МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**ОТЧЁТ ПО ВЫПОЛНЕНИЮ**

**ЛАБОРАТОРНОЙ РАБОТЫ № 13  
по дисциплине "Функциональное и логическое программирование"**

Работу выполнила\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.Ш.Пивоварова

Направление подготовки 02.03.02 Фундаментальная информатика и

информационные технологии

Направленность (профиль) Математическое и программное обеспечение  
 компьютерных технологий

Преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А.Климец

Краснодар

2021

**КЛАССИЧЕСКИЕ АЛГОРИТМЫ**

**ЗАДАНИЕ 4**

Рассмотреть методы класса String, включить в отчет половину методов. Данные методы представлены в таблице 1.

**Таблица 1. Методы класса String**

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| fun compareTo(other: String): Int | Сравнивает текущий объект с указанным. Возвращает 0, если текущий равен указанному объекту, отрицательное число, если текущий меньше указанного, или положительное число, если текущий больше указанного. |
| fun equals(other: Any?): Boolean | Указывает, равен ли другой объект текущему. Реализация должна соответствовать следующим требованиям:   1. Reflexive: для любого ненулевого x, x.equals(x) должен возвращать true 2. Symmetric: для любых ненулевых x и y, x.equals(y) должен возвращать true тогда, когда y.equals(x) возвращает true 3. Transitive: для любых ненулевых x, y и z, если x.equals(y) возвращает true и y.equals(z) возвращает true, то x.equals(z) должен возвращать true 4. Consistent: для любых ненулевых x и y, множественные вызовы x.equals(y) последовательно возвращают true или последовательно возвращают false, при условии, что никакая информация, используемая в равных сравнениях для объектов, не изменена 5. Never equal to null: для любого ненулевого x, x.equals(null) должен возвращать false |
| fun get(index: Int): Char | Возвращает символ этой строки по указанному индексу.  Если индекс выходит за пределы этой строки, выдаёт исключение IndexOutOfBoundsException. |
| fun hashCode(): Int | Возвращает значение хэш-кода для объекта. Общий контракт hashCode:   1. Каждый раз, когда он вызывается для одного и того же объекта более одного раза, метод hashCode должен последовательно возвращать одно и то же целое число, при условии, что никакая информация, используемая в равных сравнениях для объекта, не изменяется. 2. Если 2 объекта равны согласно методу equals(), то вызов метода hashCode для каждого из двух объектов должен привести к одинаковому целочисленному результату. |
| operator fun plus(other: Any?): String | Возвращает строку, полученную путем объединения этой строки со строковым представлением данного другого объекта. |
| fun toString(): String | Возвращает строковое представление объекта. |
| fun subSequence(startIndex: Int, endIndex: Int): CharSequence | Возвращает новую последовательность символов, которая является подпоследовательностью этой последовательности символов, начиная с указанного startIndex и заканчивая прямо перед указанным endIndex. |
| inline fun CharSequence.all(     predicate: (Char) -> Boolean ): Boolean | Возвращает истину, если все символы соответствуют заданному предикату. |
| fun CharSequence.any(): Boolean | Возвращает истину, если последовательность символов содержит хотя бы один символ. |
| inline fun <K, V> CharSequence.associate(     transform: (Char) -> Pair<K, V> ): Map<K, V> | Возвращает карту, содержащую пары ключ-значение, предоставленные функцией преобразования, примененной к символам данной последовательности символов.  Если у любой из двух пар будет один и тот же ключ, на карту добавляется последняя.  Возвращенная карта сохраняет порядок итерации входа исходной последовательности символов. |
| fun String.byteInputStream(     charset: Charset = Charsets.UTF\_8 ): ByteArrayInputStream | Создает новый байтовый поток ввода для строки. |
| fun String.codePointAt(index: Int): Int | Возвращает символ (кодовую точку Unicode) по указанному индексу. |
| fun String.codePointCount(     beginIndex: Int,     endIndex: Int ): Int | Возвращает количество кодовых точек Unicode в указанном текстовом диапазоне этой String. |
| fun String.concat(str: String): String | Возвращает новую строку, состоящую из добавления указанной строки в конец текущей. |
| operator fun CharSequence.contains(     other: CharSequence,     ignoreCase: Boolean = false ): Boolean | Возвращает истину, если эта последовательность символов содержит указанную другую последовательность символов в качестве подстроки.  Параметр ignoreCase – true игнорирует регистр символов при сравнении строк. По умолчанию false. |
| fun String.contentEquals(charSequence: CharSequence): Boolean | Возвращает true, если эта строка равна содержимому указанной CharSequence, в противном случае - false. |
| fun CharSequence.count(): Int | Возвращает длину этой последовательности символов. |
| inline fun CharSequence.count(     predicate: (Char) -> Boolean ): Int | Возвращает количество символов, соответствующих заданному предикату. |
| fun CharSequence.drop(n: Int): CharSequence | Возвращает подпоследовательность этой последовательности символов с удаленными первыми n символами. |
| inline fun CharSequence.dropLastWhile(     predicate: (Char) -> Boolean ): CharSequence | Возвращает подпоследовательность этой последовательности символов, содержащую все символы, кроме последних символов, которые удовлетворяют заданному предикату. |
| inline fun CharSequence.elementAtOrElse(     index: Int,     defaultValue: (Int) -> Char ): Char | Возвращает символ по заданному индексу или результат вызова функции defaultValue, если индекс находится за пределами этой последовательности символов. |
| fun String.endsWith(     suffix: String,     ignoreCase: Boolean ): Boolean | Возвращает истину, если эта строка заканчивается указанным суффиксом. |
| inline fun CharSequence.filter(     predicate: (Char) -> Boolean ): CharSequence | Возвращает последовательность символов, содержащую только те символы из исходной последовательности символов, которые соответствуют заданному предикату. |
| inline fun CharSequence.find(     predicate: (Char) -> Boolean ): Char? | Возвращает первый символ, соответствующий данному предикату, или null, если такой символ не был найден. |
| inline fun CharSequence.first(): Char | Возвращает первый символ.  NoSuchElementException - если последовательность символов пуста. |
| inline fun <R> CharSequence.flatMap(     transform: (Char) -> Iterable<R> ): List<R> | Возвращает единый список всех элементов, полученных в результате вызова функции преобразования для каждого символа исходной последовательности символов. |
| inline fun CharSequence.forEach(action: (Char) -> Unit) | Выполняет заданное действие над каждым персонажем. |
| fun String.format(vararg args: Any?): String | Использует эту строку как строку формата и возвращает строку, полученную путем подстановки указанных аргументов с использованием языкового стандарта по умолчанию. |
| inline fun <C, R> C.ifBlank(     defaultValue: () -> R ): R where C : CharSequence, C : R | Возвращает эту последовательность символов, если она не пуста и не состоит только из пробельных символов, или результат вызова функции defaultValue в противном случае. |
| fun CharSequence.indexOf(     char: Char,     startIndex: Int = 0,     ignoreCase: Boolean = false ): Int | Возвращает индекс в этой строке первого вхождения указанного символа, начиная с указанного startIndex, или -1, если ничего не найдено. |
| fun String.intern(): String | Возвращает каноническое представление для этого строкового объекта. |
| operator fun CharSequence.iterator(): CharIterator | Итератор для символов данной последовательности символов. |
| fun CharSequence.lines(): List<String> | Разбивает эту последовательность символов на список строк, разделенных любой из следующих последовательностей символов: CRLF (\r\n), LF (перевод строки \n) или CR (возврат каретки \r).  Возвращаемые строки не содержат завершающих разделителей строк. |
| @ExperimentalStdlibApi fun Char.lowercase(): String | Преобразует этот символ в нижний регистр, используя правила отображения Unicode для инвариантной локали. |
| inline fun <R> CharSequence.map(     transform: (Char) -> R ): List<R> | Возвращает список, содержащий результаты применения данной функции преобразования к каждому символу в исходной последовательности символов. |
| fun String.match(regex: String): Array<String>? | Метод match производит поиск по заданной строке с использованием регулярного выражения и возвращает массив, предоставленные результаты этого поиска. |
| infix fun CharSequence.matches(regex: Regex): Boolean | Возвращает истину, если эта последовательность символов соответствует заданному регулярному выражению. |
| fun CharSequence.maxOrNull(): Char? | Возвращает самый большой символ или null, если символов нет. |
| inline fun CharSequence.maxOf(     selector: (Char) -> Double ): Double  inline fun CharSequence.maxOf(     selector: (Char) -> Float ): Float | Возвращает наибольшее значение среди всех значений, созданных функцией селектора, примененной к каждому символу в последовательности символов.  Если какое-либо из значений, созданных функцией селектора, - NaN, возвращаемый результат - NaN.  Исключения  NoSuchElementException - если последовательность символов пуста. |
| fun CharSequence.none(): Boolean | Возвращает истину, если последовательность символов не имеет символов. |
| inline fun <S : CharSequence> S.onEach(     action: (Char) -> Unit ): S | Выполняет заданное действие с каждым символом и впоследствии возвращает саму последовательность символов. |
| fun CharSequence.random(): Char | Возвращает случайный символ из этой последовательности символов.  Исключения  NoSuchElementException - если эта последовательность символов пуста. |
| fun CharSequence.removePrefix(     prefix: CharSequence ): CharSequence | Если эта последовательность символов начинается с данного префикса, возвращает новую последовательность символов с удаленным префиксом. В противном случае возвращает новую последовательность символов с теми же символами. |
| fun String.replaceAfter(     delimiter: Char,     replacement: String,     missingDelimiterValue: String = this ): String  fun String.replaceAfter(     delimiter: String,     replacement: String,     missingDelimiterValue: String = this ): String | Заменить часть строки после первого появления данного разделителя на заменяющую строку. Если строка не содержит разделителя, возвращается значение missingDelimiterValue, которое по умолчанию соответствует исходной строке. |
| fun CharSequence.replaceRange(     startIndex: Int,     endIndex: Int,     replacement: CharSequence ): CharSequence | Возвращает последовательность символов с содержимым этой последовательности символов, где ее часть в заданном диапазоне заменяется заменяющей последовательностью символов. |
| fun String.reversed(): String | Возвращает строку с символами в обратном порядке. |
| inline fun <R> CharSequence.runningFold(     initial: R,     operation: (acc: R, Char) -> R ): List<R> | Возвращает список, содержащий последовательные значения накопления, сгенерированные путем применения операции слева направо к каждому символу и текущему значению аккумулятора, которое начинается с начального значения.  Обратите внимание, что значение acc, передаваемое в функцию операции, не должно изменяться; в противном случае это повлияет на предыдущее значение в результирующем списке.  operation - функция, которая берет текущее значение аккумулятора и символ и вычисляет следующее значение аккумулятора. |
| fun CharSequence.slice(indices: IntRange): CharSequence | Возвращает последовательность символов, содержащую символы исходной последовательности символов в указанном диапазоне индексов. |
| fun CharSequence.split(     vararg delimiters: String,     ignoreCase: Boolean = false,     limit: Int = 0 ): List<String> | Разбивает эту последовательность символов на список строк вокруг вхождений указанных разделителей.  delimiters - одна или несколько строк, которые будут использоваться в качестве разделителей.  ignoreCase - true игнорировать регистр символов при сопоставлении разделителя. По умолчанию false.  limit - максимальное количество возвращаемых подстрок. Ноль по умолчанию означает, что ограничение не установлено.  Чтобы избежать неоднозначных результатов, когда строки в разделителях имеют общие символы, этот метод переходит от начала до конца этой строки и сопоставляет в каждой позиции первый элемент в разделителях, который в данном случае равен разделителю в этой позиции. |
| fun String.substring(startIndex: Int, endIndex: Int): String | Возвращает подстроку этой строки, начиная с startIndex и заканчивая прямо перед endIndex. |
| fun CharSequence.take(n: Int): CharSequence | Возвращает подпоследовательность этой последовательности символов, содержащую первые n символов из этой последовательности символов, или всю последовательность символов, если эта последовательность символов короче. |
| fun CharSequence.toList(): List<Char> | Возвращает список, содержащий все символы. |
| fun CharSequence.toHashSet(): HashSet<Char> | Возвращает новый HashSet всех символов. |
| fun <C : MutableCollection<in Char>> CharSequence.toCollection(     destination: C ): C | Добавляет всех символов в указанную целевую коллекцию. |

**Задание 5**

Рассмотреть класс целое число. Разобрать и включить в отчет половину методов с использованием официальной документации языка. Данные методы представлены в таблице 2.

**Таблица 2. Методы класса целое число**

|  |  |
| --- | --- |
| **Метод** | **Описание** |
| infix fun and(other: Int): Int | Выполняет побитовую операцию И между двумя значениями. |
| operator fun compareTo(other: Byte): Int  operator fun compareTo(other: Short): Int  operator fun compareTo(other: Int): Int  operator fun compareTo(other: Long): Int  operator fun compareTo(other: Float): Int  operator fun compareTo(other: Double): Int | Сравнивает это значение с указанным значением для заказа. Возвращает ноль, если это значение равно указанному другому значению, отрицательное число, если оно меньше другого, или положительное число, если оно больше другого. |
| operator fun dec(): Int | Уменьшает это значение на 1. |
| operator fun div(other: Byte): Int  operator fun div(other: Short): Int  operator fun div(other: Int): Int  operator fun div(other: Long): Long  operator fun div(other: Float): Float  operator fun div(other: Double): Double | Делит это значение на другое значение. |
| fun equals(other: Int): Boolean | Указывает, равны ли 2 объекта класса Int. Вместо Int может быть Any?, объект любого класса. |
| fun hashCode(): Int | Возвращает значение хэш-кода для объекта. |
| operator fun inc(): Int | Увеличивает это значение на 1. |
| fun inv(): Int | Инвертирует биты в этом значении. |
| operator fun minus(other: Byte): Int  operator fun minus(other: Short): Int  operator fun minus(other: Int): Int  operator fun minus(other: Long): Long  operator fun minus(other: Float): Float  operator fun minus(other: Double): Double | Вычитает другое значение из этого значения. |
| infix fun or(other: Int): Int | Выполняет побитовую операцию ИЛИ между двумя значениями. |
| operator fun rangeTo(other: Byte): IntRange  operator fun rangeTo(other: Short): IntRange  operator fun rangeTo(other: Int): IntRange  operator fun rangeTo(other: Long): LongRange | Создает диапазон (массив) от этого значения до указанного другого значения. |
| operator fun rem(other: Byte): Int  operator fun rem(other: Short): Int  operator fun rem(other: Int): Int  operator fun rem(other: Long): Long  operator fun rem(other: Float): Float  operator fun rem(other: Double): Double | Вычисляет остаток от деления этого значения на другое значение. |
| infix fun shl(bitCount: Int): Int | Сдвигает это значение влево на количество битов bitCount. Функция shr сдвигает вправо. |
| operator fun times(other: Byte): Int  operator fun times(other: Short): Int  operator fun times(other: Int): Int  operator fun times(other: Long): Long  operator fun times(other: Float): Float  operator fun times(other: Double): Double | Умножает это значение на другое значение. |
| fun toString(): String | Возвращает строковое представление объекта. |
| infix fun ushr(bitCount: Int): Int | Сдвигает это значение вправо на количество битов bitCount, заполняя крайние левые биты нулями. |
| fun <T : Comparable<T>> T.coerceIn(     minimumValue: T?,     maximumValue: T? ): T | Гарантирует, что это значение находится в указанном диапазоне minimumValue..maximumValue.  Вернёт это значение, если оно находится в диапазоне, или minimumValue, если это значение меньше minimumValue, или maximumValue, если это значение больше maximumValue. |