

Първи стъпки в програмирането

Задачи за упражнение в клас и за домашно към курса "[Основи на програмирането](#)" @ СофтУни.

Тествайте решенията си в **judge системата**: <https://judge.softuni.bg/Contests/Index/2387>

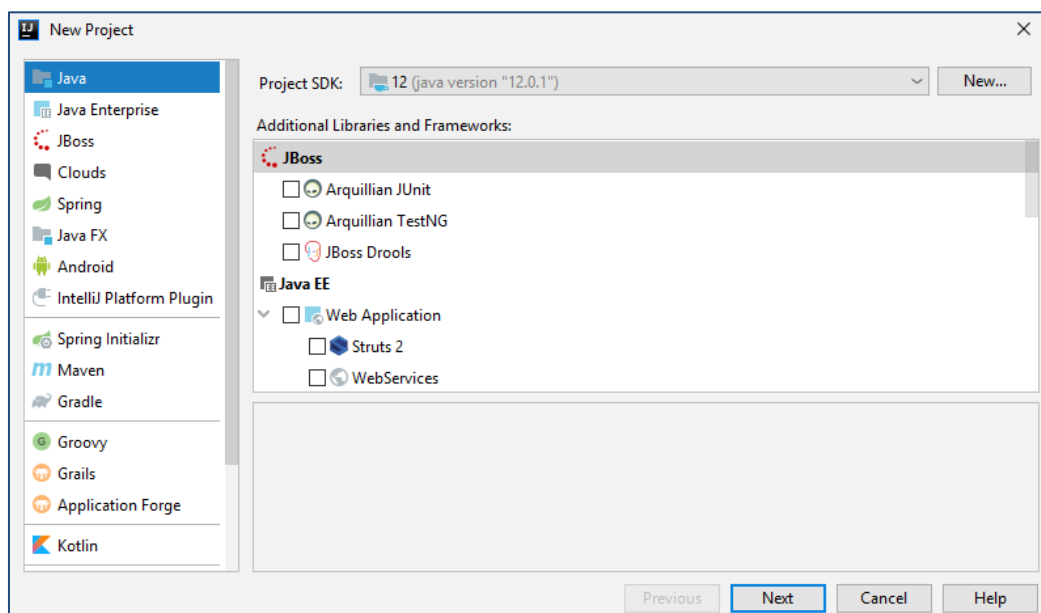
1. Конзолна програма "Hello SoftUni"

Напишете **конзолна Java програма**, която отпечатва текста "**Hello SoftUni**".

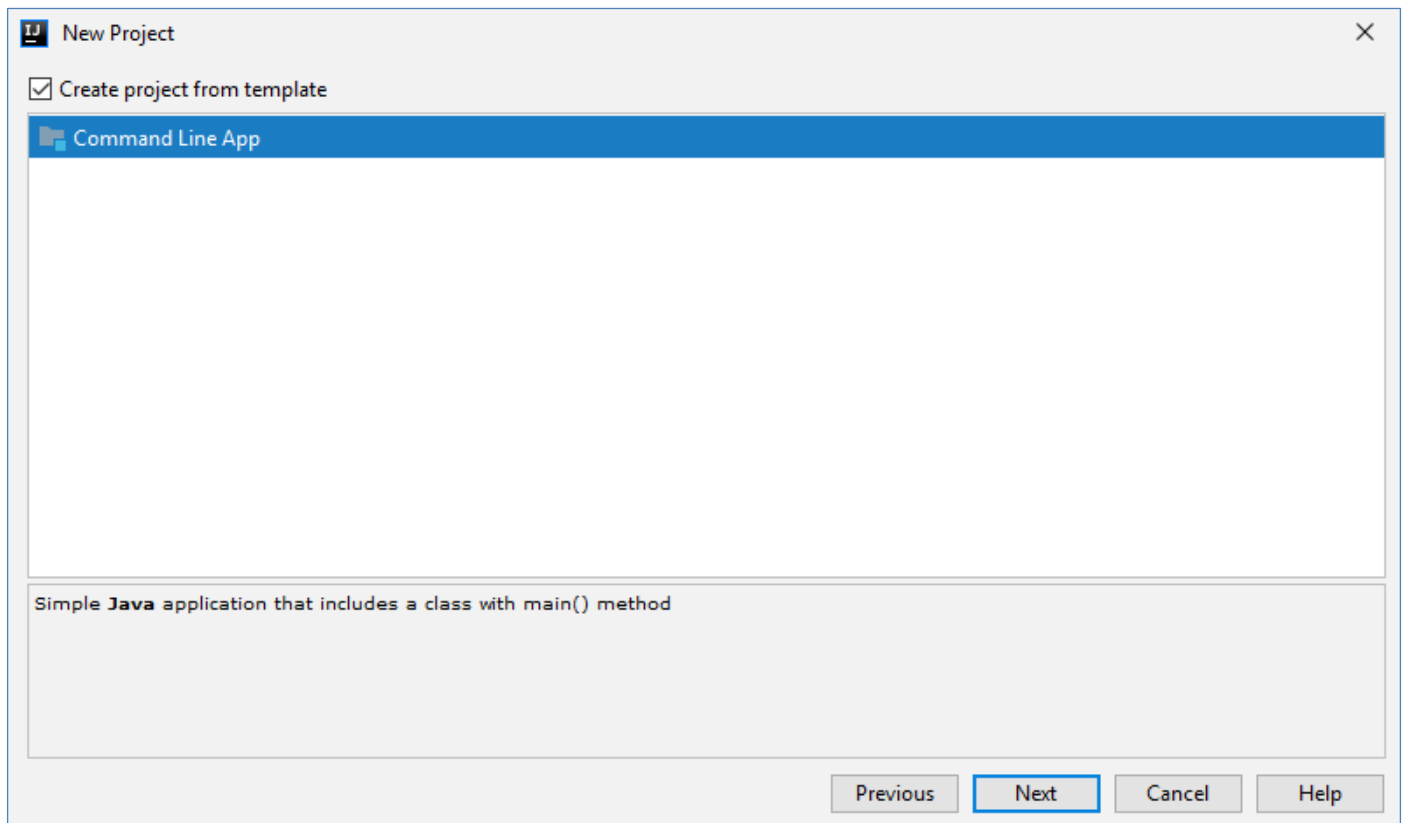
1. Стартирайте IntelliJ IDEA.
2. Създайте нов конзолен проект: [Create New Project].



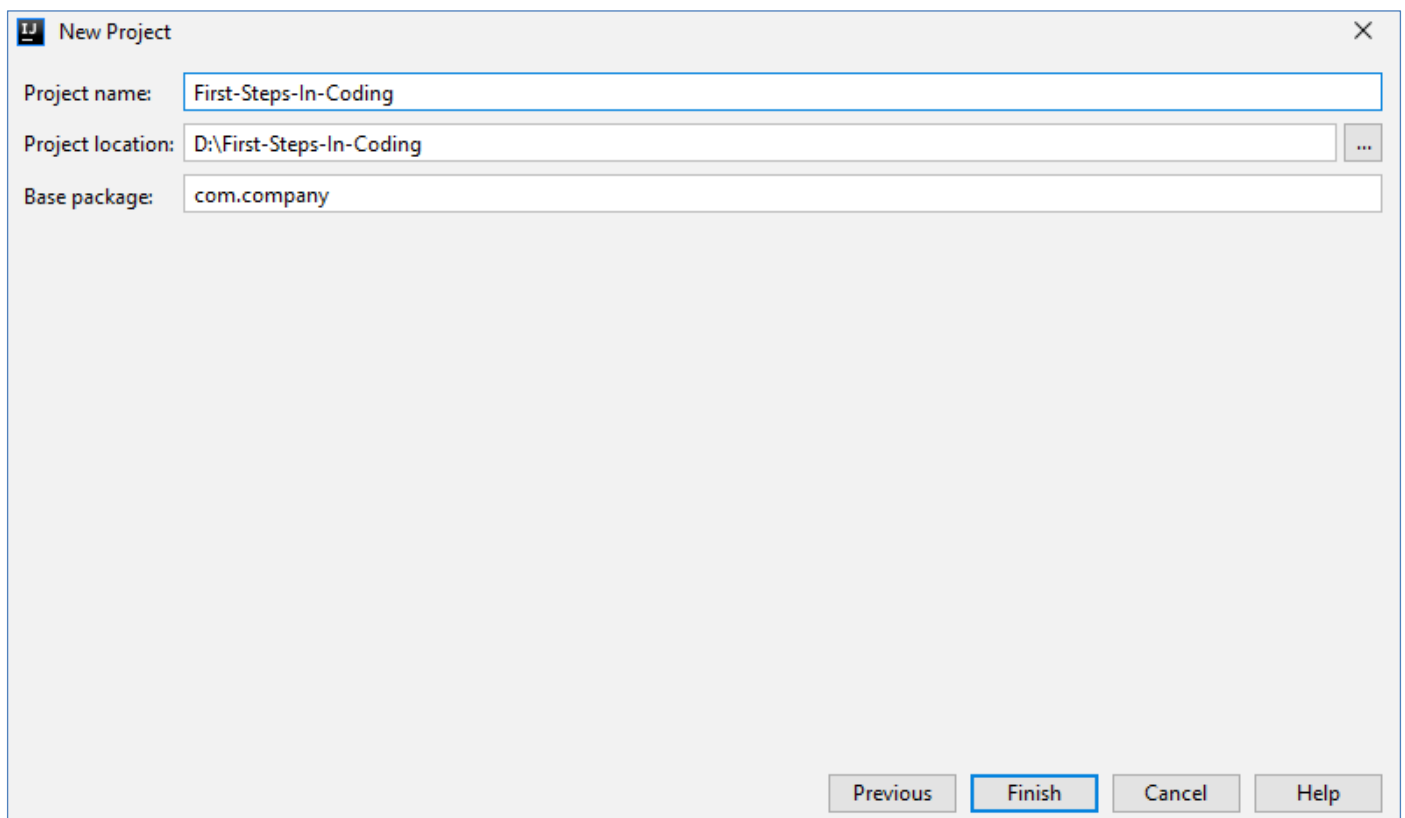
3. Изберете **Java** проект и продължете напред:



Изберете "Command Line App":



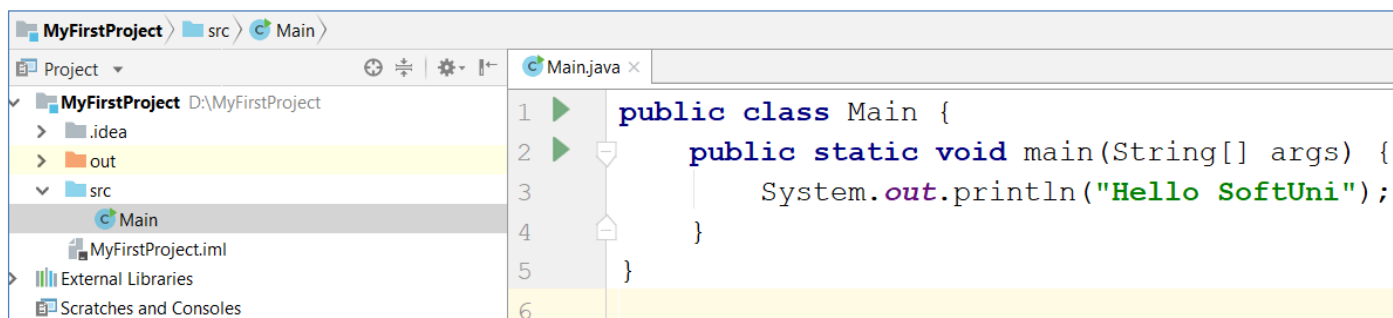
Задайте подходящо име на проекта и изберете директория, в която да бъде създаден:



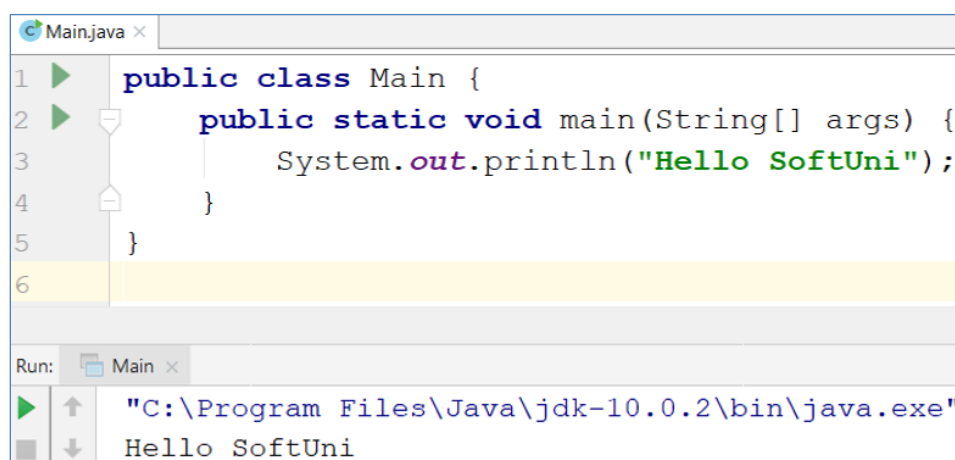
- Намерете секцията `main(String[] args)`. В нея се пише програмен код (команди) на езика **Java**.
- Напишете следния програмен код (команда за печатане на текста **"Hello SoftUni"**):

```
System.out.println("Hello SoftUni");
```

Кодът на програмата се пише отместен навътре с една табулация спрямо отварящата скоба `{`.



- Стартирайте** програмата с натискане на **[Ctrl+Shift+F10]**. Трябва да получите следния резултат:



- Тествайте** решението на тази задача в онлайн judge системата на СофтУни. За целта първо отворете <https://judge.softuni.bg/Contests/Compete/Index/2387#0>. Влезте с вашето потребителско име в СофтУни. Ще се появи прозорец за изпращане на решения за задача **"Hello SoftUni"**. Копирайте целия сорс код от IntelliJ и го поставете в полето за изпращане на решения:

Submit a solution

01. Hello SoftUni 02. Expression 03. Nums 1...20 04. Rectangle Area

01. Hello SoftUni

```
1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("Hello SoftUni");
4     }
5 }
6
```

Allowed working time: 0.100 sec.
Allowed memory: 16.00 MB
Size limit: 16.00 KB
Checker: Trim ?

Java code Submit

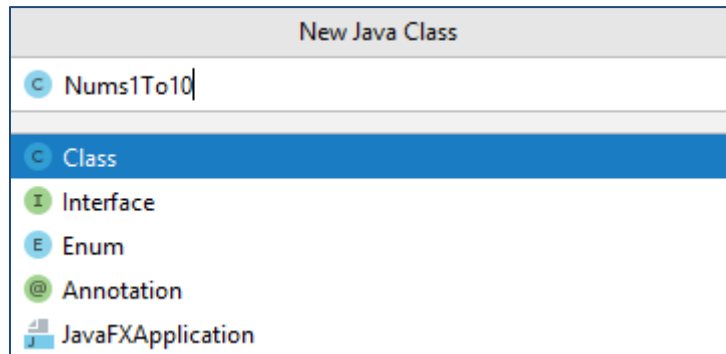
8. Изберете "Java code" от падащото меню, след което натиснете бутона "Submit".
9. Резултатът ще се появи в прозореца по-долу. За да го видите натиснете "Refresh" бутона:

Submissions			
<div>⏮ ⏪ 1 ⏩ ⏭</div>			<div>🔄</div>
Points	Time and memory used	Submission date	
✓ 100 / 100	Memory: 0.95 MB Time: 0.001 s	11:44:42 05.10.2018	Details
<div>⏮ ⏪ 1 ⏩ ⏭</div>			<div>🔄</div>

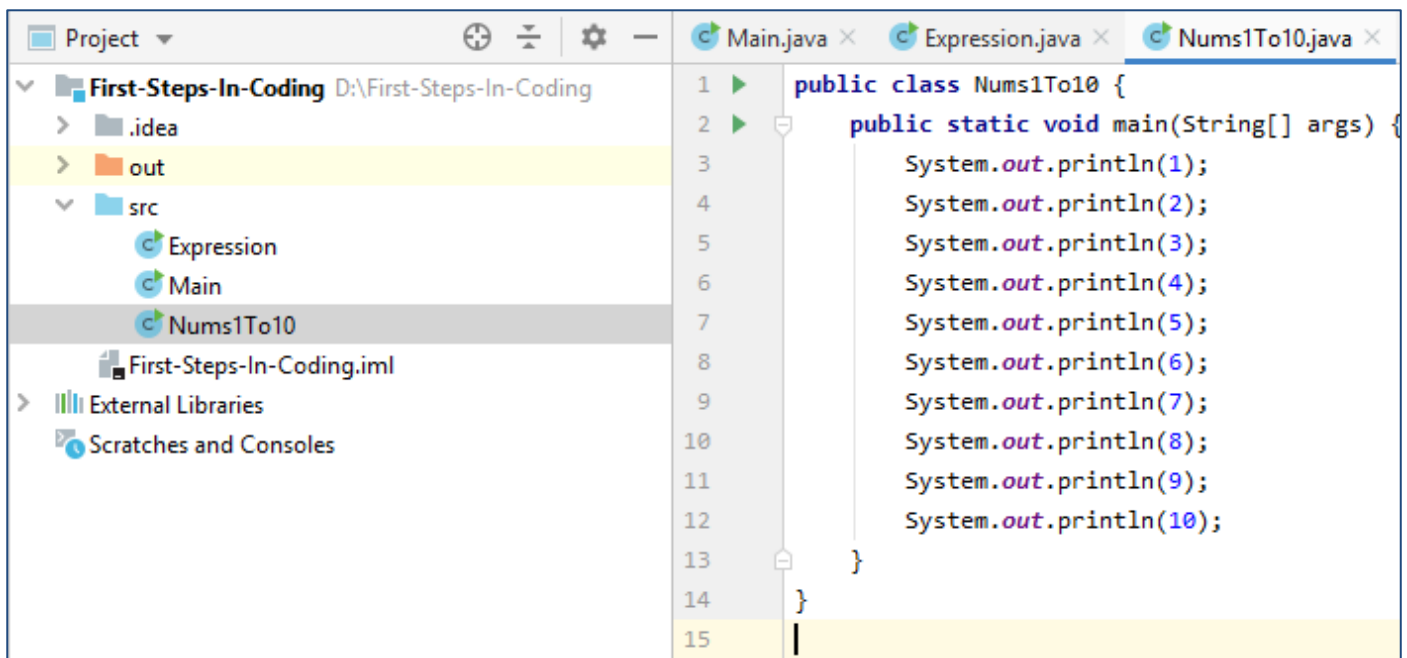
2. Числата от 1 до 10

Напишете Java конзолна програма, която отпечатва числата от 1 до 10 на отделни редове на конзолата.

1. Създайте нов Java клас с име "Nums1To10" (десен бутон върху "src" папката ➔ New ➔ Java Class):



2. Направете си **main** метод
3. Напишете 10 команди **System.out.println()**; една след друга, за да отпечатате числата от 1 до 10:



4. **Тествайте** вашето решение на задачата в judge системата.
5. Можете ли да напишете програмата по **по-умен начин**, така че да не повтаряте 10 пъти една и съща команда? Потърсете в Интернет информация за "[for loop Java](#)".

3. Пресмятане на лице на квадрат

Да се напише **конзолна програма**, която **въвежда** цяло число "a" и **пресмята** лицето на квадрат със страна "a".

Примерен вход и изход

ВХОД	ИЗХОД
5	25

Насоки

1. **Инициализирайте** променлива **side** и в нея запишете стойността въведена от конзолата:

```
public class SquareArea {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int a = Integer.parseInt(scan.nextLine());
    }
}
```

2. Инициализирайте втора променлива **area**, в която да запишете стойността за лицето на правоъгълника, получена по формулата **side * side**. Принтирайте получения резултат:

```
public class SquareArea {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int a = Integer.parseInt(scan.nextLine());
        int area = a * a;
        System.out.println(area);
    }
}
```

4. От инчове към сантиметри

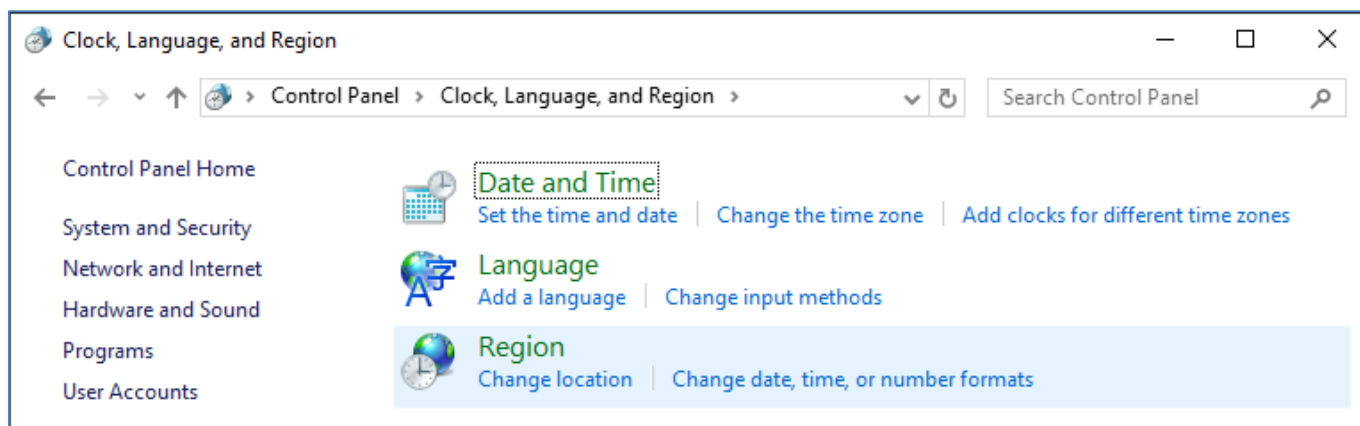
Да се напише програма, която **чете от конзолата реално число** и преобразува числото **от инчове в сантиметри**. За целта **умножете инчовете по 2.54** (защото 1 инч = 2.54 сантиметра).

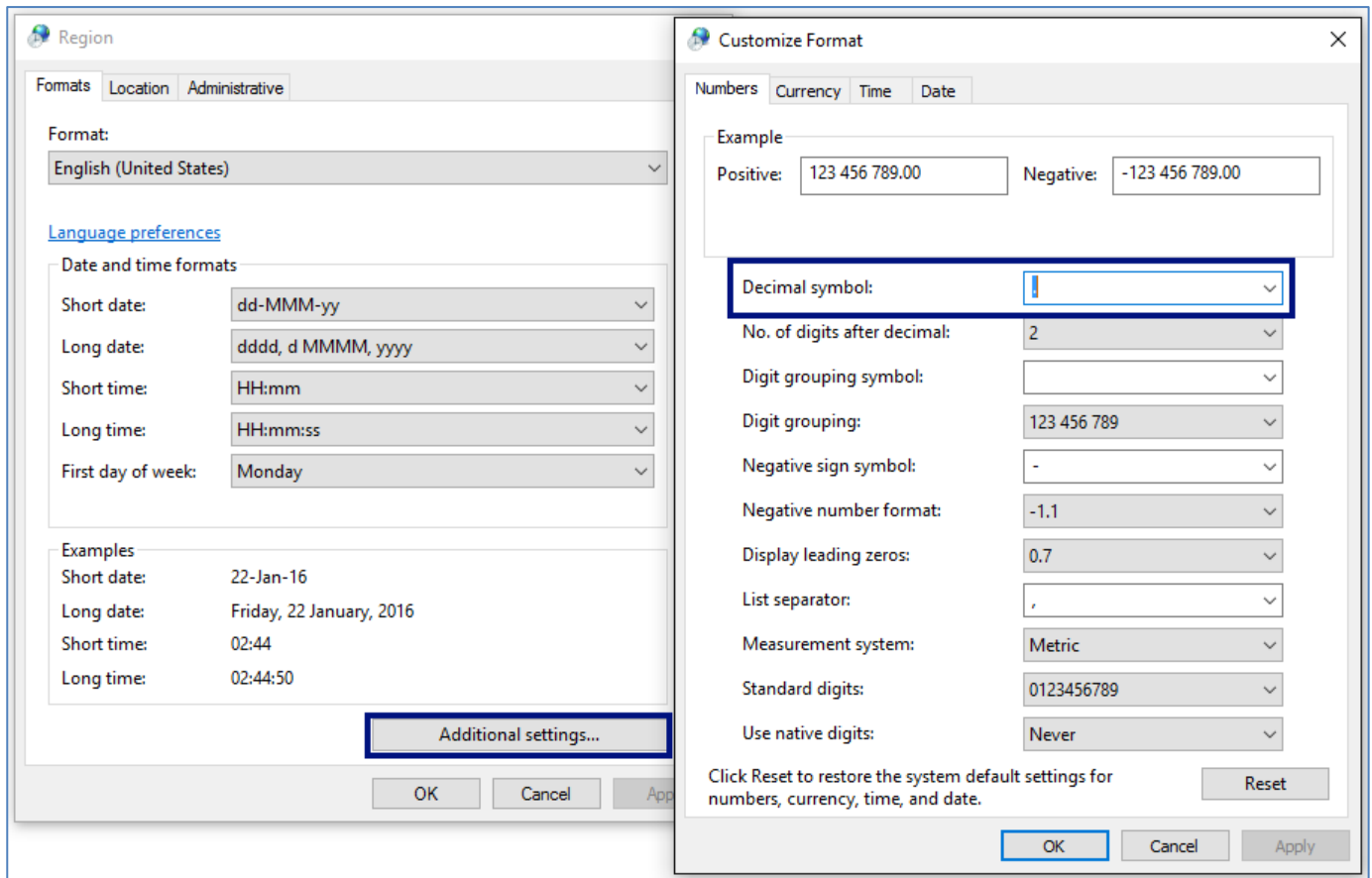
Примерен вход и изход

вход	изход
5	12.7

Внимание: в зависимост от регионалните настройки на операционната система, е възможно вместо **десетична точка** (US настройки) да се използва **десетична запетая** (BG настройки). Ако програмата очаква десетична точка и бъде въведено число с десетична запетая или на обратно (бъде въведена десетична точка когато се очаква десетична запетая), ще се получи следната грешка:

Препоръчително е **да промените настройките на компютъра си**, така че да се използва **десетична точка**:



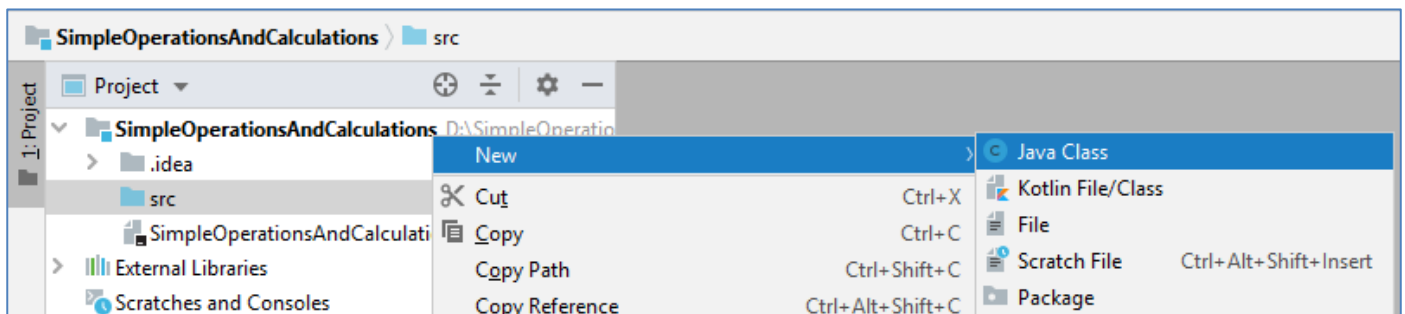


5. Поздрав по име

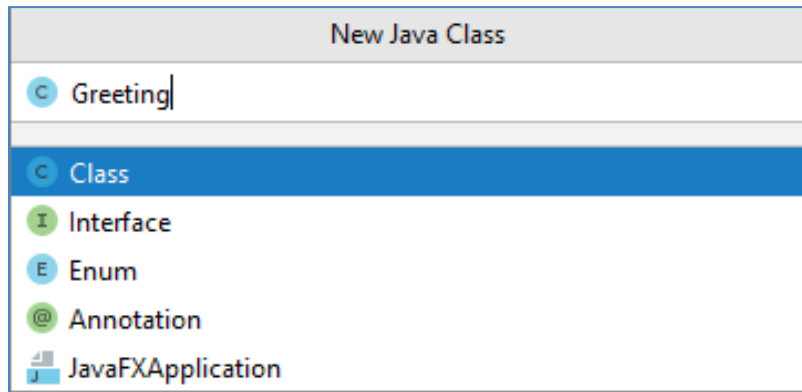
Да се напише програма, която **чете от конзолата име на човек** и отпечатва **"Hello, <name>!"**, където **<name>** е въведеното име от конзолата.

Насоки

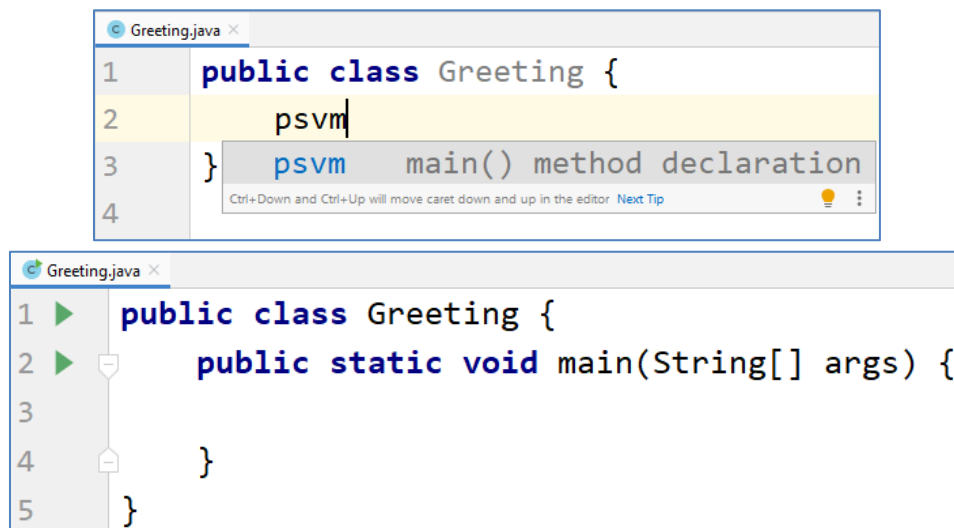
1. Първо създайте **нов Java клас** с име **"Greeting"** във вече съществуващия проект. Кликнете с десен бутон на мишката върху **"src"** папката в проекта и изберете **New -> Java Class**



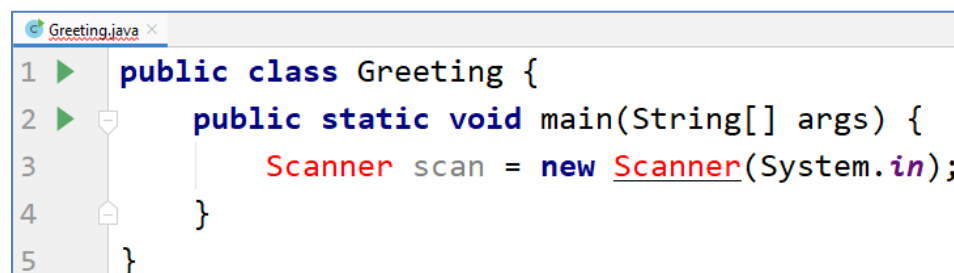
Задайте подходящо име:



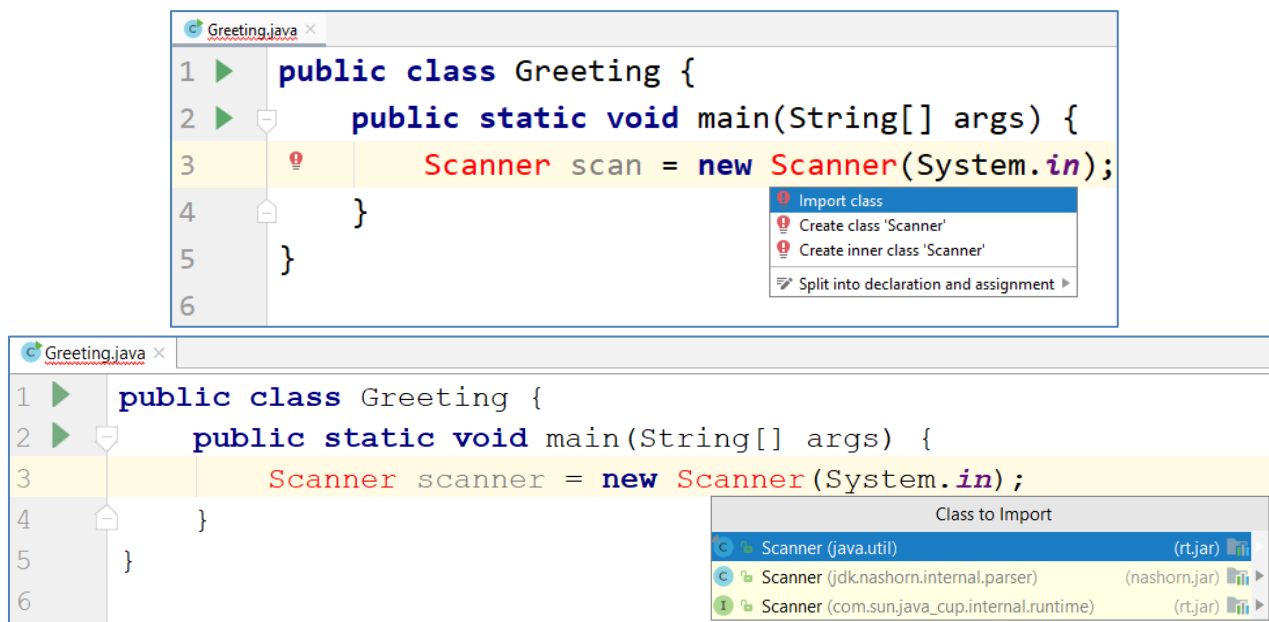
Създайте **main** метод в тялото на класа **Greeting**. Можете да направите това изписвайки съкращението **psvm** и натискайки два пъти бутона **Tab**:



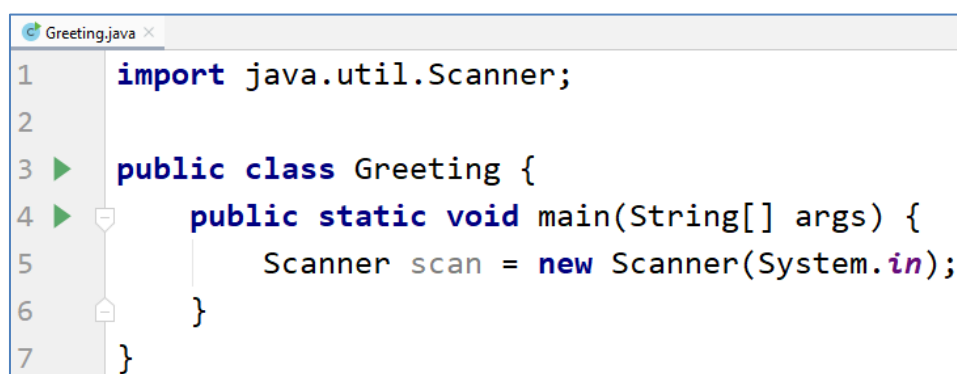
2. **Напишете кода** на програмата. Ако се затруднявате, може да ползвате примерния код по-долу
3. За да четете от конзолата, създайте обект **Scanner**:



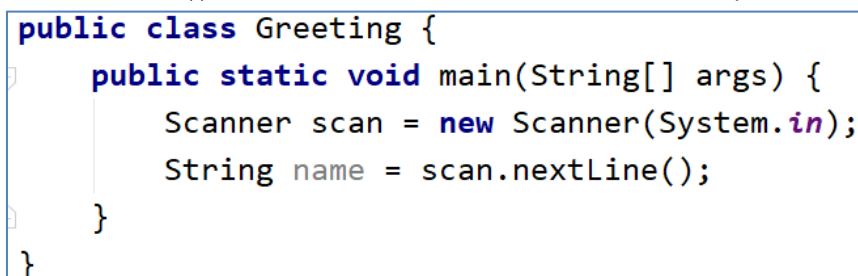
Ще забележите, че средата за разработка ни предупреждава, че този обект не може все още да бъде използван. За целта трябва да го добавим от **Java** пакета за разработка, който сме инсталирали (**jdk**). Можем да направим това, като напишем **"import java.util.Scanner;"** на първия ред от програмата, или поставим курсора върху червения текст в полето и натиснем **Alt + Enter** и изберем **Import Class -> Scanner(java.util):**



Трябва да получите следния резултат:



- Създайте променливата **name** от тип **String** и запазете в нея името, което получавате от конзолата, използвайки метода **nextLine()** от обекта **Scanner**, който създадохте в предишната стъпка:



- Изведете изхода на конзолата като използвате следния шаблон:

```
public class Greeting {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String name = scan.nextLine();
        System.out.printf("Hello, %s!", name);
    }
}
```

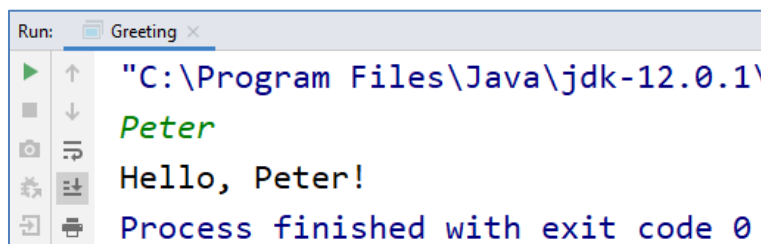
Как работи примера? Метода **printf** ни позволява чрез специална поредица от символи - **%s** да дефинираме място за текстова стойност в крайния текст, който ще бъде принтиран на конзолата. В нашият случай тази стойност ще бъде стойността на променливата **name**, която сме подали със запетайка след края на текста за принтиране.

Можете да прочетете повече за разновидностите от шаблони, които ни предлага езика **Java**, [ТУК](#).

Можем да постигнем същия резултат и с метода на **конкатенация** (долепяне) на **String** стойности и методите **println/print**:

```
public class Greeting {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String name = scan.nextLine();
        System.out.println("Hello, " + name + "!");
    }
}
```

6. Стартирайте програмата с **Ctrl + Shift + F10** и тествайте с различни входни примери:



```
Run: Greeting x
C:\Program Files\Java\jdk-12.0.1\
Peter
Hello, Peter!
Process finished with exit code 0
```

6. Съединяване на текст и числа

Напишете програма, която прочита от конзолата **име**, **фамилия**, **възраст** и **град** и печата съобщение от следния вид: "You are <firstName> <lastName>, а <age>-years old person from <town>."

Насоки

1. Създайте нов **Java** клас с името **ConcatenateData** и **main** метод в него
2. Въведете входните данни и ги запишете в променливи с подходящ тип данни:

```

public class ConcatenateData {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String firstName = scan.nextLine();
        String lastName = scan.nextLine();
        int age = Integer.parseInt(scan.nextLine());
        String town = scan.nextLine();
    }
}

```

3. Изведете на конзолата форматирания изход:

```

public class ConcatenateData {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String firstName = scan.nextLine();
        String lastName = scan.nextLine();
        int age = Integer.parseInt(scan.nextLine());
        String town = scan.nextLine();

        System.out.printf("You are %s %s, a %d-years old person from %s.",
            firstName, lastName, age, town);
    }
}

```

Можете да постигнете същия резултат с метода на конкатенация:

```

public class ConcatenateData {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String firstName = scan.nextLine();
        String lastName = scan.nextLine();
        int age = Integer.parseInt(scan.nextLine());
        String town = scan.nextLine();

        System.out.print("You are " + firstName + " " + lastName
            + ", a " + age + "-years old person from "
            + town + ".");
    }
}

```

Сами забелязвате, че метода на конкатенация има **по-дълъг запис** и създава **предпоставки за повече грешки**, спрямо метода на шаблоните.

4. Стартирайте програмата с **Ctrl + Shift + F10** и тествайте с различни входни примери

```
Run: ConcatenateData x
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaa
Peter
Petrov
21
Sofia
You are Peter Petrov, a 21-years old person from Sofia.
Process finished with exit code 0
```

7. Изготвяне на проекти

Напишете програма, която **изчислява колко часове** ще са необходими на един архитект, за да **изготви проектите** на няколко строителни обекта. Изготвянето на един проект отнема **три часа**.

Вход

От конзолата се четат **2 реда**:

1. Името на архитекта - текст
2. Брой на проектите - цяло число в интервала [0... 100]

Изход

На конзолата се отпечатва:

- "The architect {името на архитекта} will need {необходими часове} hours to complete {брой на проектите} project/s."

Примерен вход и изход

вход	изход	вход	изход
George 4	The architect George will need 12 hours to complete 4 project/s.	Sanya 9	The architect Sanya will need 27 hours to complete 9 project/s.

8. Зоомагазин

Напишете програма, която **пресмята нужните разходи** за закупуването на храна за кучета. Храната се пазарува **основно за кучета**, от зоомагазин, но понякога стопанинът им купува и за **животните на съседа му**. Една опаковка храна за **кучета е на цена 2.50лв.**, а всяка друга, която **не е** за тях **струва 4лв.**

Вход

От конзолата се четат **2 реда**:

1. Броят на кучетата – цяло число в интервала [0... 100]
2. Броят на останалите животни - цяло число в интервала [0... 100]

Изход

На конзолата се отпечатва:

"{крайната сума} lv."

Примерен вход и изход

вход	изход
5	28.5 lv.
4	

вход	изход
13	68.5 lv.
9	

9. Озеленяване на дворове

Божидара разполага с **няколко къщи** на Черноморието и **желае да озелени дворовете на някои от тях**, като по този начин създаде **уютна обстановка и комфорт на гостите си**, като за целта е наела фирма.

Напишете програма, която **изчислява необходимите средства**, които Божидара ще трябва да заплати на фирмата изпълнител на проекта. Цената на **един кв. м. е 7.61лв със ДДС**. Тъй като нейният двор е **доста голям**, фирмата изпълнител предлага **18% отстъпка от крайната цена**.

Вход

От конзолата се прочита само **един ред**:

1. **Кв. метри, които ще бъдат озеленени – реално число в интервала [0.00... 10000.00]**

Изход

На конзолата се отпечатват **два реда**:

- "The final price is: {крайна цена на услугата} lv."
- "The discount is: {отстъпка} lv."

Примерен вход и изход

Вход	Изход	Обяснения
550	The final price is: 3432.11 lv. The discount is: 753.39 lv.	Пресмятаме цената за озеленяване на целия двор: $550 * 7.61 = 4185.5$ лв. Приспадаме отстъпката от общата сума: $0.18 * 4185.5 = 753.39$ лв. Калкулираме крайната цена на услугата: $4185.5 - 753.39 \rightarrow 3432.11$ лв.
Вход	Изход	
150	The final price is: 936.03 lv. The discount is: 205.47 lv.	