте, является ли это слово палиндромом. Выведите YES или NO. При решении этой задачи нельзя пользоваться циклами, в решениях на питоне нельзя использовать срезы с шагом, отличным от 1. 9. Без двух нулей. Даны числа a и b. Определите, сколько существует последовательностей из a нулей и b единиц, в которых никакие два нуля не стоят рядом. 10. Разворот числа. Дано число n, десятичная запись которого не содержит нулей. Получите число, записанное теми же цифрами, но в противоположном порядке. При решении этой задачи нельзя использовать циклы, строки, списки, массивы, разрешается только рекурсия и целочисленная арифметика. Функция должна возвращать целое число, являющееся результатом работы программы, выводить число по одной цифре нельзя. 11. Количество единиц. Дана последовательность натуральных чисел (одно число в строке), завершающаяся двумя числами 0 подряд. Определите, сколько раз в этой последовательности встречается число 1. Числа, идущие после двух нулей, необходимо игнорировать. В этой задаче нельзя использовать глобальные переменные и параметры, передаваемые в функцию. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметров. 12. Вывести нечетные числа последовательности. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите все нечетные числа из этой последовательности, сохраняя их порядок. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции. 13. Вывести члены последовательности с нечетными номерами. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Выведите первое, третье, пятое и т.д. из введенных чисел. Завершающий ноль выводить не надо. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция не возвращает значение, а сразу же выводит результат на экран. Основная программа должна состоять только из вызова этой функции. 14. Цифры числа слева направо. Дано натуральное число N. Выведите все его цифры по одной, в обычном порядке, разделяя их пробелами или новыми строками. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика 15. Цифры числа справа налево. Дано натуральное число N. Выведите все его цифры по одной, в обратном порядке, разделяя их пробелами или новыми строками. При решении этой задачи нельзя использовать строки, списки, массивы (ну и циклы, разумеется). Разрешена только рекурсия и целочисленная арифметика. 16. Количество элементов, равных максимуму. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите, какое количество элементов этой последовательности, равны ее наибольшему элементу. В этой задаче нельзя использовать глобальные переменные. Функция получает данные, считывая их с клавиатуры, а не получая их в виде параметра. В программе на языке Python функция возвращает результат в виде кортежа из нескольких чисел, и функция вообще не получает никаких параметров. В программе на языке Java результат записывается в переменные, которые передаются в функцию по ссылке. Других параметров, кроме как используемых для возврата значения, функция не получает. Гарантируется, что последовательность содержит хотя бы одно число (кроме нуля) 17. Максимум последовательности. Дана последовательность натуральных чисел (одно число в строке), завершающаяся числом 0. Определите наибольшее значение числа в этой последовательности. В этой задаче нельзя использовать глобальные переменные и передавать какие-либо параметры в рекурсивную функцию. Функция получает данные, считывая их с клавиатуры. Функция возвращает единственное значение: максимум считанной последовательности. Гарантируется, что последовательность содержит хотя-бы одно число (кроме нуля).  **Задание на практическую работу №11**  1.Реализуйте игру-угадайку, которая имеет одно текстовое поле и одну кнопку. Программа предложит пользователю угадать число между 0-20 и дает ему три попытки. Если ему не удастся угадать, то будет выведено сообщение, что пользователь допустил ошибку в угадывании и что число меньше/больше. Если пользователь попытался три раза угадать, то программа завершается с соответствующим сообщением. Если пользователь угадал, то программа должна тоже завершаться с соответствующим сообщением об этом. 2. Реализация приложения Java, который имеет макет границы и надписи для каждой области в макете. Теперь определим события мыши, чтобы описать действия: a. Когда мышь входит CENTER программа показывает диалоговое окно (Добро пожаловать в ЦАО) b. Когда мышь входит WEST программа показывает диалоговое окно (Добро пожаловать в ЗАО) c. Когда мышь входит SOUTH программа показывает диалоговое окно (Добро пожаловать ЮЗАО) d. Когда мышь входит в NORTH программа показывает диалоговое окно (Добро пожаловать в) e. Когда мышь входит EAST программа показывает диалоговое окно (Добро пожаловать в СВАО) 3. Реализация программу на Java с JTextArea и двумя меню: Цвет: должна быть возможность выбора одного из трех возможных цветов: синий, красный и черный Шрифт: должна быть возможность выбора одного из трех: “Times New Roman”, “MS Sans Serif”, “Courier New”.  Вы должны написать программу, которая с помощью меню, может изменять шрифт и цвет текста, написанного в JTextArea  **Задание на практическую работу №12**  1. Напишите программу, в которой определен перечисляемый тип Planet (планета). Данный тип должен описывать планеты Солнечной системы. Для каждого элемента перечисления с помощью конструктора задайте массу и радиус.  2. Добавьте в программу меню, в результате работы которого пользователь сможет узнать силу притяжения для выбранной им планеты.  3. Разработайте GUI программу для этого задания с обработкой событий  **Задание на практическую работу №13**  1.Написать тестовый класс, который создает массив класса Student и сортирует массив iDNumber. 2. Напишите класс SortingStudentsByGPA (по среднему баллу студентов) который реализует интерфейс Comparator таким образом, что она сортирует студентов с их итоговым баллом в порядке убывания. В качестве алгоритма сортировки использовать методы сортировок: слиянием и быструю сортировку. 3. Напишите программу, которая объединяет два списка данных о студентах в один отсортированный списках.  **Задание на практическую работу №14**  1.Написать реализацию алгоритмов поиска для задания из предыдущей практической работы, в качестве параметров методы сортировки и поиска должны принимать массив интерфейсных ссылок.  2.Реализовать в классе типа сортируемого объекта метод equals().  3.Реализовать следующие алгоритмы линейного и бинарного поиска: в виде итеративного и рекурсивного поиска.  4.Сравнить время работы и сложность алгоритмов.  Замечание: для реализации бинарного поиска упорядочить массив.  **Задание на практическую работу №15**  1. Протестировать работу коллекции ArrayList. 2. Протестировать работу коллекции LinkedList. 3. Создать свою коллекцию, такую же, как и ArrayList из объектов классов разработанных в предыдущих программах.  4. продемонстрировать работу с собственной коллекцией  **Задание на практическую работу №16**  1 Карточная игра “в пьяницу”. Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9. Условия игры. В игре “в пьяницу” карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт - проигрывает. Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту (“шестерка берет туза”). Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды). Входные данные Программа получает на вход две строки: первая строка содержит 5 карт первого игрока, вторая - 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой. Выходные данные Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 106 ходов игра не заканчивается, программа должна вывести слово botva. Пример ввода  1 3 5 7 9 2 4 6 8 0 Вывод  second 5  **Задание на практическую работу №17**  1. Реализовать запись в файл введённой с клавиатуры информации 2. Реализовать вывод информации из файла на экран 3. Заменить информацию в файле на информацию, введённую с клавиатуры 4. Добавить в конец исходного файла текст,  **Задание на практическую работу №18**  1.Необходимо инстанцировать класс и выполнить exceptionDemo (). Программа даст сбой, и вы получите следующее сообщение java.lang.ArithmeticException: / by zero at Exception1.exceptionDemo(Exception1.java:12)  Это говорит нам о том, что программа пытается выполнить деление на ноль, который он не в состоянии выполнить. Выполните следующее:  1.1 Замените 2/0 на 2,0 / 0,0 и повторно вызовите метод. Объясните, что произойдет  1.2 Измените код в листинге 1 на следующий: public class Exception2 {   public void exceptionDemo() {   Scanner myScanner = new Scanner( System.in );   System.out.print( "Enter an integer ");  String intString = myScanner.next();  int i = Integer.parseInt(intString); System.out.println( 2 / i );   }  } 2. С помощью перехватывания исключений можно оказывать влияние на поведение программы. В вашем решении в предыдущем упражнении вы можете добавить новый пункт - catch в начале списка пунктов catch. Выполите это, чтобы поймать общее исключение класса Exception.  3.Перезапустите программу с приведенными выше данными и обратите внимание на ее поведение. Объясните новое поведение программы 4.И наконец добавьте блок finally к решению. Повторно запустите программу, чтобы проверить ее поведение. Объясните новое поведение программы  **Задание на практическую работу №19**  1. Написать программную реализацию для онлайн покупок в зарубежном интернет-магазине. Клиент совершает онлайн покупку. При оформлении заказа у пользователя запрашивается фио и номер ИНН. В программе проверяется, действителен ли номер ИНН для такого клиента. Исключение будет выдано в том случае, если введен недействительный ИНН. 2. Предлагается модернизировать задачу из практической работы №13 – задача сортировки студентов по среднему баллу. Необходимо разработать пользовательский интерфейс для задачи поиска и сортировки (использовать массив интерфейсных ссылок- пример в лекции 5. Дополнить ее поиском студента по фио – в случае отсутствия такого студента необходимо выдавать собственное исключение.  **Задание на практическую работу №20**  1. Написать метод для конвертации массива строк/чисел в список.  2. Написать класс, который умеет хранить в себе массив любых типов данных (int, long etc.). 3. Реализовать метод, который возвращает любой элемент массива по индексу. 4. Написать функцию, которая сохранит содержимое каталога в список и выведет первые 5 элементов на экран. 5. \*Реализуйте вспомогательные методы в классе Solution, которые должны создавать соответствующую коллекцию и помещать туда переданные объекты. Методы newArrayList, newHashSet параметризируйте общим типом T. Метод newHashMap параметризируйте парой <K, V>, то есть типами К- ключ и V-значение. Аргументы метода newHashMap должны принимать  **Задание на практическую работу №21**  1. Реализовать класс Абстрактная фабрика для комплексных чисел 2. Реализовать класс Абстрактная фабрика для различных типов стульев: Викторианский стул, Многофункциональный стул, Магический стул, а также интерфейс Стул, от которого наследуются все классы стульев, и класс Клиент, который использует интерфейс стул в своем методе Sit (Chair chair). 3. Вводная. Компании нужно написать редактор текста, редактор изображений и редактор музыки. В трёх приложениях будет очень много общего: главное окно, панель инструментов, команды меню будут весьма схожими. Чтобы не писать повторяющуюся основу трижды, вам поручили разработать основу (каркас) приложения, которую можно использовать во всех трёх редакторах. На совещании в компании была принята следующая архитектура: Исходные данные. Есть некий базовый интерфейс IDocument, представляющий документ неопределённого рода. От него впоследствии будут унаследованы конкретные документы: TextDocument, ImageDocument, MusicDocument и т.п. Интерфейс IDocument перечисляет общие свойства и операции для всех документов. • При нажатии пунктов меню File -> New и File -> Open требуется создать новый экземпляр документа (конкретного подкласса). Однако каркас не должен быть привязан ни к какому конкретному виду документов. • Нужно создать фабричный интерфейс ICreateDocument. Этот интерфейс содержит два абстрактных фабричных метода: CreateNew и CreateOpen, оба возвращают экземпляр IDocument • Каркас оперирует одним экземпляром IDocument и одним экземпляром ICreateDocument. Какие конкретные классы будут подставлены сюда, определяется во время запуска приложения. Требуется:  1. создать перечисленные классы. Создать каркас приложения — окно редактора с меню File. В меню File реализовать пункты New, Open, Save, Exit. 2. продемонстрировать работу каркаса на примере текстового редактора. Потребуется создать конкретный унаследованный класс TextDocument и фабрику для него — CreateTextDocument. В данной практической работе рекомендуется использовать следующие паттерны: Абстрактная фабрика и фабричный метод.  **Задание на практическую работу №22**  1.Доработайте класс Student предусмотрите поле для хранения даты рождения, перепишите метод toString() таким образом, чтобы он разработайте метод, возвращал строковое представление даты рождения по вводимому в метод формату даты (например, короткий, средний и полный формат даты). 2. Напишите пользовательский код, который формирует объекты Date и Calendar по следующим данным, вводимым пользователем: <Год><Месяц><Число> <Часы1><минуты> 3Написать программу, выводящую фамилию разработчика, дату и время получения задания, а также дату и время сдачи задания. Для получения последней даты и времени использовать класс Date из пакета java.util.\* (Объявление Dated=newDate() или метод System.currentTimeMillis().  **Задание на практическую работу №23**  1. Создать словарь (Map<String, String>), записать в него десять записей по принципу «фамилия» - «имя». Удалить людей, имеющих одинаковые имена. Вывести словарь на печать 2. Описать класс City, с полями: город, страна. Записать исходные данные в HashMap, где ключ - страна, значение - город. Map<String,ArrayList<String>> map = new HashMap(); Записать необходимо таким образом, чтобы каждому ключу(стране) соответствовал список значений(городов).  **Задание на практическую работу №24**  1. Разработать класс Person, в котором имеется функция, возвращающая Фамилию И.О. Функция должна учитывать возможность отсутствия значений в полях Имя и Отчество. Программу оптимизировать с точки зрения быстродействия. 2. Доработать класс адреса, который из полученной строки формата "Страна[d] Регион[d] Город[d] Улица[d] Дом[d] Корпус[d] Квартира" ([d] – разделитель, например, «запятая») выбирает соответствующие части и записывает их в соответствующие поля класса Address. Учесть, что в начале и конце разобранной части адреса не должно быть пробелов. Все поля адреса строковые. Разработать проверочный класс не менее чем на четыре адресных строки. В программе предусмотреть две реализации этого метода: а) разделитель – только запятая (использовать метод split()); Внимание, при разработке нужно учесть, что  б) разделитель – любой из символов ,.;- (класс StringTokenizer). 3. Реализуйте класс Shirt: Метод toString() выводит объяснение и значение полей построчно. Дан также строковый массив: shirts[0] = "S001,Black Polo Shirt,Black,XL"; shirts[1] = "S002,Black Polo Shirt,Black,L"; shirts[2] = "S003,Blue Polo Shirt,Blue,XL"; shirts[3] = "S004,Blue Polo Shirt,Blue,M"; shirts[4] = "S005,Tan Polo Shirt,Tan,XL"; shirts[5] = "S006,Black T-Shirt,Black,XL"; shirts[6] = "S007,White T-Shirt,White,XL"; shirts[7] = "S008,White T-Shirt,White,L"; shirts[8] = "S009,Green T-Shirt,Green,S"; shirts[9] = "S010,Orange T-Shirt,Orange,S"; shirts[10] = "S011,Maroon Polo Shirt,Maroon,S"; Преобразуйте строковый массив в массив класса Shirt и выведите его на консоль. 4. Разработайте класс, который получает строковое представление телефонного номера в одном из двух возможных строковых форматов: +<Код страны><Номер 10 цифр>, например “+79175655655” или “+104289652211” или 8<Номер 10 цифр> для России, например “89175655655” и преобразует полученную строку в формат: +<Код страны><Три цифры>–<Три цифры>–<Четыре цифры> 5. В методе main считай с консоли имя файла, который содержит слова, разделенные пробелом. В методе getLine() используя StringBuilder расставьте все слова в таком порядке, чтобы последняя буква данного слова совпадала с первой буквой следующего не учитывая регистр. Каждое слово должно участвовать 1 раз.  **Задание на практическую работу №25**  1.Ознакомиться с классами Pattern, Matcher и PatternSyntaxException  2. Выбрать IPv4 адреса во всех возможных, представлениях: десятичном, шестнадцатеричном и восьмеричном. С точками и без. Подробнее про IP адреса можно узнать в википедии  3. Составить регулярное выражение, определяющее является ли заданная строка IP адресом, записанным в десятичном виде. – пример правильных выражений: 127.0.0.1, 255.255.255.0. – пример неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.  4. Создать запрос для вывода только правильно написанных выражений со  скобками (количество открытых и закрытых скобок должно быть одинаково). – пример правильных выражений: (3\*+\*5)\*–\*9\*×\*4. – пример неправильных выражений: ((3\*+\*5)\*–\*9\*×\*4.  **Задание на практическую работу №26**  1. Реализовать очередь на массиве • Найдите инвариант структуры данных «очередь». Определите функции, которые необходимы для реализации очереди. Найдите их пред- и постусловия. • Реализуйте классы, представляющие циклическую очередь с применением массива. В результате выполнения задания должны быть реализованы следующие классы: класс ArrayQueueModule должен реализовывать один экземпляр очереди с использованием переменных класса, класс ArrayQueueADT должен реализовывать очередь в виде абстрактного типа данных (с явной передачей ссылки на экземпляр очереди), класс ArrayQueue должен реализовывать очередь в виде класса (с неявной передачей ссылки на экземпляр очереди)  В задании у классов должны быть реализованы следующие функции(процедуры) / методы:  enqueue – добавить элемент в очередь;  element – первый элемент в очереди;  dequeue – удалить и вернуть первый элемент в очереди;  size – текущий размер очереди;  isEmpty – является ли очередь пустой;  clear – удалить все элементы из очереди. Замечания: инвариант, пред- и постусловия записываются в исходном коде в виде комментариев. Также обратите внимание на инкапсуляцию данных и кода во всех трех реализациях. Напишите тесты реализованным классам.  2. Реализация структуры данных Очередь на связном списке  Для выполнения задания необходимо: -Определить интерфейс очереди Queue и описать его контракт. -Реализовать класс LinkedQueue — очередь на связном списке. -Выделить общие части классов LinkedQueue и ArrayQueue в базовый класс AbstractQueue. Напишите тесты реализованным классам. Дополнительные задания 3. Вычисление выражений 3.1. Разработайте классы Const, Variable, Add, Subtract, Multiply, Divide для вычисления выражений с одной переменной. 3.2 Классы должны позволять составлять выражения вида  new Subtract(  new Multiply(  new Const(2),  new Variable("x")  ),  new Const(3)  ).evaluate(5) При вычислении такого выражения вместо каждой переменной подставляется значение, переданное в качестве параметра методу evaluate (на данном этапе имена переменных игнорируются). Таким образом, результатом вычисления приведенного примера должно стать число 7. 3.3. Для тестирования программы должен быть создан класс Main, который вычисляет значение выражения x2−2x+1, для x, заданного в командной строке. 3.4. При выполнение задания следует обратить внимание на: • Выделение общего интерфейса создаваемых классов. • Выделение абстрактного базового класса для бинарных операций. Задание 4  4.1. Доработайте предыдущее задание, так что бы выражение строилось по записи вида   x \* (y - 2)\*z + 1 Для этого реализуйте класс ExpressionParser с методом TripleExpression parse(String). 4.2. В записи выражения могут встречаться: умножение \*, деление /, сложение +, вычитание -, унарный минус -, целочисленные константы (в десятичной системе счисления, которые помещаются в 32-битный знаковый целочисленный тип), круглые скобки, переменные (x, y и z) и произвольное число пробельных символов в любом месте (но не внутри констант). 4.3. Приоритет операторов, начиная с наивысшего • унарный минус; • умножение и деление; • сложение и вычитание. Пример для выражения 1000000\*x\*x\*x\*x\*x/(x-1) вывод программы должен иметь следующий вид:   x f  0 0  1 division by zero  2 32000000  3 121500000  4 341333333  5 overflow  **Задание на практическую работу №27**  1.Реализовать в виде паттерна MVC  Написать калькулятор для чисел с использованием RPN (Reverse Polish Notation в пер. на русск. яз. - обратной польской записи)  2. Напишите программу-калькулятор арифметических выражений, записанных в обратной польской нотации (RPN-калькулятор).  Постройте систему тестов и проверьте, что ваш калькулятор успешно проходит все тесты и «защищён от дурака» (как дурака-пользователя программы, так и дурака-программиста, использующего ваш стек, и калькулятор). Например, если вводится выражение, в котором число операций превосходит число помещенных в стек элементов (например, 1 2 + \*), то программа не допустит уменьшения переменной sp до отрицательных значений, а выдаст предупреждение «Невозможно выполнить POP для пустого стека». Работу можно выполнить в двух вариантах (1вариант проще, второй-труднее)  3. Реализовать игру "змейка" в виде GUI c использованием подходящих паттернов и модели MVC. Программа должна быть декомпозирована на соответствующие классы: BodyPos, GameController, GameModel GameView, SnakePanel, Main  **Задание на практическую работу №28**  Написать программу реализующую оплату в интернет-магазине клиентом покупок. Использование паттерна Стратегия должно реализовать выбор платёжного метода в интернет-магазине. Когда пользователь сформировал заказ, он получает выбор из нескольких платёжных средств: электронного кошелька или кредитной карты. В данном случае конкретные стратегии платёжных методов не только проводят саму оплату, но и собирают необходимые данные на форме заказа.  **Задание на практическую работу №29**  В процессе написания тестовых заданий ознакомьтесь с принципами создания динамических структур в Java, механизмом исключений и концепцией интерфейсов.  Замечание: в процессе выполнения задания НЕЛЬЗЯ пользоваться утилитными классами Java (за исключением java.util.HashMap).  1.Создайте класс Drink – напитка. Класс описывает сущность – напиток и характеризуется следующими свойствами - стоимостью, названием и описанием. Класс должен быть определен как неизменяемый (Immutable class). Создайте класс Dish , описывает напитки 2. Создайте интерфейс Item – для работы с позициями заказа. Интерфейс определяет 3 метода: возвращает стоимость, возвращает название., возвращает описание позиции. Классы Drink и Dish, которые должны реализовывать этот интерфейс. 3. Создайте класс InternetOrder, который моделирует сущность интернет заказ в ресторане или кафе. Класс основан на циклическом двусвязном списке с выделенной головой и может хранить как блюда, так и напитки. Внимание: список реализуется самостоятельно!  Конструкторы: не принимающий параметров (для списка создается только головной элемент, сам список пуст), принимающий массив позиций заказа (создаем список из N позиций). Класс InternetOrder моделирует сущность ""нтернет-заказ"" в ресторане или кафе. Класс основан на циклическом двусвязном списке с выделенной головой и может хранить как блюда, так и напитки. Внимание: список вы реализуется самостоятельно! Класс Dish сделайте неизменяемым (аналогично Drink). Класс Order должен хранить (удалять и добавлять) не только экземпляры класса Dish, но и Drink (для этого разработайте классы Order и TablesOrderManager). Методы: −добавляющий позицию в заказ (принимает ссылку типа Item). Пока этот метод возвращает истину после выполнения операции добавления элемента.  4. Переименуйте класс Order из предыдущего задания в RestaurantOrder. Создайте интерфейс Order – позиции заказа.  Интерфейс должен определять следующие методы:  −добавления позиции в заказ (принимает ссылку типа Item), при этом возвращает логическое значение.  −удаляет позицию из заказа по его названию (принимает название блюда или напитка в качестве параметра). Возвращает логическое значение.  −удаляет все позиции с заданным именем (принимает название в качестве параметра). Возвращает число удаленных элементов.  −возвращает общее число позиций заказа в заказе.  −возвращает массив позиций заказа.  −возвращает общую стоимость заказа.  −возвращает число заказанных блюд или напитков (принимает название в качестве параметра).  −возвращает массив названий заказанных блюд и напитков (без повторов).  −возвращает массив позиций заказа, отсортированный по убыванию цены.  Замечание: Классы InternetOrder и RestaurantOrder должны реализовывать интерфейс Order. 5. Переименуйте класс TablesOrderManager в OrderManager. Добавьте ему еще одно поле типа java.util.HasMap<String, Order>, которое содержит пары адрес-заказ, и методы (работающие с этим полем):  Методы класса: −перегрузка метода добавления заказа. В качестве параметров принимает строку – адрес и ссылку на заказ.  −перегрузка метода получения заказа. В качестве параметра принимает строку – адрес.  −перегрузка метода удаления заказа. В качестве параметра принимает строку – адрес заказа. −перегрузка метода добавления позиции к заказу. В качестве параметра принимает адрес и Item.  −возвращающий массив имеющихся на данный момент интернет-заказов.  −возвращающий суммарную сумму имеющихся на данный момент интернет-заказов.  −возвращающий общее среди всех интернет-заказов количество заказанных порций заданного блюда по его имени. Принимает имя блюда в качестве параметра. Методы должны работать с интерфейсными ссылками Order и Item.  6.Создайте пользовательское исключение OrderAlreadyAddedException, выбрасываемое при попытке добавить заказ столику или по адресу, если со столиком или адресатом уже связан такой заказ.  Конструктор классов Drink и Dish должен выбрасывать исключение java.lang.IllegalArgumentException при попытке создать блюдо или напиток со стоимостью меньше 0, без имени или описания (если параметры имя и описание - пустые строки).  Создайте не объявляемое исключение IllegalTableNumber, выбрасываемое в методах, принимающих номер столика в качестве параметра, если столика с таким номером не существует.  **Задание на практическую работу №30**  1. Разработка кода по UML диаграмме с использованием классов, разработанных в практической работе №30.  2. Разработка интерфейса пользователя для интерактивного взаимодействия.    **Задание на практическую работу №31**  1. Использовать Maven для подключения модуля JUnit  2.Задание на виды тестирования  2.1 Модульное тестирование: − В среде IDE IntelliJ IDEA открыть исходный код тестируемого приложения (собственное, или из предыдущей практической работы)  − Добавить Unit-тест для одной из функций. − Запустить тест и просмотреть результаты. − Создать несколько разных тестов для проверки значений и перехвата исключений.  2.2 Покрытие кода тестами: − Установить параметры сбора статистики покрытия кода. − Повторить модульные тесты и просмотреть данные о покрытии кода.  2.3 Нагрузочное тестирование: − Для тестируемого приложения (собственное или из предыдущей практической работы ) провести профайлинг (оценку производительности). − Выполнить тест и просмотреть результаты.  **Задание на практическую работу №32**  Задание на сериализацию. Для задания из практической работы №30 обеспечить сохранение объектов в файл и чтение из файла | | | | |