



CodingBootcamp

Week 1 Day 0

Аннотация: Этот документ является практическим руководством к Week 1 Day 0 в CodingBootcamp.

Академия Ковалевского



Содержание

1. Таблицы кодировок ASCII	2
2. Управление строками	3
3. Практическая работа	4



1. Таблицы кодировок ASCII

Справка

Определение

В вычислительных машинах символы не могут храниться иначе, как в виде последовательностей бит. Для передачи символа и его корректного отображения ему должна соответствовать уникальная последовательность нулей и единиц. Для этого были разработаны таблицы кодировок. ASCII представляет собой кодировку для представления десятичных цифр, латинского и национального алфавитов, знаков препинания и управляющих символов.

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Рекомендации

- Найти информацию о таблицах кодировок



2. Управление символами в строке

Справка

Извлечение символов

Для извлечения символов по индексу в классе `String` определен метод `charAt(int index)`. Он принимает индекс, по которому надо получить символ и возвращает извлеченный символ. Как и в массивах индексация начинается с нуля.

Рекомендации

- Найти [информацию](#) об управлении символами в строке в Java
- Повторить пройденную информацию о классе `StringBuilder`

3. Практическая работа

Класс StringUtils

isAsciiUppercase

Написать метод `isAsciiUppercase(char ch)`, который принимает на вход `char` и возвращает `true`, если символ — это буква от A до Z (верхний регистр) и `false` — любой другой ASCII символ. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом.

Прототип метода:

```
1. public static boolean isAsciiUppercase(char ch);
```

Пример:

```
1. isAsciiUppercase('A'); // → true;
2. isAsciiUppercase('d'); // → false;
3. isAsciiUppercase((char)257); // → IllegalArgumentException
```

isAsciiLowercase

Написать метод `isAsciiLowercase(char ch)`, который принимает на вход `char` и возвращает `true`, если символ — это цифра от 0 до 9 и `false` - любой другой ASCII символ. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом..

Прототип метода:

```
1. public static boolean isAsciiLowercase(char ch);
```

Пример:

```
1. isAsciiLowercase('A'); // → false
2. isAsciiLowercase('d'); // → true
3. isAsciiLowercase((char)257); // → IllegalArgumentException
```



isAsciiNumeric

Написать метод `isAsciiNumeric(char ch)`, который принимает на вход `char` и возвращает `true`, если символ — это цифра от 0 до 9 и `false` — любой другой ASCII символ. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом.

Прототип метода:

```
1. public static boolean isAsciiNumeric(char ch);
```

Пример:

```
1. isAsciiNumeric('Z'); // → false
2. isAsciiNumeric('1'); // → true
3. isAsciiNumeric((char)257); // → IllegalArgumentException
```

isAsciiAlphabetic

Написать метод `isAsciiAlphabetic(char ch)`, который принимает на вход `char` и возвращает `true`, если символ — это буква от a до z (нижний регистр) или от A до Z (верхний регистр) и `false` — любой другой ASCII символ. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом.

Прототип метода:

```
1. public static boolean isAsciiAlphabetic(char ch);
```

Пример:

```
1. isAsciiAlphabetic('Z'); // → true
2. isAsciiAlphabetic('a'); // → true
3. isAsciiAlphabetic('1'); // → false
4. isAsciiAlphabetic((char)257); // → IllegalArgumentException
```





toAsciiUppercase

Написать метод `toAsciiUppercase(char ch)`, который принимает на вход `char` и возвращает символ в верхнем регистре. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом.

Прототип метода:

```
1. public static char toAsciiUppercase(char ch);
```

Пример:

```
1. toAsciiUppercase('a'); // → 'A'
2. toAsciiUppercase('B'); // → 'B'
3. toAsciiUppercase('1'); // → '1'
4. toAsciiUppercase('@'); // → '@'
5. toAsciiUppercase((char)257); // → IllegalArgumentException
```

toAsciiLowercase

Написать метод `toAsciiLowercase(char ch)`, который принимает на вход `char` и возвращает символ в нижнем регистре. Ожидается, что метод кинет `IllegalArgumentException`, если входящий символ не является ASCII символом.

Прототип метода:

```
1. public static char toAsciiLowercase(char ch);
```

Пример:

```
1. toAsciiLowercase('a'); // → 'a'
2. toAsciiLowercase('B'); // → 'b'
3. toAsciiLowercase('1'); // → '1'
4. toAsciiLowercase('@'); // → '@'
5. toAsciiLowercase((char)257); // → IllegalArgumentException
```





makeUppercase

Написать метод `makeUppercase(char[] input)`, который принимает на вход массив символов и возвращает строку символов в верхнем регистре. Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static StringBuilder makeUppercase(char[] input);
```

Пример:

```
1. makeUppercase({'h', 'e', 'l', 'l', 'o'}); // → "HELLO"
2. makeUppercase({'@', 'h', 'e', 'l', 'l', 'o', '1', '2'}); // → "@HELLO12"
3. makeUppercase({'h', 'e', 'l', 'l', 'o', (char)257}); // →
   IllegalArgumentException
```

makeLowercase

Написать метод `makeLowercase(char[] input)`, который принимает на вход массив символов и возвращает строку символов в нижнем регистре. Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static StringBuilder makeLowercase(char[] input);
```

Пример:

```
1. makeLowercase({'h', 'e', 'l', 'l', 'o'}); // → "hello"
2. makeLowercase({'@', 'h', 'e', 'l', 'l', 'o', '1', '2'}); // → "@hello12"
3. makeLowercase({'h', 'e', 'l', 'l', 'o', (char)257}); // →
   IllegalArgumentException
```





makeCamel

Написать метод `makeCamel(char[] input)`, который принимает на вход массив символов и возвращает строку символов в нижнем ВЕРХНЕМ регистре последовательно (начиная с нижнего). Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static StringBuilder makeCamel(char[] input);
```

Пример:

```
1. makeCamel({'h', 'e', 'l', 'L', 'o'}); // → "hElLo"
2. makeCamel({'@', 'h', 'e', 'l', 'L', 'o', '0', '1', '2'}); // → "@HeLlO012"
3. makeCamel({'h', 'e', 'l', 'L', 'o', (char)257}); // → IllegalArgumentException
```

isStringAlphaNumerical

Написать метод `isStringAlphaNumerical(char[] input)`, который принимает на вход массив символов и возвращает `true`, если все символы — это буквы или цифры, или пробелы. Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static boolean isStringAlphaNumerical(char[] input);
```

Пример:

```
1. isStringAlphaNumerical({'a', 'b', '1', ' '}); // → true
2. isStringAlphaNumerical({'a', ',', '1', ' '}); // → false
3. isStringAlphaNumerical({'a', (char)257, '1', ' '}); // →
   IllegalArgumentException
```





concatStrings

Написать метод `concatStrings(char[][] input)`, который принимает на вход двумерный массив символов, соединяет их и возвращает новый одномерный результирующий массив. Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static char[] concatStrings(char[][] input);
```

Пример:

```
1. concatStrings({'a', 'b'}, {'c', 'd'}); // → {'a', 'b', 'c', 'd'}  
2. concatStrings({'a', 'b', (char)257}, {'c', 'd'}); // → IllegalArgumentException
```

toInt

Написать метод `toInt(char[] input)`, которая принимает на вход массив символов и возвращает число `int`. Ожидается, что метод кинет `IllegalArgumentException`, если хоть один из входящих символов не является ASCII.

Прототип метода:

```
1. public static int toInt(char[] input);
```

Пример:

```
1. toInt({'1', '2'}); // → 12  
2. toInt({'0'}); // → 0  
3. toInt({'-', '1', '1'}); // → -11  
4. toInt({'(char)257', '2'}); // → IllegalArgumentException
```

⚠ Требования

Можно использовать

Классы: `java.util.stream.IntStream`.

Запрещено использовать

Методы: `Integer.valueOf`.

Классы: `String` (но можно использовать `StringBuilder`).