# Упражнения: Прости пресмятания

Задачи за упражнение в клас и за домашно към курса [„Основи на програмирането“ @ СофтУни](https://softuni.bg/courses/programming-basics).

## Празно IntelliJ решение (Blank Project)

Създайте празно решение (**Blank Project**) във IntelliJ. Класовете във Java ни позволяват да сложим няколко решения в един и същи проект. Тази възможност е изключително удобна когато искаме да работим по няколко проекта и бързо да превключваме между тях или искаме да обединим логически няколко взаимосвързани проекта.

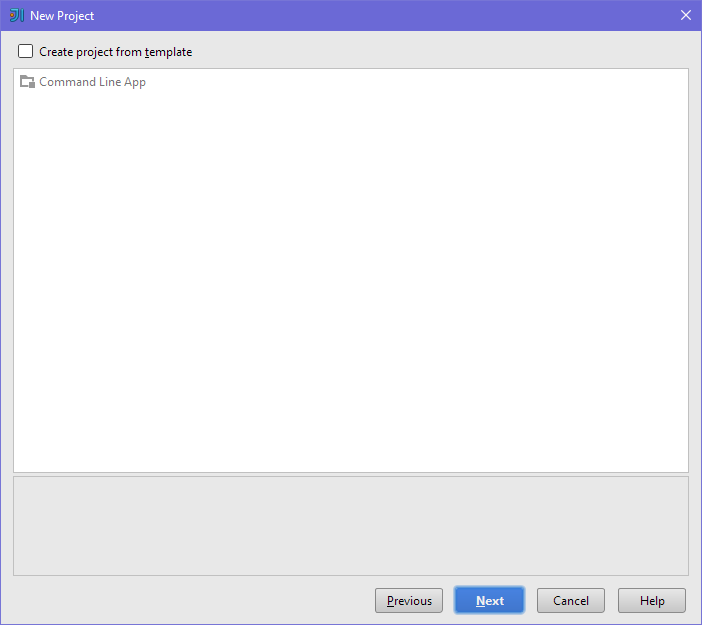
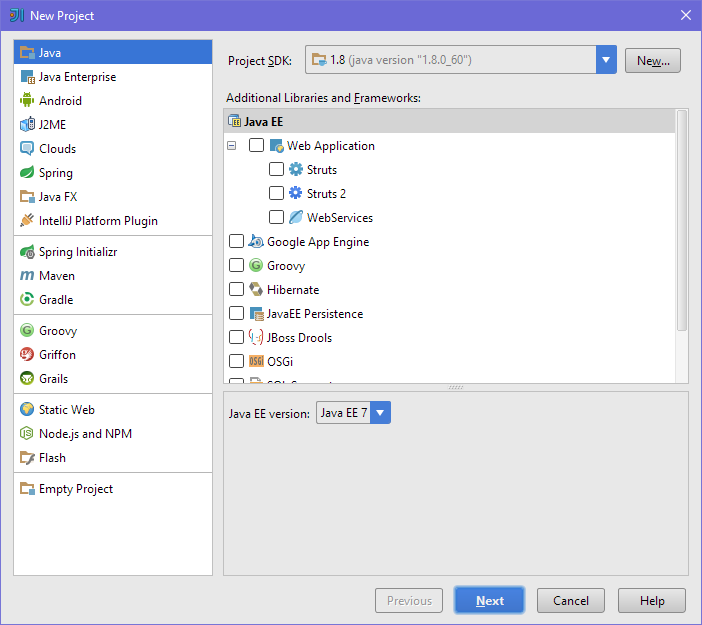
В настоящото практическо занимание ще използваме **празен проект и няколко класа** за да организираме решенията на задачите от упражненията – всяка задача в отделен клас и всички класове в общ проект.

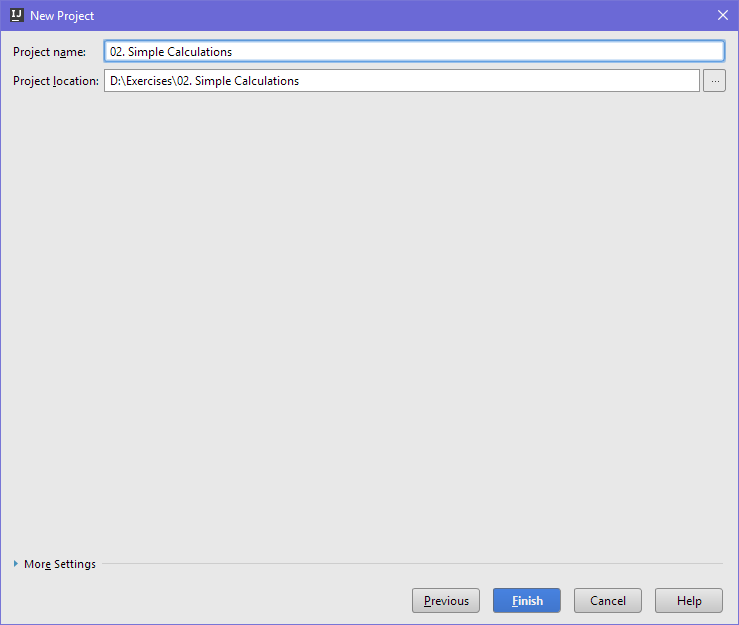
1. Стартирайте IntelliJ IDEA.
2. Създайте нов **проект**: [Create New Project].



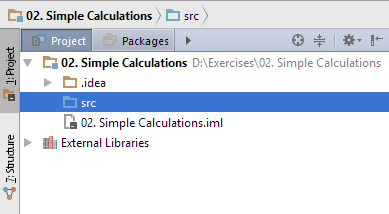
1. Изберете от диалоговия прозорец [Java] 🡪 [Next] 🡪 [Next] и дайте подходящо име на проекта, например “Simple Calculations”:

🡪





Сега имате създаден **празен IntelliJ проект** (с 0 класа в него):

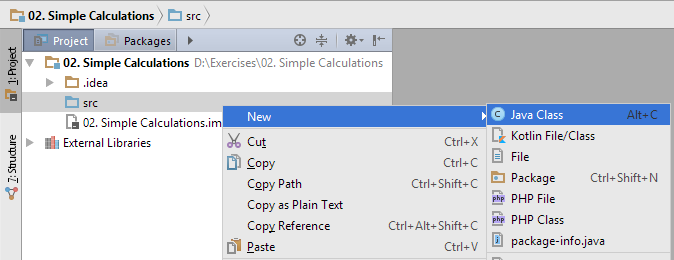


Целта на този blank project e да добавяте в него **по един клас за всяка задача** от упражненията.

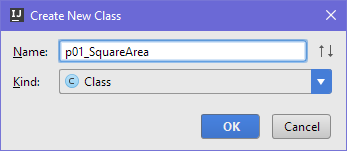
## Пресмятане на лице на квадрат

Първата задача от тази тема е следната: да се напише **конзолна програма**, която **въвежда цяло число** a и **пресмята лицето на квадрат** **със страна** a. Задачата е тривиално лесна: въвеждате число от конзолата, умножавате го само по себе си и печатате получения резултат на конзолата.

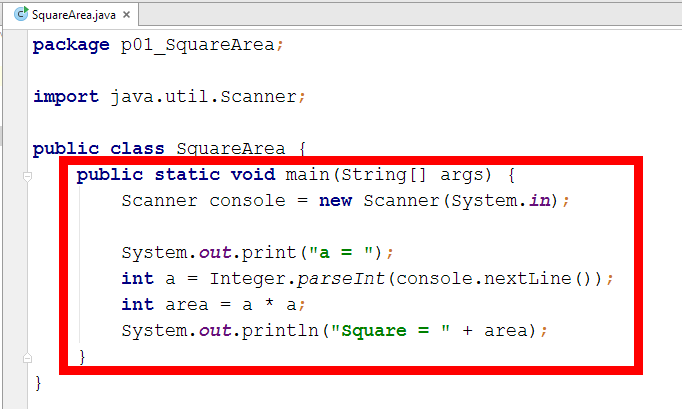
1. Създайте **нов проект** в съществуващото IntelliJ решение. В Project Explorer кликнете с десен бутон на мишката върху „**src“ папката**. Изберете [New] 🡪 [Java Class]:



1. Ще се отвори диалогов прозорец за избор на име на класа. Напишете подходящо име, например “**p01**\_SquareArea” (*Понеже име на клас не може да започва с цифра или да съдържа тирета или други специални знаци, ще сложим „p“ отпред, за да можем да създадем класа)*:
2. Изберете подходящо име на класа, например “p01\_SquareArea”:

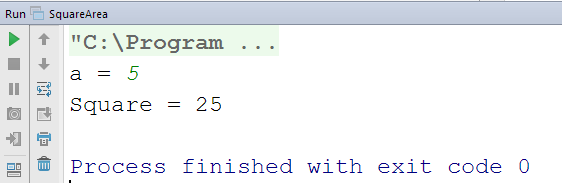


1. Вече имате проект с едно конзолно приложение в него. Остава да напишете кода за решаване на задачата.
2. Отидете в тялото на метода Main(string[] args) и напишете кода от картинката по-долу:

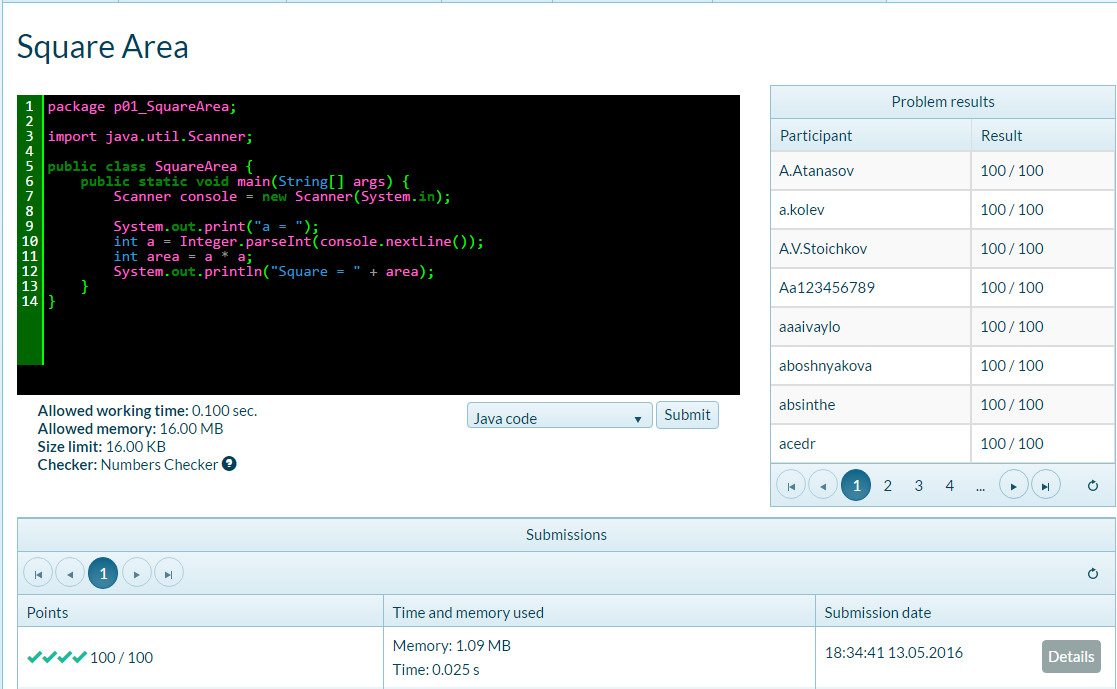


Кодът въвежда цяло число с a = Integer.parseInt(console.nextLine()), след това изчислява area = a \* a и накрая печата стойността на променливата area.

1. **Стартирайте** програмата с [Ctrl+Shift+F10] и я **тествайте** с различни входни стойности:

****

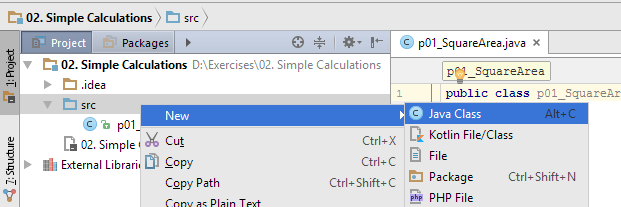
1. **Тествайте** решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#0>. Трябва да получите 100 точки (напълно коректно решение):



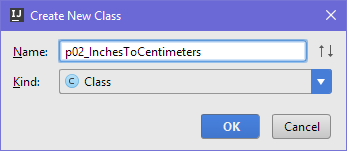
## От инчове към сантиметри

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

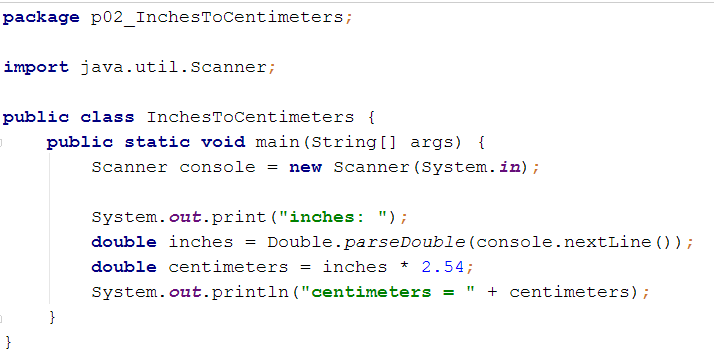
1. Първо създайте **нов Java клас** в решението “**Simple Calculations**”. Кликнете с десен бутон върху папката „src” в проекта и изберете [New] 🡪 [Java Class]:



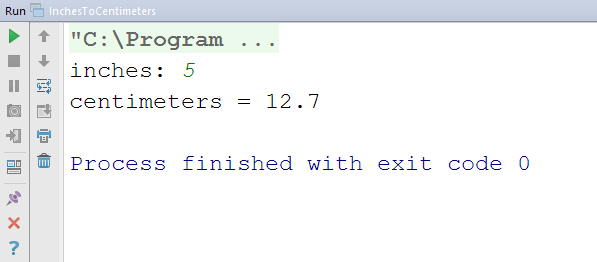
Задайте име “p02\_InchesToCentimeters”:



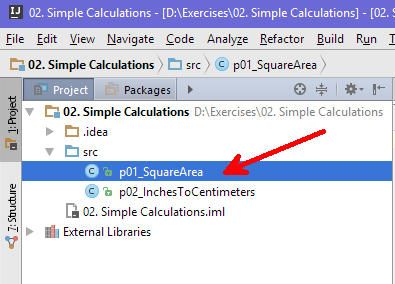
1. **Напишете кода** на програмата. Може да си помогнете с примерния код от картинката:



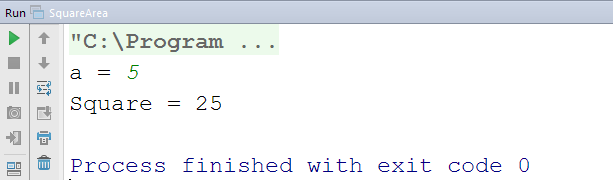
1. **Стартирайте програмата**, както обикновено с [Ctrl+Shift+F10]:



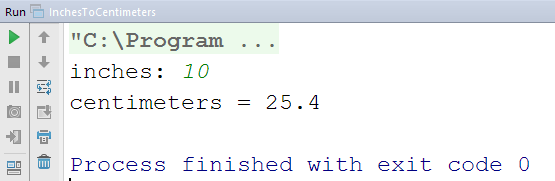
1. Сега **превключете към преходната програма** (лице на квадрат). Това става с двоен клик на мишката върху файла **p01\_**SquareArea в панела [Project] на IntelliJ:



1. Натиснете пак **[Ctrl+Shift+F10]**. Този път трябва да се стартира другият клас:

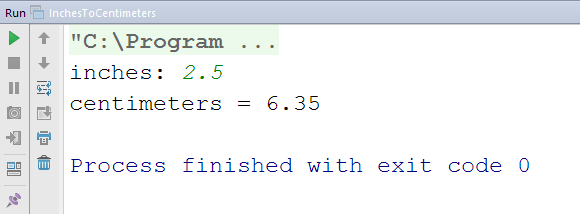


1. Превключете обратно към класа “**p02\_**InchesТoCentimeters” и го стартирайте с **[Ctrl+Shift+F10]**:



**Превключването между проектите** е много лесно, нали? Просто избираме файла със сорс кода на програмата, кликваме го два пъти с мишката и при стартиране тръгва програмата от този файл.

1. Тествайте с **дробни числа**, например с 2.5:



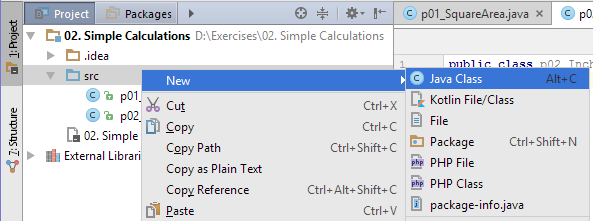
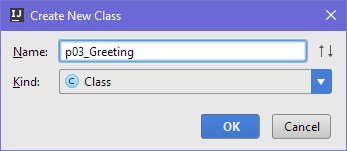
1. Вече е време за **тестване в judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#1>. Решението би трябвало да бъде прието като напълно коректно:



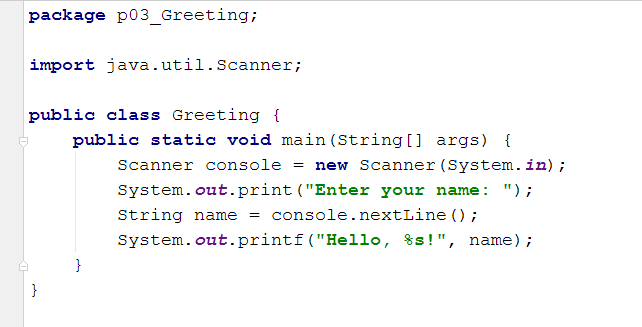
## Поздрав по име

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

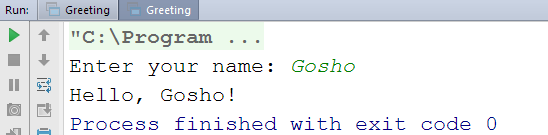
1. Първо създайте **нов Java клас** със име “**p03\_**Greeting” в решението “Simple Calculations”:

🡺 

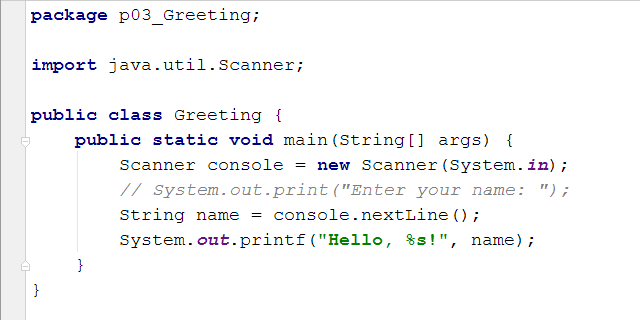
1. **Напишете кода** на програмата. Ако се затруднявате, може да ползвате примерния код по-долу:



1. **Стартирайте** програмата с **[Ctrl+Shift+F10]** и я тествайте:



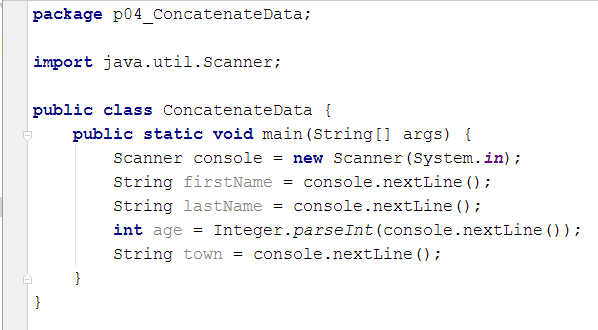
1. Тествайте в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#2>. Преди да пратите решението сложете коментар на втория ред, който печата “Enter your name”:



## Съединяване на текст и числа

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

1. Добавете към текущото IntelliJ решение още един **клас** със име “p04\_ConcatenateData”.
2. **Напишете кода**, който чете входните данни от конзолата:



1. **Допишете код**, който отпечатва описаното в условието на задачата съобщение.

D:\junk\ShareX\2016-05\2016-05-13_20-23-04.png

На горната картинка кодът е нарочно даден размазан, за да помислите как да си го напишете сами.

1. **Тествайте** решението локално с [Ctrl+Shift+F10] и въвеждане на примерни данни.
2. Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#3>.

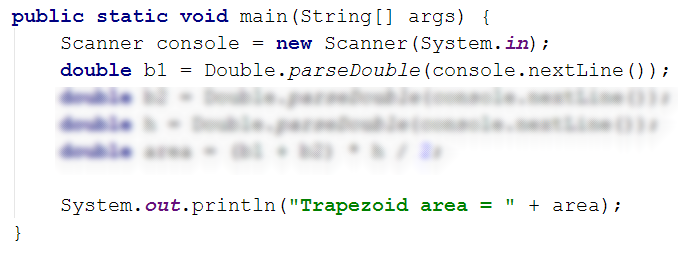
## Лице на трапец

Напишете програма, която чете от конзолата три числа b1, b2 и h и **пресмята лицето на трапец** с основи b1 и b2 и височина h. **Формулата за лице на трапец** е (b1 + b2) \* h / 2.

На фигурата по-долу е показан трапец със страни 8 и 13 и височина 7. Той има лице (8 + 13) \* 7 / 2 = 73.5.



1. Добавете към текущото IntelliJ решение още един **Java клас** със име “**p05\_**TrapezoidArea”.
2. **Напишете кода**, който чете входните данни от конзолата, пресмята лицето на трапеца и го отпечатва:



Кодът на картинката е нарочно размазан, за да си го доизмислите и допишете сами.

1. **Тествайте** решението локално с [Ctrl+Shift+F10] и въвеждане на примерни данни.
2. Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#4>.

## Периметър и лице на кръг

Напишете програма, която чете от конзолата **число** r и пресмята и отпечатва **лицето** и **периметъра на кръг** / окръжност с радиус r.

|  |  |
| --- | --- |
| **вход** | **изход** |
| 3 | Area = 28.2743338823081  Perimeter = 18.8495559215388 |
| 4.5 | Area = 63.6172512351933  Perimeter = 28.2743338823081 |

За изчисленията можете да използвате следните формули:

* area = Math.PI \* r \* r
* perimeter = 2 \* Math.PI \* r

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#5>.

## Лице на правоъгълник в равнината

**Правоъгълник** е зададен с **координатите** на два от своите срещуположни ъгъла (x1, y1) – (x2, y2). Да се пресметнат **площта** и **периметъра** му. **Входът** се чете от конзолата. Числата x1, y1, x2 и y2 са дадени по едно наред. **Изходът** се извежда на конзолата и трябва да съдържа два реда с по една число на всеки от тях – лицето и периметъра.

|  |  |
| --- | --- |
| **вход** | **изход** |
| 60  20  10  50 | 1500  160 |
| 30  40  70  -10 | 2000  180 |
| 600.25  500.75  100.50  -200.5 | 350449.6875  2402 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#6>.

## Лице на триъгълник

Напишете програма, която чете от конзолата **страна** и **височина** на **триъгълник** и пресмята неговото лице. Използвайте **формулата** за лице на триъгълник: area = a \* h / 2. Закръглете резултата до **2 знака след десетичната точка** използвайки “[**%.2f**](https://docs.oracle.com/javase/tutorial/essential/io/formatting.html)“.

|  |  |
| --- | --- |
| **вход** | **изход** |
| 20  30 | Triangle area = 300 |
| 15  35 | Triangle area = 262.5 |
| 7.75  8.45 | Triangle area = 32.74 |
| 1.23456  4.56789 | Triangle area = 2.82 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#7>.

## Конзолен конвертор: от градуси °C към градуси °F

Напишете програма, която чете **градуси по скалата на Целзий** (°C) и ги преобразува до **градуси по скалата на Фаренхайт** (°F). Потърсете в Интернет подходяща [формула](http://bfy.tw/3rGh), с която да извършите изчисленията. Закръглете резултата до **2 знака след десетичната точка**. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 25 | 77 | 0 | 32 | -5.5 | 22.1 | 32.3 | 90.14 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#8>.

## Конзолен конвертор: от радиани в градуси

Напишете програма, която чете **ъгъл в [радиани](https://bg.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B8%D0%B0%D0%BD)** (rad) и го преобразува в **[градуси](https://bg.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D0%B4%D1%83%D1%81_(%D1%8A%D0%B3%D1%8A%D0%BB))** (deg). Потърсете в Интернет подходяща формула. Числото **π** в Java програми е достъпно чрез Math.PI. Закръглете резултата до най-близкото цяло число. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 3.1416 | 180 | 6.2832 | 360 | 0.7854 | 45 | 0.5236 | 30 |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#9>.

## Конзолен конвертор: USD към BGN

Напишете програма за **конвертиране на щатски долари** (USD) **в български лева** (BGN). **Закръглете** резултата до **2 цифри** след десетичната запетая. Използвайте фиксиран **курс** между долар и лев: **1 USD** = **1.79549 BGN**.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 20 | 35.91 BGN | 100 | 179.55 BGN | 12.5 | 22.44 BGN |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#10>.

## \* Конзолен междувалутен конвертор

Напишете програма за **конвертиране на парична сума от една валута в друга**. Трябва да се поддържат следните валути: **BGN**, **USD**, **EUR**, **GBP**. Използвайте следните фиксирани валутни курсове:

|  |  |  |  |
| --- | --- | --- | --- |
| **Курс** | **USD** | **EUR** | **GBP** |
| **1 BGN** | 1.79549 | 1.95583 | 2.53405 |

**Входът** e **сума за конвертиране** + **входна валута** + **изходна валута**. **Изходът** е едно число – преобразуваната сума по посочените по-горе курсове, закръглен до **2 цифри** след десетичната точка. Примери:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |  | **вход** | **изход** |
| 20  USD  BGN | 35.91 BGN | 100  BGN  EUR | 51.13 EUR | 12.35  EUR  GBP | 9.53 GBP | 150.35  USD  EUR | 138.02 EUR |

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#11>.

## \*\* Пресмятане с дати: 1000 дни на Земята

Напишете програма, която въвежда **рождена дата** във формат “dd-MM-yyyy” и пресмята датата, на която се навършват **1000 дни** от тази рождена дата и я отпечатва в същия формат.

|  |  |
| --- | --- |
| **вход** | **изход** |
| 25-02-1995 | 20-11-1997 |
| 07-11-2003 | 02-08-2006 |
| 30-12-2002 | 24-09-2005 |
| 01-01-2012 | 26-09-2014 |
| 14-06-1980 | 10-03-1983 |

**\* Подсказки**: потърсете информация за класа LocalDate в Java и по-конкретно разгледайте методите **LocalDate.parse(String),** plusDays(count) и format(DateTimeFormatter). С тяхна помощ може да решите задачата, без да е необходимо да изчислявате дни, месеци и високосни години.

Тествайте решението си в **judge системата**: <https://judge.softuni.bg/Contests/Practice/Index/151#12>. Не печатайте нищо допълнително на конзолата освен изискваната дата!

# Изпитни задачи от минали издания на курса

## \*Учебна зала

*Първа задача от междинния изпит на 6 март 2016. Тествайте решението си* [***тук***](https://judge.softuni.bg/Contests/Practice/Index/169#0)*.*

**Учебна зала** има правоъгълен размер w на h метра, без колони във вътрешността си. Залата е разделена на две части – лява и дясна, с коридор приблизително по средата. В лявата и в дясната част има **редици с бюра**. В задната част на залата има голяма **входна врата**. В предната част на залата има **катедра** с подиум за преподавателя. Едно **работно място** заема **70 на 120 cm** (маса с размер 70 на 40 cm + място за стол и преминаване с размер 70 на 80 cm). **Коридорът** е широк поне **100 cm**. Изчислено е, че заради **входната врата** (която е с отвор 160 cm) се губи точно **1 работно място**, а заради **катедрата** (която е с размер 160 на 120 cm) се губят точно **2 работни места**. Напишете програма, която въвежда размери на учебната зала и изчислява **броя работни места в нея** при описаното разположение (вж. фигурата).

### Вход

От конзолата се четат 2 **числа**, по едно на ред: h (дължина в метри) и w (широчина в метри).

Ограничения: **3** ≤ h ≤ w ≤ **100**.

### Изход

Да се отпечата на конзолата едно цяло число: **броят места** в учебната зала.

### Примерен вход и изход

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Чертеж** | **Обяснения** |
| 15  8.9 | 129 |  | Залата е дълга 1500 cm. В тях могат да бъдат разположени **12 реда** (12 \* 120 cm = 1440 + 60 cm остатък).  Залата е широка 890 cm. От тях 100 cm отиват за коридора в средата. В останалите 790 cm могат да се разположат по **11 бюра** **на ред** (11 \* 70 cm = 770 cm + 20 cm остатък).  **Брой места** = **12 \* 11 - 3** = 132 - 3 = **129** (имаме 12 реда по 11 места = 132 минус 3 места за катедра и входна врата). |
| 8.4  5.2 | 39 |  | Залата е дълга 840 cm. В тях могат да бъдат разположени **7 реда** (7 \* 120 cm = 840, без остатък).  Залата е широка 520 cm. От тях 100 cm отиват за коридора в средата. В останалите 420 cm могат да се разположат по **6 бюра** **на ред** (6 \* 70 cm = 420 cm, без остатък).  **Брой места** = **7 \* 6 - 3** = 42 - 3 = **39** (имаме 7 реда по 6 места = 42 минус 3 места за катедра и входна врата). |

## \*Зеленчукова борса

*Първа задача от междинния изпит на 26 март 2016. Тествайте решението си* [***тук***](https://judge.softuni.bg/Contests/Practice/Index/179#0)***.***

Градинар продавал реколтата от градината си на зеленчуковата борса. Продава **зеленчуци за** **N лева на килограм** и **плодове за M лева за килограм**. Напишете програма, която да **пресмята приходите от реколтата в евро** ( ако приемем, че **едно евро** е равно на **1.94лв**).

### Вход

От конзолата се четат **4 числа**, по едно на ред:

* Първи ред – Цена за килограм зеленчуци – число с плаваща запетая
* Втори ред – Цена за килограм плодове – число с плаваща запетая
* Трети ред – Общо килограми на зеленчуците – цяло число
* Четвърти ред – Общо килограми на плодовете – цяло число

**Ограничения**: **Всички числа ще са в интервала от 0.00 до 1000.00**

### Изход

Да се отпечата на конзолата **едно число с плаваща запетая**: **приходите от всички плодове и зеленчуци в** **евро**.

### Примерен вход и изход

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 0.194  19.4  10  10 | 101 | Зелечуците струват – 0.194лв. \* 10кг. = 1.94лв.  Плодовете струват – 19.4лв. \* 10кг. = 194лв.  Общо – 195.94лв. = 101евро |
| 1.5  2.5  10  10 | 20.6185567010309 |  |