# Инструкции за работа с дебъгер

Този документ е допълнителен ресурс към курса [„Основи на програмирането“ @ СофтУни](https://softuni.bg/courses/programming-basics). Той има цел да запознае курсистите с **работата с дебъгер**, **начинът на използване** и неговата **приложимост**.

## Въведение

Често в процеса на писане на код допускаме **грешки**, които могат да са в резултат на **невнимание**, **недостатъчно знания** или непознаване на начина на действие на някои функции в езика. Мястото на проявление на тези грешки в една програма понякога е **трудно забележимо** и обикновено можем да съдим за тях само по крайния резултат. За тази цел средата за разработка ни предлага **инструмент**, с **когото** можем да проследим изпълнението на програмите **ред по ред и да наблюдаваме промяната на данните, така че да локализираме лесно мястото на грешките.**

## Начин на работа с дебъгер

### Стартиране на програмата в дебъг режим

За пример ще използваме решение на задачата "**Обръщение според възраст и пол**". Дадена е програма, която не изпълнява изискванията на даденото условие:

#### Обръщение според възраст и пол

Да се напише **конзолна програма**, която **прочита възраст** (реално число) и **пол** (“m” или “f”), въведени от потребителя, и отпечатва **обръщение** измежду следните:

* „Mr.“ – мъж (пол „m“) на 16 или повече години
* „Master“ – момче (пол „m“) под 16 години
* „Ms.“ – жена (пол „f“) на 16 или повече години
* „Miss“ – момиче (пол „f“) под 16 години

##### Примерен вход и изход:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 12  f | Miss | 17  m | Mr. | 25  f | Ms. | 13.5  m | Master |

##### Решение:

При изпълнение на дадената програма със стойности "f" за **пола** и "17" за **възрастта**, получаваме резултат на конзолата "Mr.". Това е **некоректно** спрямо условието, според което трябва да получим "**Ms.**". Грешката е трудно забележима и лесно може да се открие, ако проследим начина на изпълнение на програмата. За целта трябва да стартираме програмата в **дебъг режим** и сложим точка на начало за проследяване (в ляво от кода), в нашия случай в началото на серията от проверки:

import java.util.Scanner;

public class PersonalTitles {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double age = Double.parseDouble(scanner.nextLine());

String gender = scanner.nextLine().toLowerCase();

if(age >= 16) {

if("f".equals(gender)) {

System.out.println("Ms.");

} else if("m".equals(gender)) {

System.out.println("Mr.");

}

} else if(age < 16) {

if("m".equals(gender)) {

System.out.println("Master");

} else if("f".equals(gender)) {

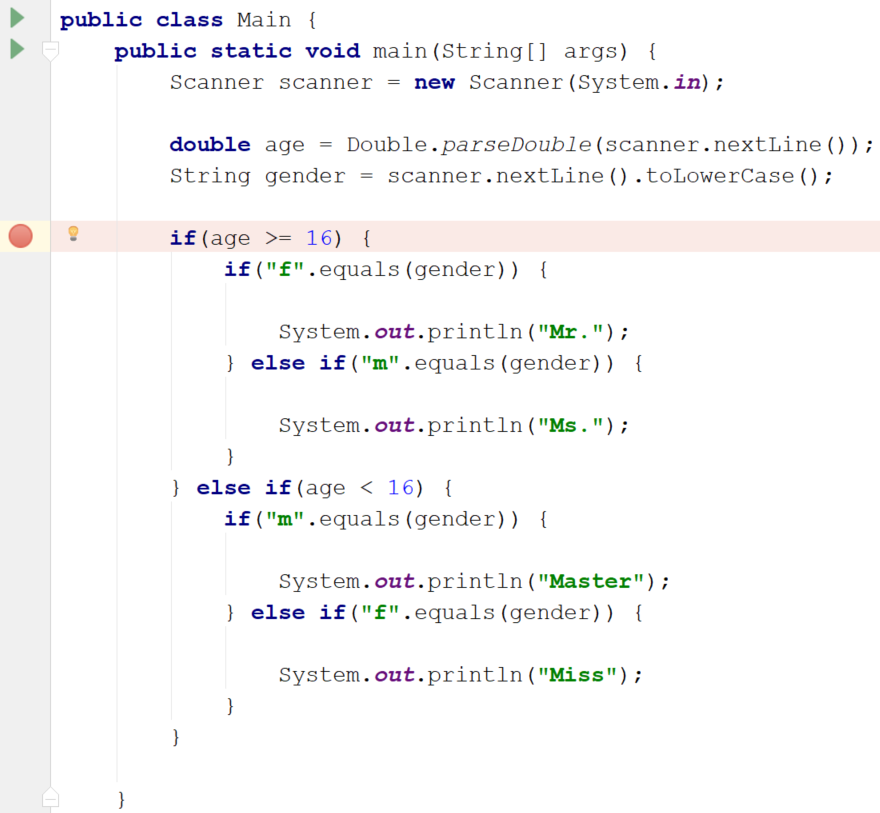
System.out.println("Miss");

}

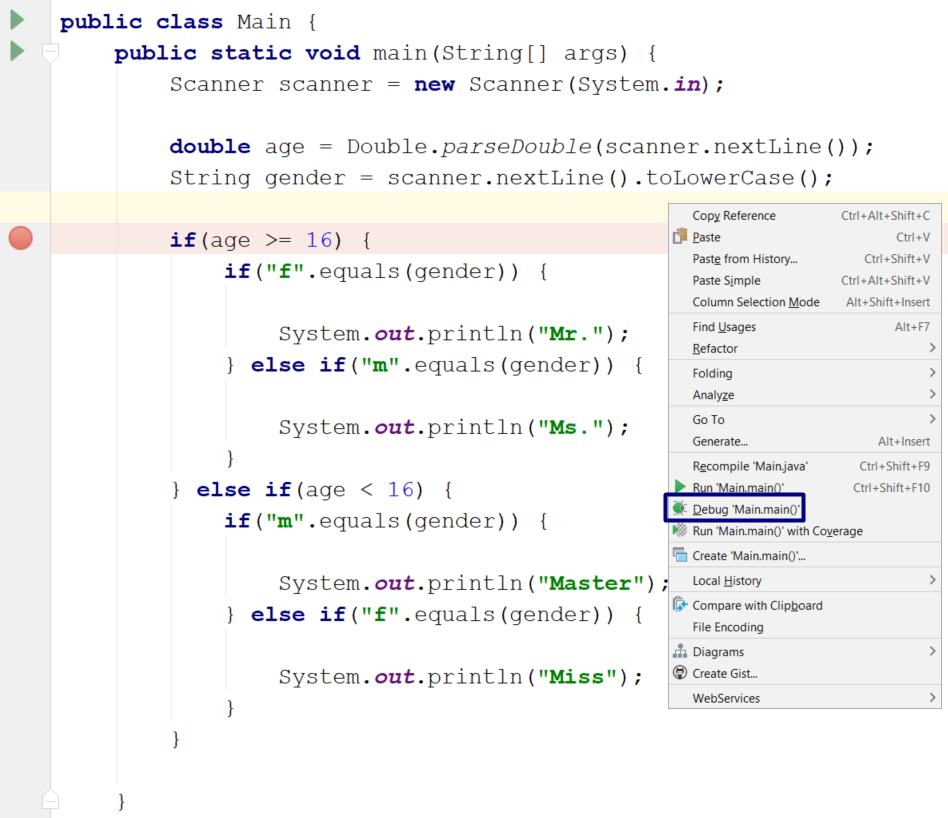
}

}

}



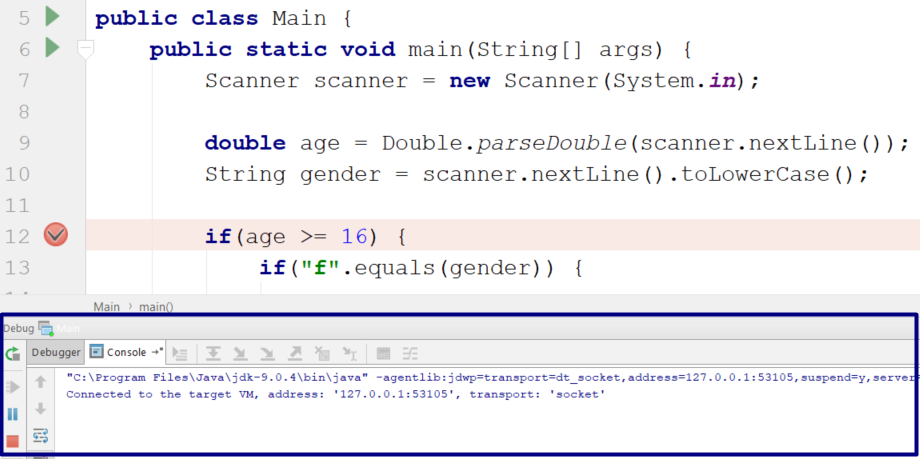
Стартираме програмата като достъпим менюто за команди в средата за разработка с десен бутон на мишката и изберем "**Debug {име на програмата}**":



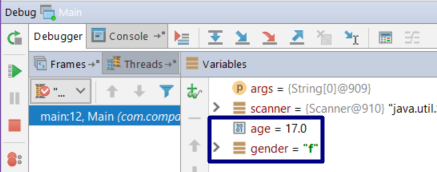
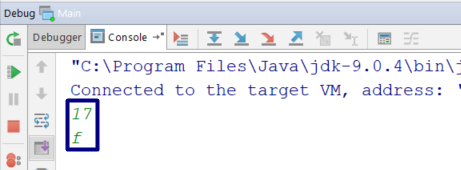
или Shift + F9.

### Работа с дебъгера и проследяване на изпълнението на програмата

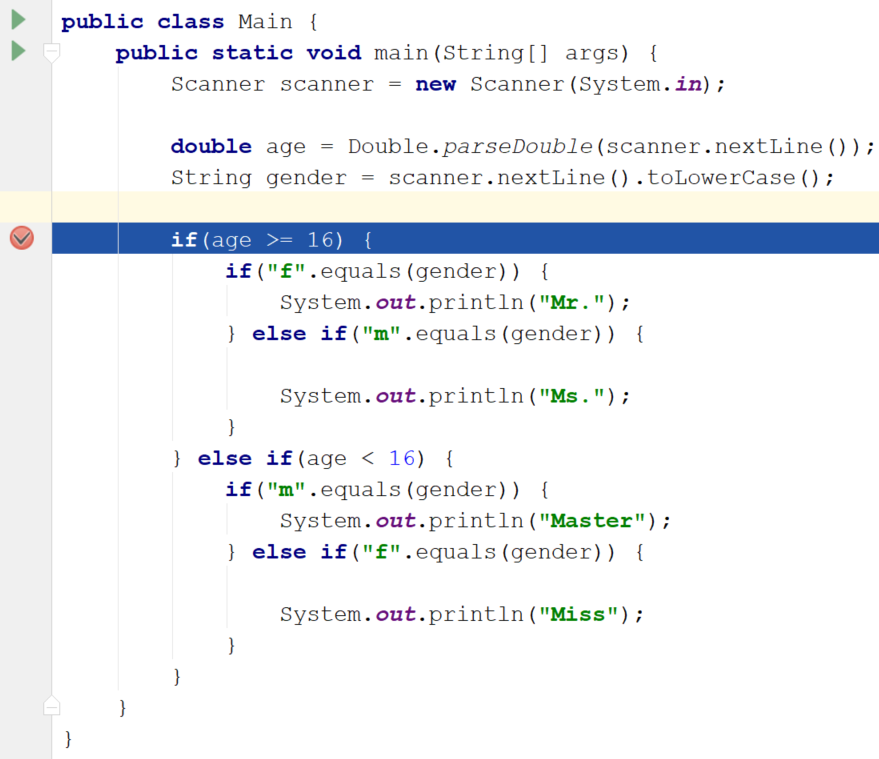
Ще се визуализира панела за дебъг режим:



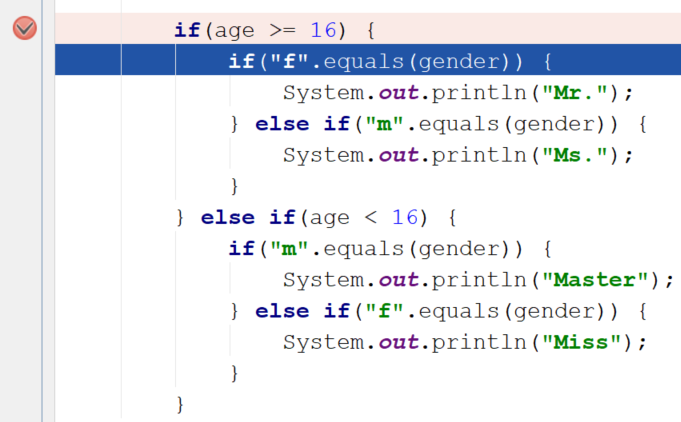
След като въведем стойност за двата входни параметъра и натиснем "**Enter**", в менюто "**Debugger**" можем да видим всички **дефинирани променливи и техните стойности** в програмата до момента:



Дебъгера е стигнал до реда, от когото ще започнем проследяването:

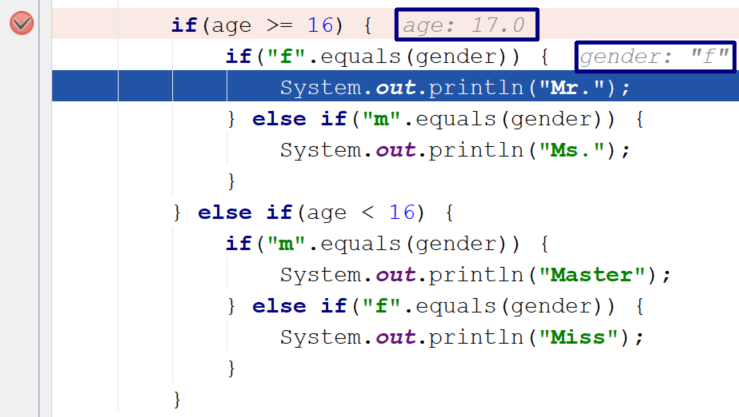


С **F8** преминаваме към **следващата стъпка**. Със **син цвят** е оцветен редът, който **предстои** да се изпълни. Според написаната логика на решението трябва да се премине към блокът от код във вложената if-конструкция:

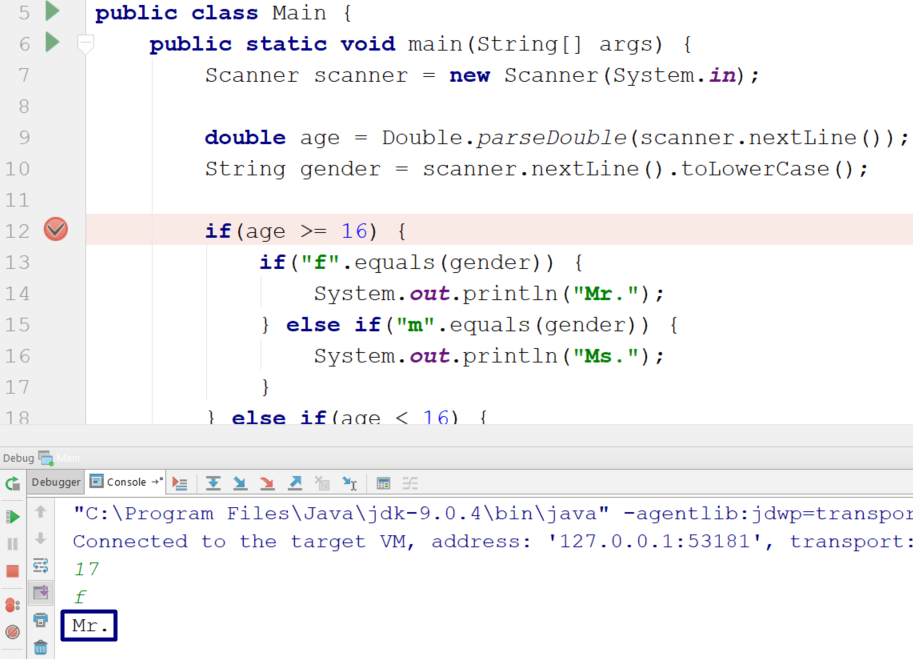


**Внимание: Командите на даден ред, до когото е стигнал дебъгера се изпълняват след като преминем на следващия.**

В конкретния случай следваща стъпка ще бъде да изпълним кода във **вложената if-конструкция** (в противен случай програмата ще премине към проверка на следващото условие):



Следващата стъпка на програмата ще бъде да принтира текста "Mr.":



След като командата се изпълни, ние можем да локализираме лесно грешката. Програмата приключва своето изпълнение, защото няма друг програмен код за изпълнение след серията от проверки. По този начин успяхме да намерим грешката в написаната логика и лесно можем да я поправим.