|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Требуемая компетенция** | **Трайб/команда** | **Описание задачи** | **Требования по реализации** | **Критерии приёмки** | **ФИО представителя от команды/трайба для ревью задачи** |
| Java (7-8 грейд) | ДИТ КИБ/ Цифровой корпоративный банк | Заполнение полей класса, получая значения из параметров метода main()  Дан класс Person:  public class Person {      private String firstName;      private String lastName;  }  Используя входящие параметры метода main(String[] args), заполните поля класса Person и выведите их значения в виде строки в System.out.  **Заполнение полей класса, реализовав интерактивный ввод данных, используя входной поток System.in**  Необходимо реализовать интерактивное заполнение полей класса Person используя коносль, утилитный класс java.util.Scanner и входной поток System.in. Выведите значения полей класса Person в виде строки в System.out.  public class Main1 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          ...      }  }  **Заполнение списка объектов Person, реализовав интерактивный ввод данных, используя входной поток System.in**  Необходимо реализовать интерактивное заполнение списка объектов Person в виде бесконечного цикла. Поля класса Person должны заполняться с использованием консоли, утилитного класса java.util.Scanner и входного потока System.in. Консольное приложение должно предусмотреть точку выхода из бесконечного цикла заполнения списка объектов Person. Выведите значения полей списка объектов Person виде строк в System.out.  **Ввод, сортировка, вывод списка объектов Person**  Заполнение списка объектов Person аналогичен условиям предыдущего задания. Выведите значения полей списка объектов Person виде строк в System.out, предварительно отсортировав список объектов Personпо полю lastName.  **Реализация интерактивного меню для работы со списком объектов Person**  Необходимо реализовать интерактивное меню с пунктами:   * **Add** (добавление объекта Person в список); * **Show** (вывод списка объектов Person в виде строк в System.out); * **Exit** (выход);   public class Main4 {      public static void main(String[] args) {          List<Person> personList = new ArrayList<>();          Scanner scanner = new Scanner(System.in);          ...          System.out.println("Menu:");          System.out.println("1.Add");          System.out.println("2.Show");          System.out.println("3.Exit");          ...          }      }  }  **Реализация интерактивного меню для работы со списком объектов Person**  Добавить в меню пункты:   * **Show sorted unique** (вывод отсортированного списка объектов Person по полую lastName, исключая все дублирующие объекты по этому полю из списка в виде строк в System.out);   **Реализация интерактивного меню для работы со списком объектов Person**  Добавить в меню пункт:   * **Save to file** (запись исходного списка объектов Person в файл);   **Реализация интерактивного меню для работы со списком объектов Person, используя принципы ООП**  Необходимо реализовать интерактивное меню, используя принципы ООП  Интерфейс для описания действия со списком объектов Person:  private interface Exec {      void exec(List<Person> data) throws Exception;  }  Класс описывающий пункт меню:  private static class MenuItem {      // Наименование пункта меню      private String name;      // Доступное действие      private Exec exec;  }  Класс описывающий объект меню:  private static class Menu {      private List<MenuItem> items;      ...  }  Построение меню:  public class Main7 {      public static void main(String[] args) {          List<Person> personList = new ArrayList<>();          Scanner scanner = new Scanner(System.in);          Menu menu = new Menu(scanner);          ...      }  }  **Реализация интерактивного меню для работы со списком объектов Person, используя принципы ООП**  Добавить в меню пункты:   * **Read from file** (чтение списка объектов Person из файла); * **Clear data in memory** (очистка ранее загруженного списка объектов Person); | Предоставить исходный код программы, в том числе Файлы настроек (при их наличии). | 1. В каком виде предоставлены исходники (как проект или просто набор классов)  2. Стиль написания кода (разнесение по классам и т.д)  3. Тот ли HTTP- метод использован |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |