

13_1

1. Create Person class with the following attributes (String firstName, String lastName). Write constructor, toString and getters methods.
2. Create Account class with the following attributes String IBAN, Person client, double balance. Write constructor, toString and getters methods.
3. Create BankProcessing class with an attribute Account[] bankAccounts. Write a constructor, which takes an array of Accounts as argument. Write toString method. Write the method which return the array of all of the bank clients.
4. In the Account class write secureToString method which works like normal toString but instead IBAN put in the result string something like DE45 *****35 (replaces all characters of IBAN string to char '*' except four first and two last characters).
5. In the BankProcessing implement the secureToString method which displays the accounts using Account. secureToString method.
6. In the BankProcessing implement the method which returns the sum of all accounts balances.

13_2 Given an array of int. Implement the method that iterates the array and swaps the elements at adjacent indexes if the value of the element with bigger index is less than value of the element with lower index. Return the result array.

Example {1, 4, 2, 6, 1} -> {1, 2, 4, 1, 6}

By steps: {1, 4, 2, 6, 1} -> {1, 2, 4, 6, 1}

{1, 2, 4, 6, 1} -> {1, 2, 4, 1, 6}

13_3 Given an array of integers, and two indexes: index1, index2. Implement the method that returns the index of the element with lower value which is between index1 and index 2.

Example ({1, 4, 2, 6, 1}, 2, 4) -> 3

13_1

1. Создайте класс Person со следующими атрибутами (String firstName, String lastName). Реализуйте конструктор, методы toString и методы getters.
2. Создайте класс Account со следующими атрибутами String IBAN, Person client, double баланс. Реализуйте конструктор, методы toString и методы getters.
3. Создайте класс BankProcessing с атрибутом Account [] bankAccounts. Реализуйте конструктор, который принимает массив Account [] в качестве аргумента. Напишите метод toString. Напишите метод, который возвращает массив всех клиентов банка.
4. В классе Account напишите метод secureToString, который работает как обычный toString, но вместо IBAN вставляет в строку результата что-то вроде DE45 ***** 35 (заменяет все символы строки IBAN, кроме четырех первых и двух последних символов, на символ '*').

5. В BankProcessing реализуйте метод `secureToString`, который отображает счета, используя метод `Account.secureToString`.
6. В BankProcessing реализуйте метод, который возвращает сумму денег в банке, т.е. сумму балансов всех счетов.

13_2 Дан массив `int`. Реализовать метод, который пробегает по массиву и переставляет рядом стоящие элементы в случае, если элемент с меньшим индексом, больше, чем элемент с большим индексом. Вернуть получившийся массив.

Example {1, 4, 2, 6, 1} -> {1, 2, 4, 1, 6}

By steps: {1, 4, 2, 6, 1} -> {1, 2, 4, 6, 1}

{1, 2, 4, 6, 1} -> {1, 2, 4, 1, 6}

13_3 Дан массив `int` и два индекса: `index1`, `index2`. Реализовать метод, который вернет индекс элемента с минимальным значением, который находится в интервале между `index1` и `index 2`.

Example ({1, 4, 2, 6, 1}, 2, 4) -> 3