Федеральное государственное автономное образовательное учреждение

высшего образования

Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

Высшая школа «Киберфизические системы и управление»

**Отчет по лабораторной работе**

по дисциплине «Системный подход к разработке программного обеспечения»

|  |  |  |
| --- | --- | --- |
| Выполнил:  студент гр. 3530902/00201 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Позолотин О. В. |
|  | <*подпись*> |  |
|  |  |  |
|  |  |  |
| Руководитель:  Кандидат т.н. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Нестеров С. А. |
|  | <*подпись*> |  |

«\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2023 г.

Санкт-Петербург

2023

**Презентация 4.1**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Презентация 4.2**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Практика**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Изображение выглядит как текст

Автоматически созданное описание**

**Блинов**

Класс String

Каждая строка, создаваемая с помощью оператора new или с помощью литерала (заключенная в двойные апострофы), является экземпляром класса String. Особенностью объекта класса String является то, что его значение не может быть изменено после создания объекта при помощи какого-либо метода класса, так как любое изменение строки приводит к созданию нового объекта. Класс String поддерживает несколько конструкторов, например: String(), String(String str), String(byte[] asciichar), String(char[] unicodechar), String(StringBuffer sbuf), String(StringBuilder sbuild) и др. Эти конструкторы используются для создания объектов класса String на основе инициализации значениями из массива типа char, byte и др. Например, при вызове конструктора.

Класс String содержит следующие методы для работы со строками:

String concat(String s) или «+» — слияние строк;

boolean equals(Object ob) и equalsIgnoreCase(String s) — сравнение строк

с учетом и без учета регистра соответственно;

int compareTo(String s) и compareToIgnoreCase(String s) — лексикографическое сравнение строк с учетом и без учета их регистра. Метод осуществляет

вычитание кодов первых различных символов вызывающей и передаваемой

строки в метод строк и возвращает целое значение. Метод возвращает значение нуль в случае, когда equals() возвращает значение true;

boolean contentEquals(StringBuffer ob) — сравнение строки и содержимого объекта типа StringBuffer;

String substring(int n, int m) — извлечение из строки подстроки длины

m-n, начиная с позиции n. Нумерация символов в строке начинается с нуля;

String substring(int n) — извлечение из строки подстроки, начиная с позиции n;

int length() — определение длины строки;

int indexOf(char ch) — определение позиции символа в строке;

static String valueOf(значение) — преобразование переменной базового

типа к строке;

String toUpperCase()/toLowerCase() — преобразование всех символов вызывающей строки в верхний/нижний регистр;

String replace(char с1, char с2) — замена в строке всех вхождений первого

символа вторым символом;

String intern() — заносит строку в «пул» литералов и возвращает ее объектную ссылку;

String trim() — удаление всех пробелов в начале и конце строки;

char charAt(int position) — возвращение символа из указанной позиции

(нумерация с нуля);

boolean isEmpty() — возвращает true, если длина строки равна 0;

char[] getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) — извлечение символов строки в массив символов;

static String format(String format, Object… args), format(Locale l, Stringformat, Object… args) — генерирует форматированную строку, полученную с использованием формата, интернационализации и др.; String[] split(String regex), String[] split(String regex, int limit) — поиск вхождения в строку заданного регулярного выражения (разделителя) и деление исходной строки в соответствии с этим на массив строк.

Во всех случаях вызова методов, изменяющих строку, создается новый объект типа String.

Классы StringBuilder и StringBuffer являются «близнецами» и по своему

предназначению близки к классу String, но в отличие от последнего содержимое и размеры объектов классов StringBuilder и StringBuffer можно изменять.

Основным отличием StringBuilder от StringBuffer является потокобезопасность последнего. Более высокая скорость обработки есть следствие отсутствия потокобезопасности класса StringBuilder. Его следует применять, если не

существует вероятности использования объекта в конкурирующих потоках.

С помощью соответствующих методов и конструкторов объекты классов

StringBuffer, StringBuilder и String можно преобразовывать друг в друга.

Конструктор класса StringBuffer (также как и StringBuilder) может принимать

в качестве параметра объект String или неотрицательный размер буфера.

Следует обратить внимание на следующие методы:

void setLength(int n) — установка размера буфера;

void ensureCapacity(int minimum) — установка гарантированного минимального размера буфера;

int capacity() — возвращение текущего размера буфера;

StringBuffer append(параметры) — добавление к содержимому объекта

строкового представления аргумента, который может быть символом, значением базового типа, массивом и строкой;

StringBuffer insert(параметры) — вставка символа, объекта или строки

в указанную позицию;

StringBuffer deleteCharAt(int index) — удаление символа;

StringBuffer delete(int start, int end) — удаление подстроки;

StringBuffer reverse() — обращение содержимого объекта.

В классе присутствуют также методы, аналогичные методам класса String,

такие как replace(), substring(), charAt(), length(), getChars(), indexOf() и др