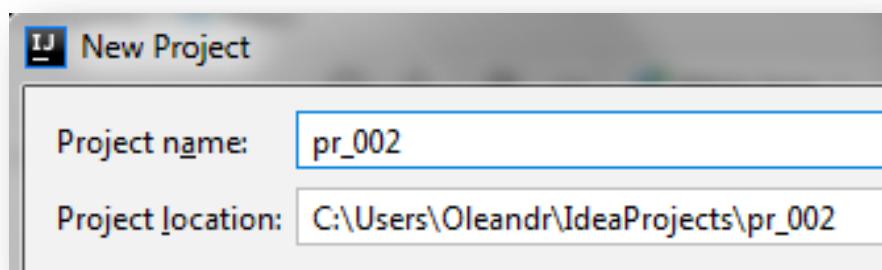


Создание второго проекта

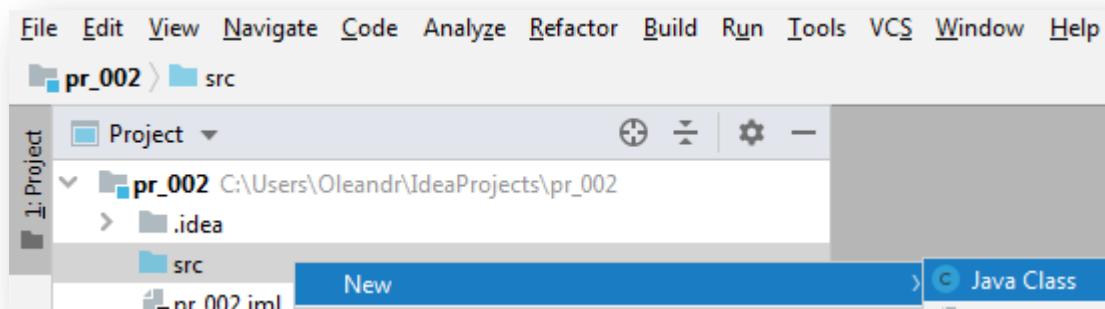
File->New->Project

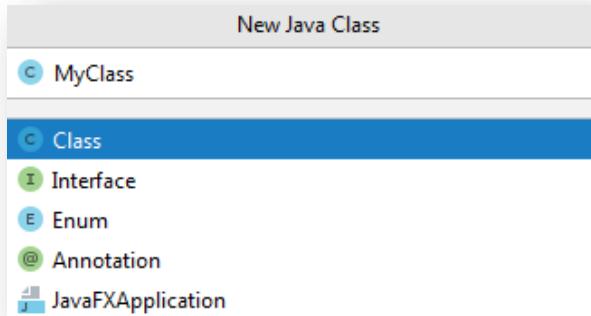
Next

Next



Finish





Наберите **psvm** и нажмите клавишу **Tab**

MyClass.java

```
1 public class MyClass {  
2     psvm  
3     psvm main() method declaration  
4 }
```

Press Ctrl+Space to see non-imported classes

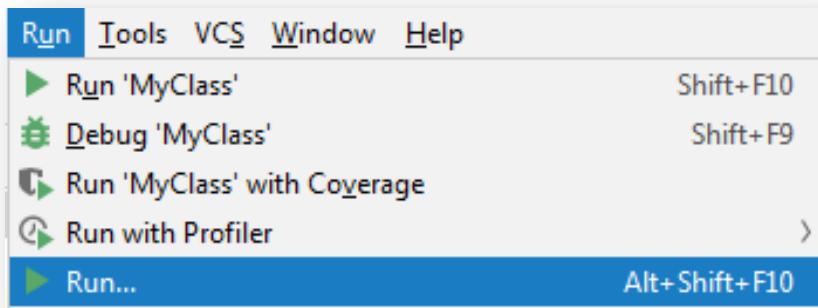
MyClass.java

```
1 ► public class MyClass {  
2 ►     □ public static void main(String[] args) {  
3 ►         □ }  
4 ►     □ }  
5  
6 }
```

Наберите **sout** и нажмите клавишу **Tab**

```
System.out.println("Цикл for. Задания");
```

Запустите программу на исполнение



Наберите **for i** и нажмите клавишу **Tab**

```
for (int i = 0; i < ; i++) {  
}
```

A screenshot of an IDE code editor showing a portion of a Java code snippet. The code consists of a single-line `for` loop. The variable `i` is underlined with a red squiggly line, and a red rectangular box highlights the letter `i` in the identifier `i`. This visual cue indicates that `i` is the current tab stop or the next character the user will type.

Цикл for. Упражнения

```
for (int i = 0; i < 5; i++) {  
    // do something  
}
```

Напишите три вида бесконечных циклов for:

```
for ( ; ; ) {  
    System.out.println("Yes/No");  
}
```

Прервите выполнение цикла с помощью **Ctr + F2**

Закомментируйте цикл **Ctrl+Shift+ /**

```
for (int i=0 ; ; i++) {  
    System.out.println(i);  
}
```

Прервите выполнение цикла с помощью **Ctr + F2**

Закомментируйте цикл **Ctrl+Shift+ /**

```
for (int i = 3; i !=2 ; i++) {  
    System.out.println(i);  
}
```

Создайте конечные циклы:

```
for (int i=5 ; i>0 ; i--) {  
    System.out.println(i);  
}
```

```
for (int i = 0; i !=8 ; i=i+2) {  
    System.out.println(i);  
}
```

```
for (int i = 10; i >=0 ; --i) {  
    System.out.println(i);  
}
```

Преобразуйте программу к следующему виду:

```
public class MyClass {  
  
    public static void first(int n) {  
        for (int i=n ; i>0 ; i--) System.out.println(i);  
    }  
  
    public static void second(int n) {  
        for (int i = 0; i !=n ; i=i+2) System.out.println(i);  
    }  
  
    public static void main(String[] args) {  
        first(2);  
        second(10);  
    }  
}
```

Вложенные классы

```
public static void first(int n)  
public static void second(int n)
```

Модификатор public

Если член класса «public», то он может быть использован в любом месте.

Модификатор static

static - переводится как "статичный", "постоянный", "неизменный". Его задача - сделать переменную или метод "независимыми" от объекта.

Условный оператор. Упражнения

Предскажите, а затем проверьте - что будет выведено после выполнения следующих фрагментов кода:

```
// фрагмент 1
int a=5;
double b=5.0;

if (a == b)
    //Если a равно b - выводим сообщение
    System.out.println("a и b равны!");
else
    //Если a не равно b - выводим сообщение
    System.out.println("a и b не равны!");

// фрагмент 2
if (a < b)
    a = 0;
System.out.println(a);

// фрагмент 3
// тернарный оператор
System.out.println(a==b ? "a и b равны!" : "a и b не равны!");

// фрагмент 4
int a=5;
double b=2;
float c= 3;
long d=1;

if (a == b)
    //Если a равно b - выводим сообщение
    System.out.println("a и b равны!");
else if (a == c)
    //Если a равно c - выводим сообщение
    System.out.println("a и c равны!");
else if (a == d)
    //Если a равно d - выводим сообщение
    System.out.println("a и d равны!");
else
    System.out.println("a не равно b, c, d");
```

Оператор выбора switch — case

Предскажите, а затем проверьте - что будет выведено после выполнения следующих фрагментов кода:

```
//фрагмент 1
int n = 2;
String trafficLights;
switch (n) {
    case 1: trafficLights = "Красный";
    break;
    case 2: trafficLights = "Желтый";
    break;
    case 3: trafficLights = "Зеленый";
    break;
    default: trafficLights = "Ошибка";
    break;
}
System.out.print(trafficLights);
```

```
//фрагмент 2
int n = 2;
String trafficLights;
switch (n) {
    case 1: trafficLights = "Красный";
    case 2: trafficLights = "Желтый";
    case 3: trafficLights = "Зеленый";
    default: trafficLights = "Ошибка";
} System.out.print(trafficLights);
```

Проверьте - что будет выведено, если последовательно присваивать следующие значения переменной `int n = 2; 6; 8`

```
int n = 3;
String number;
switch (n) {
    case 1:
    case 2:
    case 3:
        number = "между 1 и 3";
        break;
    case 4: number = "четыре";
        break;
    default: number = "меньше 1, или больше 4";
}
System.out.print(number);
```

Простые числовые типы

Тип	Разрядность	MIN	MAX
byte	8 бит	-128	127
short	16 бит	-32768	32767
int	32 бит	-2147483648	2147483647
long	64 бит	-9223372036854775808	9223372036854775807
float	32 бит	-3.4E+38	3.4E+38
double	64 бит	-1.7E+308	1.7E+308

Преобразования типов. Упражнения

Предскажите, а затем проверьте - что будет выведено после выполнения следующих фрагментов кода:

```
//фрагмент 1
// byte от -128 до 127
int n1=-129;
System.out.print((byte)n1);
```

```
//фрагмент 2
int n2 = 128;
System.out.print((byte)n2);
```

```
//фрагмент 3
// short от -32768 до 32767
int c1 = 32768;
System.out.print((short)c1);
```

```
//фрагмент 4
//int от -2147483648 до 2147483647
double f1 = 3E+9;
System.out.print((int)f1);
```

```
//фрагмент 5
//double от 1.7E-308 до 1.7E+308
double f2 = 1.7E+308;
System.out.print((float)f2);
```

```
//фрагмент 6
//double от 1.7E-308 до 1.7E+308
double f3 = -1.7E-308;
System.out.print((float)f3);
```

```
//фрагмент 7
float f4 = (float) 2.0E+20;
System.out.print((long)f4);
```

```
//фрагмент 8
double f5 = 1.987654321;
System.out.print((int)f5);
```

Некоторые математические функции

```
// random(): возвращает случайное число от 0.0 до 1.0
    System.out.println(Math.random());

// abs(double a): возвращает модуль числа a
    System.out.println(Math.abs(-7.5));

// floorDiv(int a, int b): возвращает целочисленный результат
// деления a на b
    System.out.println(Math.floorDiv(13, 2));

// round(double d): возвращает число d, округленное до ближайшего
// целого числа
    System.out.println(Math.round(3.05));

//      sqrt(double value): возвращает квадратный корень числа
// value
    System.out.println(Math.sqrt(4));

// pow(double a, double b): возвращает число a, возведенное в
// степень b
    System.out.println(Math.pow(4, 2));

//      log(double a): возвращает натуральный логарифм числа a
    System.out.println(Math.log(Math.exp(1)));

// toDegrees(double value) переводит радианы в градусы и
// toRadians(double value) - градусы в радианы
    System.out.println(Math.toDegrees(Math.PI/6));

// Радиан – угол, соответствующий дуге, длина которой равна её
// радиусу
// 57,3 градусов == 1 радиан
    System.out.println(Math.toRadians(57.3));

// sin(double value): возвращает синус угла value
    System.out.println(Math.sin(Math.PI/6));
```

Деление и целочисленное деление, остаток от деления

```
System.out.println(26/3); // 8
System.out.println(26%3); // 2
```

```
System.out.println(26.5/3); // 8.8333
System.out.println(26.5%3); // 2.5
```

Подключение библиотек Math и System

```
import static java.lang.Math.* ;
import static java.lang.System.*;

public class MyClass {

    public static void main(String[] args) {

        out.println(sqrt(16));
    }
}
```

Константы

Ключевое слово **final** означает, что присвоить данной переменной какое-нибудь значение можно лишь один раз, после чего изменить его уже нельзя.

Принято использовать в именах констант только прописные буквы.

```
import static java.lang.Math.* ;
import static java.lang.System.*;

public class MyClass {

    public static final double MY_PI = 3.14;

    public static void main(String[] args) {

        int r = 2;

        out.println(2 * MY_PI * r); // 12.56
        out.println(2 * PI * r); // 12.566370614359172
    }
}
```

Логические операции

! — оператор отрицания;

&& — оператор логическое И (сокращенный);

|| — оператор логическое ИЛИ (сокращенный);

```
System.out.println(true && true); //true
```

```
System.out.println(true || false); //true
```

```
System.out.println(!true && true); //false
```

```
System.out.println(!true || false); //false
```

Сокращенное вычисление

```
System.out.println(false && все равно, что тут будет); //false
```

```
System.out.println(true || все равно, что тут будет); //true
```

Побитовые операции

& — оператор побитовое И;

| — оператор побитовое ИЛИ;

^ — оператор побитовое исключающее ИЛИ.

~ — побитовый оператор “НЕ”

```
int x = 2;  
int y = 4;
```

```
System.out.println(x|y);
```

```
System.out.println(x&y);
```

```
System.out.println(Integer.toBinaryString(x));
```

```
System.out.println(Integer.toBinaryString(y));
```

Задания

Используйте внутренние классы для решения задачи, проверку выполняйте в классе main.

1. task001. Найти наибольшее значение из двух целочисленных переменных

```
public class MyClass {  
  
    public static int task001(int a, int b) {  
        //1. Найти наибольшее значение из двух переменных  
        return ;  
    }  
  
    public static void main(String[] args) {  
        System.out.println(task001(5,3));  
    }  
}
```

2. task002. Найти наибольшее значение из трех целочисленных переменных

```
public static int task002(int a, int b, int c) {  
    // Найти наибольшее значение из трех переменных
```

3. Если a – четное, то посчитать a*b, иначе a+b

```
public static int task003(int a, int b) {  
    //Если a - четное, то посчитать a*b, иначе a+b
```

4. Определить какой четверти принадлежит точка с целочисленными координатами (x,y)

```
public static int task004(int x, int y) {  
    // Определить какой четверти принадлежит точка с координатами (x,y)  
  
    System.out.println(task004( 1, 1));  
    System.out.println(task004(-1, 1));  
    System.out.println(task004(-1, -1));  
    System.out.println(task004( 1, -1));
```

5. Найти сумму только положительных из трех целых чисел

```
public static int task005(int a, int b, int c) {  
    // Найти сумму только положительных из трех целых чисел  
    System.out.println(task005(1,-2,6));
```

6. Есть два конверта со сторонами (a,b) и (c,d) определить, можно ли один конверт вложить в другой.

```
public static void task006(int a, int b, int c, int d) {  
    // Есть два конверта со сторонами (a,b) и (c,d) определить,  
    // можно ли один конверт вложить в другой.
```

```
task006(1,2,3,4);  
task006(4,3,2,1);
```

```
task006(1,5,3,4);  
task006(4,1,2,1);
```

7. Написать программу определения оценки студента по его рейтингу, на основе следующих правил:

Рейтинг	Оценка
0-19	F
20-39	E
40-59	D
60-74	C
75-89	B
90-100	A

```
public static String task007(int n) {  
    /*      Написать программу определения оценки студента по его  
    // рейтингу.  
        0-19    F  
        20-39   E  
        40-59   D  
        60-74   C  
        75-89   B  
        90-100  A  
 */
```

```
System.out.println(task007(97));  
System.out.println(task007(-73));
```

8. Вывести шахматную доску с заданными размерами n и m, по принципу:

```
* * *  
* *  
* * *  
* *
```

```
public static void task008(int n, int m) {
```

9. Вывести на экран квадрат, со стороной a

```
*****  
* * *  
* * *  
* * *  
*****
```

```
public static void task009(int a) {
```

10. Проверить простое ли число? (число называется простым, если оно делится только само на себя и на 1)

```
public static boolean task010(int n)
```

11. Вывести все простые числа в диапазоне от 2 до n

```
public static void task011(int n) {
```

12. Вывести все делители числа

```
public static void task012(int n) {
```

13. Посчитать сумму цифр заданного числа

```
public static int task013(int n) {
```

14. Вывести число, которое является зеркальным отображением последовательности цифр заданного числа, например, задано число 123, вывести 321.

15. Вывести все натуральные числа, квадрат которых больше меньше заданного числа n.