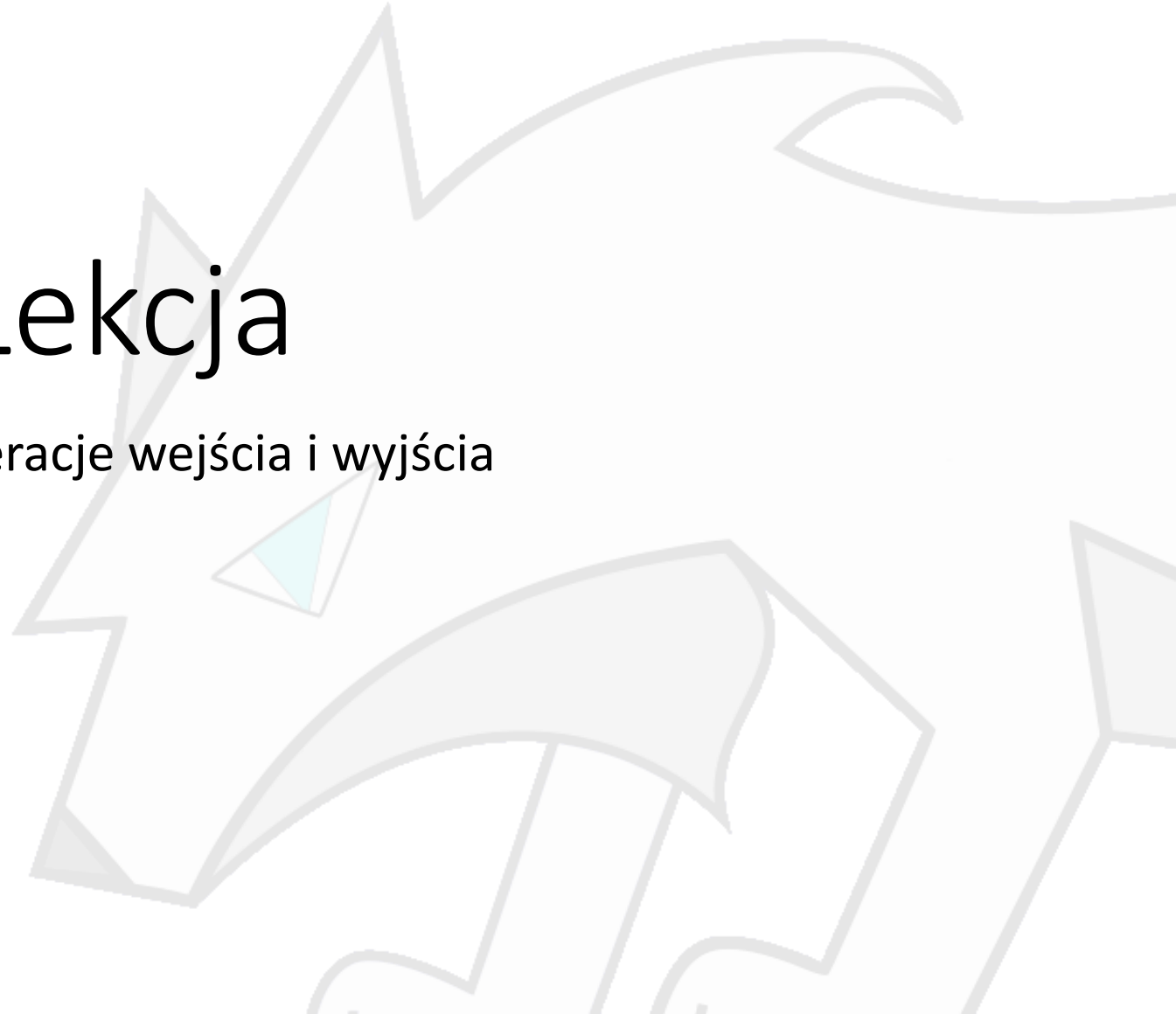


# 8. Lekcja

Klasa String i operacje wejścia i wyjścia



# String

	0	1	2	3	4	5
str	G	e	e	k	s	\0
Address	0x23452	0x23453	0x23454	0x23455	0x23456	0x23457

- Klasa String jest klasą wbudowaną w Javę. Służy ona do reprezentacji tekstów i jest jedną z najpopularniejszych klas Java.
- Implementacja klasy String to tak naprawdę char[] gdzie na końcu znajduje się znak końca, '\0' oznaczający koniec łańcucha znaku.

# Metody klasy String

- `str.isEmpty()` – sprawdza czy zmienna str to String pusty („”)
- `str.compareTo(„TEXT”)` – zamienia łańcuchy znaków na wartość leksykograficzną i porównuje zwracając 3 przedziały wartości 0, >0, <0.
- `str.endsWith(„TEXT”)` – sprawdza czy zmienna str kończy się wartością „TEXT”.
- `str.equals(„TEXT”)` – sprawdza czy zmienna str jest równa wartości „TEXT”.
- `str.replace(„TEXT”, „XYZ”)` – jeżeli w zmiennej str występuje ciąg znaków TEXT zostanie on zastąpiony przez XYZ.
- `str.concat(„TEXT”)` – dodaje ciąg znaków TEXT na koniec łańcucha znaków.
- `str.contains(„TEXT”)` – sprawdza czy w zmiennej str zawiera w sobie ciąg znaków „TEXT”



# Inicjalizacja obiektów klasy String

```
String a = "A";  
String b = new String( original: "B");  
String c = String.valueOf(new char[]{'C'});
```

- Inicjalizacja bezpośrednio przez cudzysłów. Każde użycie cudzysłowów powoduje inicjalizację obiektu klasy String nawet bez przypisania do zmiennej.
- Inicjalizacja bezpośrednio przez konstruktor.
- Inicjalizacja pośrednia przez statyczną metodę klasy String. Podczas inicjalizacji następuje konwersja danych z innego typu np. tablica charów, int, double itd

# Działanie operatorów +, == i metoda equals

- Klasa string jest jedyną dla której działa operator „+”. Służy on do łączenia dwóch łańcuchów znaków (konkatenacja)
- Operator == nie porównuje zawartości znaków tylko ich adres w pamięci.
- Metoda „equals” służy do porównywania wartości dwóch łańcuchów.

```
String s1 = "AAA";  
String s2 = "BBB";  
String s3 = "AAA";  
String s4 = new String( original: "BBB");  
String s5 = String.valueOf(new char[]{'A','A','A'});
```

```
System.out.println("Add s1 to s2. " + s1 + s2);  
System.out.println("String s1 and s3 is equale. " + (s1 == s3));  
System.out.println("String s2 and s4 is equale. " + (s2 == s4));  
System.out.println("String s1 and s5 is equale. " + (s1 == s5));  
System.out.println("String s1 and s3 is equale. " + s1.equals(s3));  
System.out.println("String s2 and s4 is equale. " + s2.equals(s4));  
System.out.println("String s1 and s5 is equale. " + s1.equals(s5));
```

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java

Demo > src > com > company > Main > main

Project Demo C:\Users\dobne\IdeaProjects\Demo

- .idea
- out
- src
  - com.company
    - Main
- Demo.iml
- External Libraries
- Scratches and Consoles

Main.java

```
String s1 = "AAA";
String s2 = "BBB";
String s3 = "AAA";
String s4 = new String( original: "BBB");
String s5 = String.valueOf(new char[]{'A','A','A'});

System.out.println("Add s1 to s2. " + s1 + s2);
System.out.println("String s1 and s3 is equale. " + (s1 == s3));
System.out.println("String s2 and s4 is equale. " + (s2 == s4));
System.out.println("String s1 and s5 is equale. " + (s1 == s5));
System.out.println("String s1 and s3 is equale. " + s1.equals(s3));
System.out.println("String s2 and s4 is equale. " + s2.equals(s4));
System.out.println("String s1 and s5 is equale. " + s1.equals(s5));
```

Run: Main

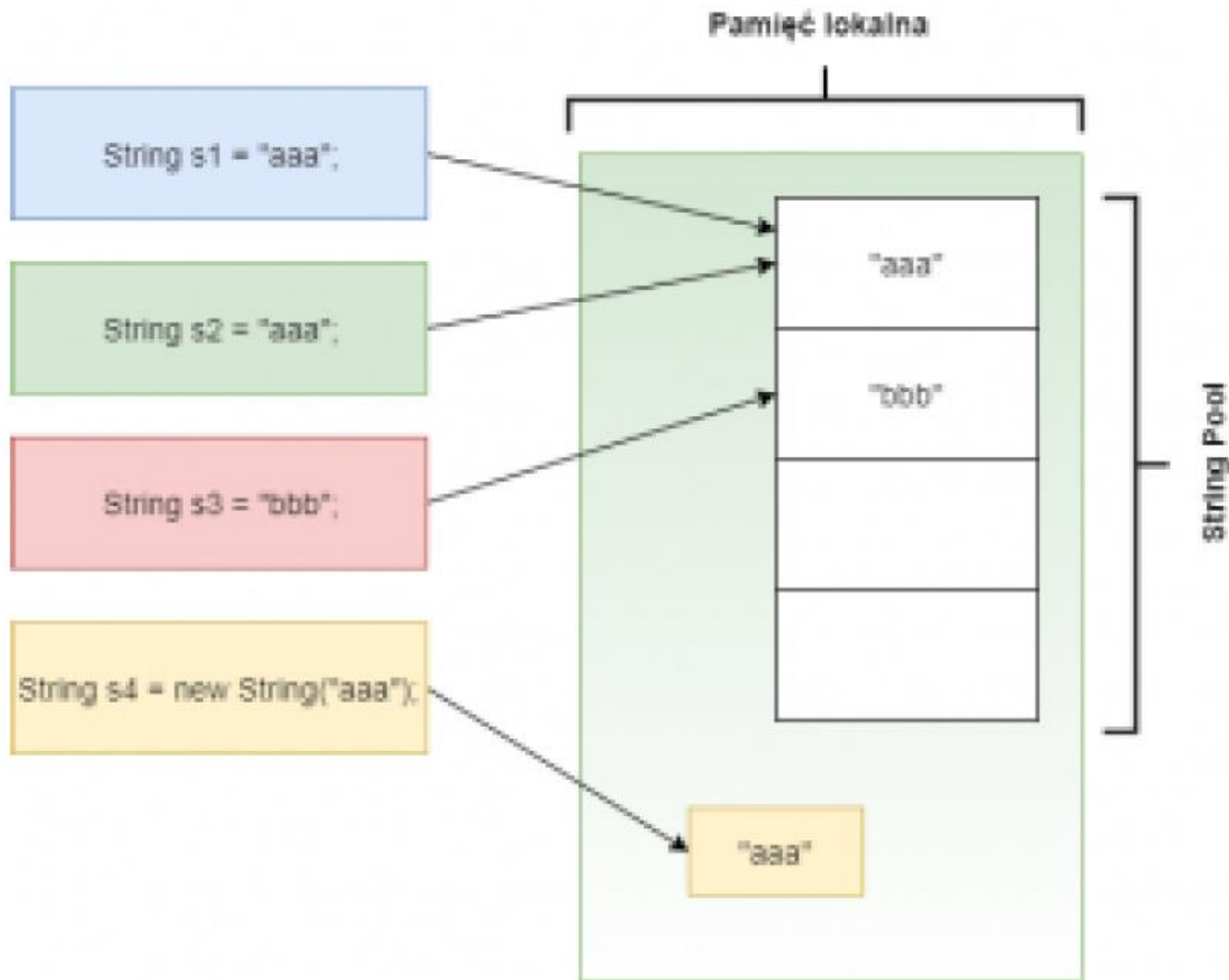
```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=60
Add s1 to s2. AAABBB
String s1 and s3 is equale. true
String s2 and s4 is equale. false
String s1 and s5 is equale. false
String s1 and s3 is equale. true
String s2 and s4 is equale. true
String s1 and s5 is equale. true

Process finished with exit code 0
```

4: Run TODO 6: Problems Terminal Build Event Log

Build completed successfully in 7 s 890 ms (a minute ago) 21:9 CRLF UTF-8 4 spaces

# String Pool



- Każdy obiekt inicjalizowany przez cudzysłów trafia do pamięci zwanej String Pool. Jest to forma optymalizacji pamięciowej.
- Każdy obiekt inicjalizowany w inny sposób trafia do pamięci lokalnej, poza String Pool.

# Standardowe wyjście

W języku Java standardowym wyjściem jest konsola programu. Możemy wyróżnić 3 główne metody do wyświetlania informacji na ekran.

- `.print()` – wypisuje na standardowe wyjście wartość parametru bez dodania nowej linii.
- `.println()` – wypisuje na standardowe wyjście wartość parametru dodając nową linię na końcu
- `.format()` – wypisuje na standardowe wyjście wartości z listy argumentów formatując je odpowiednio w wartości tekstowej.

```
System.out.print("Line without line separator");  
System.out.println("Line with line separator");  
System.out.format("Line with format");
```



FileEditViewNavigateCodeAnalyzeRefactorBuildRunToolsVCSWindowHelp

Demo [C:\Users\dobne\IdeaProjects\Demo] - Main.java

Demo > src > com > company > Main > main

Project

Demo

C:\Users\dobne\IdeaProjects\Demo

> .idea

> out

> src

> com.company

Main

Demo.iml

> External Libraries

> Scratches and Consoles

Main.java

```
5 public static void main(String[] args) {
6
7     System.out.print("This ");
8     System.out.print(" text ");
9     System.out.print("is in one line");
10    System.out.print('\n');
11
12    System.out.println();
13    System.out.println("This is in another line");
14    System.out.println("and this in another too");
15
16    System.out.format("The float number %f. ", 1.1);
17    System.out.format("Here is new line .%n");
18    System.out.format("First int number %d and second int number %d", 10, 20);
19
20 }
21
22 }
```

Run: Main

```
"C:\Program Files\Java\jdk-11.0.8\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.2\lib\idea_rt.jar=56
This  text is in one line

This is in another line
and this in another too
The float number 1,100000. Here is new line .
First int number 10 and second int number 20
Process finished with exit code 0
```

4: Run

TODO

6: Problems

Terminal

Build

Event Log

Build completed successfully in 7 s 916 ms (9 minutes ago)

19:1 CRLF UTF-8 4 spaces

# Standardowe wejście

Do pobierania wartości z klawiatury od użytkownika służy klasa Scanner. Klasę inicjalizuje się podając jako parametr standardowe wejście.

```
Scanner scanner = new Scanner(System.in);

byte numberByte = scanner.nextByte();
short numberShort = scanner.nextShort();
int numberInt = scanner.nextInt();
long numberLong = scanner.nextLong();

float numberFloat = scanner.nextFloat();
double numberDouble = scanner.nextDouble();

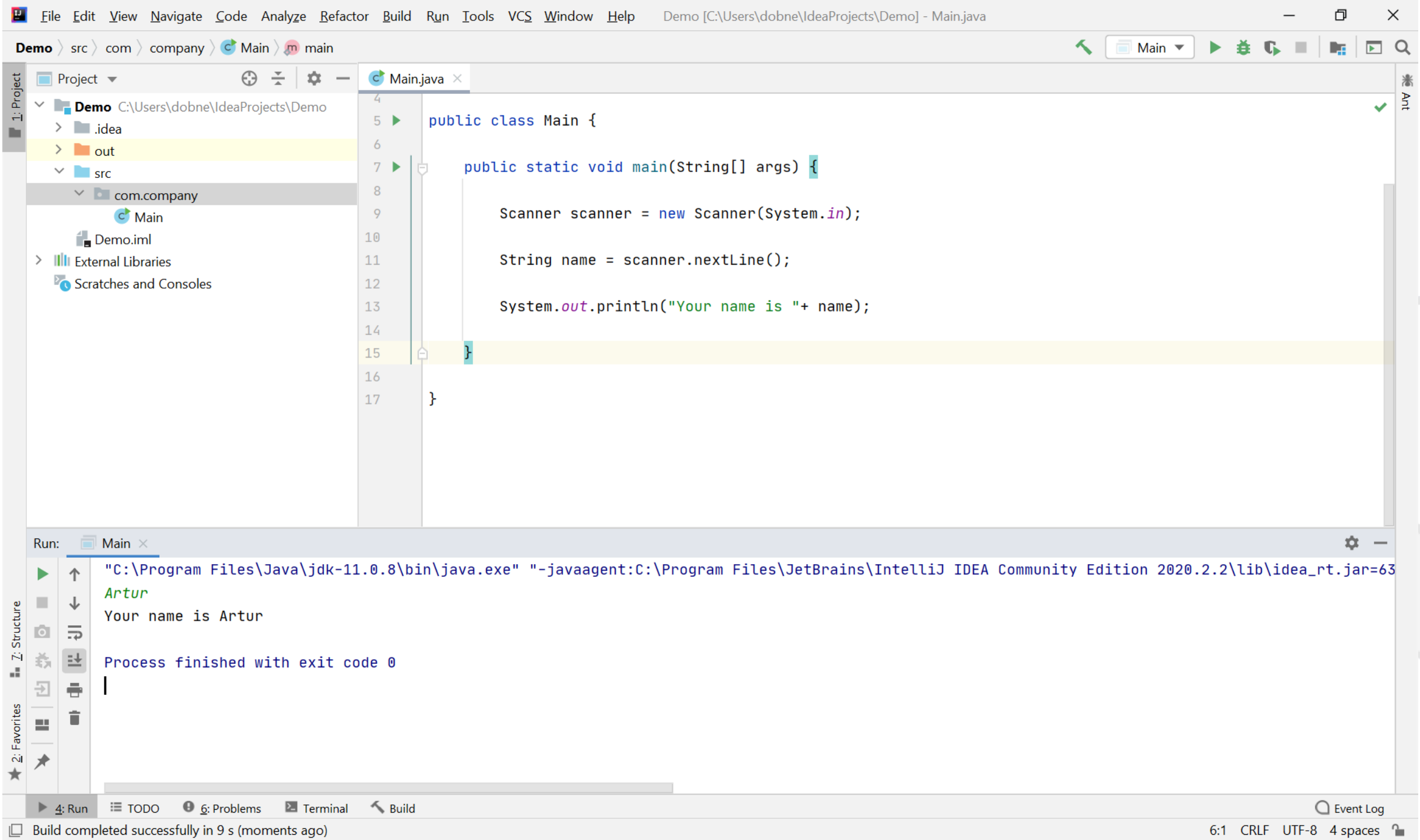
boolean bool = scanner.nextBoolean();

String string = scanner.nextLine();
```

```
Scanner scanner = new Scanner(System.in);

int number = scanner.nextInt();

System.out.println("My number from keyboard is " + number);
```



Czas na zadania ;)

