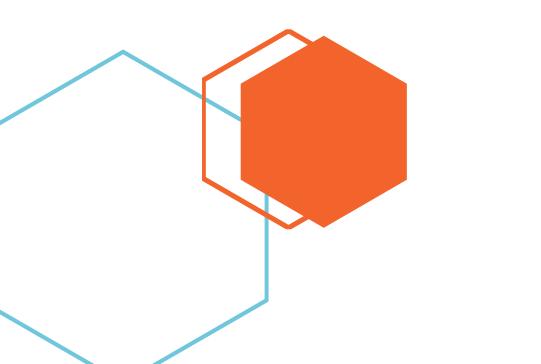
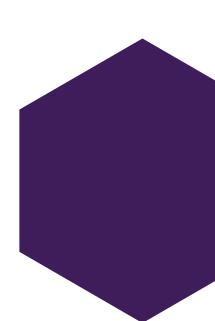


Week 1 Day 0

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

AKANENNIM KORANERCKOTO





#### • •

# Содержание

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



## 1. Таблицы кодировок ASCII

## Оправка

### 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.

						A	SCII	Coc	de Cl	hart						
لـ	0	1	2	3	ι 4	5	6	7	8	9	<sub>L</sub> A	В	С	D	E	<u> </u>
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	S0	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		;	٧	II	۸	?
4	0	Α	В	С	D	Ε	F	G	Н	Ι	J	K	L	М	N	0
5	Р	Q	R	S	T	U	V	W	Χ	Υ	Z	[	\	]	^	
6	,	а	b	U	d	е	f	g	h	i	j	k	ι	m	n	0
7	р	q	r	S	t	u	V	W	Х	у	Z	{		}	~	DEL

## **—** Рекомендации

• Найти информацию о таблицах кодировок



# 2. Управление символами в строке

## Оправка

### 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 <u>информацию</u> об управлении символами в строке в Java
- Повторить пройденную информацию о классе StringBuilder



3. Практическая работа

# KAacc 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:

```
    isAsciiUppercase('A'); // → true;
    isAsciiUppercase('d'); // → false;
    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);
```

```
    isAsciiLowercase('A'); // → false
    isAsciiLowercase('d'); // → true
    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:

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

## **isAsciiAlphabetic**

Написать метод is Ascii Alphabetic (char ch), который принимает на вход char и возвращает true, если символ — это буква от а до z (нижний регистр) или от A до Z (верхний регистр) и false — любой другой ASCII символ. Ожидается, что метод кинет Illegal Argument Exception, если входящий символ не является ASCII символом.

Прототип метода:

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

```
    isAsciiAlphabetic('Z'); // → true
    isAsciiAlphabetic('a'); // → true
    isAsciiAlphabetic('1'); // → false
    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:

```
    toAsciiUppercase('a'); // → 'A'
    toAsciiUppercase('B'); // → 'B'
    toAsciiUppercase('1'); // → '1'
    toAsciiUppercase('@'); // → '@'
    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);
```

```
    toAsciiLowercase('a'); // → 'a'
    toAsciiLowercase('B'); // → 'b'
    toAsciiLowercase('1'); // → '1'
    toAsciiLowercase('@'); // → '@'
    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:

```
    makeUppercase({'h', 'e', 'l', 'L', 'o'}); // → "HELLO"
    makeUppercase({'@','h', 'e', 'l', 'L', 'o', 'l', '2'}); // → "@HELLO12"
    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);
```

```
    makeLowercase({'h', 'e', 'l', 'L', 'o'}); // → "hello"
    makeLowercase({'@','h', 'e', 'l', 'L', 'o', 'l', '2'}); // → "@hello12"
    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 BEPXHEM 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:

```
    makeCamel({'h', 'e', 'l', 'L', 'o'}); // → "hElLo"
    makeCamel({'@','h', 'e', 'l', 'L', 'o', '0', 'l', '2'}); // → "@HeLlo012"
    makeCamel({'h', 'e', 'l', 'b', (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);
```

```
    isStringAlphaNumerical({'a', 'b', '1', ''}); // → true
    isStringAlphaNumerical({'a', ',', '1', ''}); // → false
    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:

```
    concatStrings({'a', 'b'}, {'c', 'd'}); // → {'a', 'b', 'c', 'd'}
    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

Kлассы: java.util.stream.IntStream.

#### Запрещено использовать

Методы: Integer.valueOf.

Классы: String (но можно использовать StringBuilder).