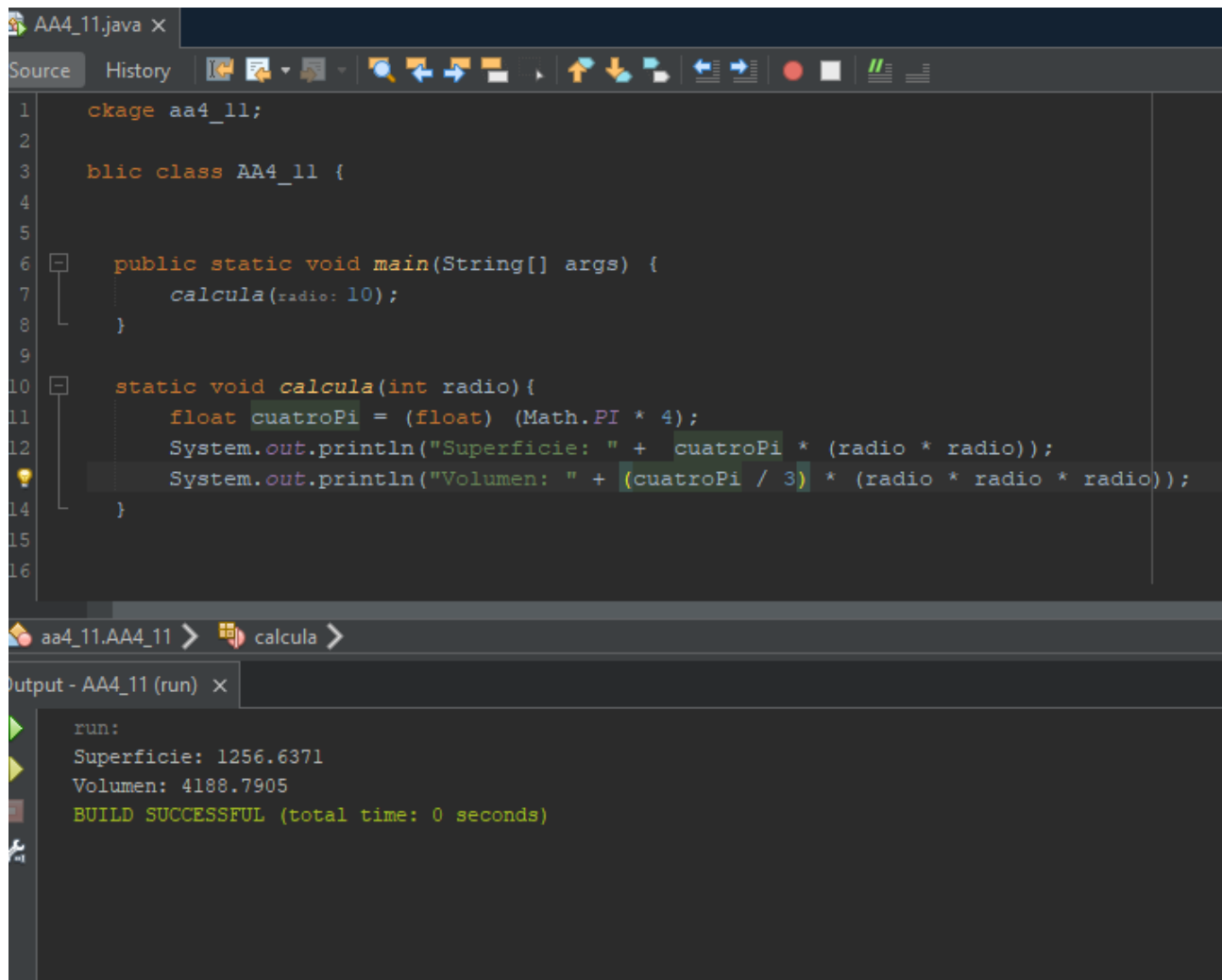


4.11. Diseña una función que calcule y muestre la superficie y el volumen de una esfera.

$$\text{Superficie} = 4\pi \cdot \text{radio}^2$$

$$\text{Volumen} = \frac{4\pi}{3} \cdot \text{radio}^3$$



The screenshot shows an IDE with a Java file named AA4_11.java. The code defines a class AA4_11 with a main method and a static method calcula. The main method calls calcula with a radius of 10. The calcula method calculates the surface area and volume using the formulas provided in the problem statement. The output window shows the results: Superficie: 1256.6371 and Volumen: 4188.7905. The build was successful.

```
1 package aa4_11;
2
3 public class AA4_11 {
4
5
6     public static void main(String[] args) {
7         calcula(10);
8     }
9
10    static void calcula(int radio) {
11        float cuatroPi = (float) (Math.PI * 4);
12        System.out.println("Superficie: " + cuatroPi * (radio * radio));
13        System.out.println("Volumen: " + (cuatroPi / 3) * (radio * radio * radio));
14    }
15
16 }
```

aa4_11.AA4_11 > calcula >

Output - AA4_11 (run) x

```
run:
Superficie: 1256.6371
Volumen: 4188.7905
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.12 Implementa la función

`static double distancia(double x1, double y1, double x2, double y2)`
que calcula y devuelve la distancia euclídea que separa los puntos (x1, y1) y (x2, y2). La fórmula para calcular esta distancia es:

$$distancia = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

```
1 package aa4_12;
2
3 public class Aa4_12 {
4
5     public static void main(String[] args) {
6         System.out.println("Distancia: " + distancia(x1: 5.5d, y1: 9.9d, x2: 3.3d, y2: 7.7d));
7     }
8
9     static double distancia(double x1, double y1, double x2, double y2){
10         return Math.sqrt( Math.pow((x1 - x2), 2) + Math.pow((y1 - y2), 2));
11     }
12 }
13
```

aa4_12.Aa4_12 > main >

Output - Aa4_12 (run) x

```
run:
Distancia: 3.111269837220809
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.13. Crea la función `muestraPares(int n)` que muestre por consola los primeros `n` números pares.

```
1 package aa4_13;
2
3 public class Aa4_13 {
4
5     public static void main(String[] args) {
6         muestraPares(n: 5);
7     }
8
9     static void muestraPares(int n){
10         System.out.println("Primeros " + n + " numeros pares:");
11         int contador = 0;
12         for (int i=0; contador < n ; i++){
13             if( i % 2 == 0){
14                 System.out.println((contador + 1) + "# numero par: " + i);
15                 contador++;
16             }
17         }
18     }
19 }
```

aa4_13.Aa4_13 > muestraPares > for (int i = 0; contador < n; i++) > if (i % 2 == 0) >

Output - Aa4_13 (run) x

```
run:
Primeros 5 numeros pares:
1# numero par: 0
2# numero par: 2
3# numero par: 4
4# numero par: 6
5# numero par: 8
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.14. Escribe una función a la que se pase como parámetros de entrada una cantidad de días, horas y minutos. La función calculará y devolverá el número de segundos que existen en los datos de entrada.

```
1 package aa4_14;
2
3 public class Aa4_14 {
4
5     public static void main(String[] args) {
6         System.out.println("Segundos totales: " + segundos(dias: 5, horas: 10, min: 12));
7     }
8
9     static double segundos(int dias, int horas, int min){
10         return (dias * 24 * 60 * 60) + (horas * 60 * 60) + (min * 60);
11     }
12
13 }
14
```

aa4_14.Aa4_14 > segundos >

Output - Aa4_14 (run) x

```
run:
Segundos totales: 468720.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.15. Diseña una función a la que se le pasan las horas y minutos de dos instantes de tiempo, con el siguiente prototipo:

```
static int diferenciaMin(int horal, int minutol, int hora2, int minuto2)
```

La función devolverá la cantidad de minutos que existen de diferencia entre los dos instantes utilizados.

```
1 package aa4_15;
2
3 public class Aa4_15 {
4
5     public static void main(String[] args) {
6         System.out.println("Minutos de diferencia: " + diferenciaMin(horal: 10, min1: 25, hora2: 11, min2: 50));
7     }
8
9     static int diferenciaMin(int horal, int min1, int hora2, int min2){
10         int horas = (horal > hora2) ? (horal - hora2) * 60 : (hora2 - horal) * 60;
11         int minutos = 0;
12         if(horal > hora2) {
13             if(min1 > min2){
14                 minutos = min1 - min2;
15             }else if(min1 < min2){
16                 minutos = 60 - min2 + min1;
17             }
18         }else{
19             if (min2 > min1){
20                 minutos = min2 - min1;
21             }else if(min2 < min1){
22                 minutos = 60 - min1 + min2;
23             }
24         }
25         return horas + minutos;
26     }
27 }
28
29
```

aa4_15.Aa4_15 > diferenciaMin >

Output - Aa4_15 (run) X

```
run:
Minutos de diferencia: 85
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
1 package aa4_15;
2
3 public class Aa4_15 {
4
5     public static void main(String[] args) {
6         System.out.println("Minutos de diferencia: " + diferenciaMin(horal: 15, min1: 25, hora2: 11, min2: 50));
7     }
8
9     static int diferenciaMin(int horal, int min1, int hora2, int min2){
10         int horas = (horal > hora2) ? (horal - hora2) * 60 : (hora2 - horal) * 60;
11         int minutos = 0;
12         if(horal > hora2) {
13             if(min1 > min2){
14                 minutos = min1 - min2;
15             }else if(min1 < min2){
16                 minutos = 60 - min2 + min1;
17             }
18         }else{
19             if (min2 > min1){
20                 minutos = min2 - min1;
21             }else if(min2 < min1){
22                 minutos = 60 - min1 + min2;
23             }
24         }
25         return horas + minutos;
26     }
27 }
28
29
```

aa4_15.Aa4_15 > main >

Output - Aa4_15 (run) X

```
run:
Minutos de diferencia: 275
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.16. Implementa la función `divisoresPrimos()` que muestra, por consola, todos los divisores primos del número que se le pasa como parámetro.

Entiendo que se refiere a obtener los números que son divisores del número pasado y que además son en si mismos número primos.

```
1 package com.mycompany.aa4_16;
2
3 public class Aa4_16 {
4
5     public static void main(String[] args) {
6         divisoresPrimos(num: 500);
7     }
8
9     static void divisoresPrimos(int num) {
10         int divisor = 2;
11         if (num < 2) {
12             System.out.println("El numero " + num + "no tiene divisores primos");
13         } else {
14             while (divisor < num) {
15                 if (num % divisor == 0) {
16                     if (esDivisorPrimo(num: divisor)) {
17                         System.out.println("El número " + divisor +
18                             " es divisor de " + num + " y ademas es primo.");
19                     }
20                 }
21                 divisor++;
22             }
23         }
24     }
25     static boolean esDivisorPrimo(int num){
26         boolean primo = true;
27         int i = 2;
28         if(num < 2){
29             primo = false;
30         }
31         while (i < num && primo == true) {
32             if (num % i == 0){
33                 primo = false;
34             }
35             i++;
36         }
37         return primo;
38     }
39 }
40
```

```
--- exec:3.1.0:exec (default-cli) @ Aa4_16 ---
El número 2 es divisor de 500 y ademas es primo.
El número 5 es divisor de 500 y ademas es primo.
-----
BUILD SUCCESS
-----
```

4.17. Escribe una función que decida si dos números enteros positivos son amigos. Dos números a y b son amigos si la suma de los divisores propios (distintos de él mismo) de a es igual a b . Y viceversa.
Para probar se pueden usar los números 220 y 284, que son amigos.

```
1 package com.mycompany.aa4_17;
2
3 public class Aa4_17 {
4
5     public static void main(String[] args) {
6         int a = 220;
7         int b = 284;
8         String respuesta = sonAmigos(a, b) ? " SI" : " NO";
9         System.out.println("Los numeros " + a + " y " + b
10             + respuesta + " son amigos");
11     }
12
13     static boolean sonAmigos(int a, int b) {
14         int sumaDivisores = 0;
15         int divisor = 1;
16
17         while (divisor < a) {
18             if (a % divisor == 0) {
19                 sumaDivisores += divisor;
20             }
21             divisor++;
22         }
23         return sumaDivisores == b;
24     }
25 }
```

Output X

Debugger Console X Run (Aa4_17) X

Compiling 1 source file with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Aa4_17 ---

Los numeros 220 y 284 SI son amigos

BUILD SUCCESS

Con los números $a= 350$ y $b = 600$:

--- exec:3.1.0:exec (default-cli) @ Aa4_17 ---

Los numeros 350 y 600 NO son amigos

BUILD SUCCESS

4.18. Crea una función que muestre por consola una serie de números aleatorios enteros. Los parámetros de la función serán: la cantidad de números aleatorios que se mostrarán y los valores mínimos y máximos que estos pueden tomar.

```
5
6 public class Aa4_18 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10        System.out.println(x: "Introduzca cuantos numeros aleatorios quiere: ");
11        int cantidad = scanner.nextInt();
12        System.out.println(x: "Introduzca valor mínimo de los numeros aleatorios: ");
13        int min = scanner.nextInt();
14        System.out.println(x: "Introduzca valor máximo de los numeros aleatorios: ");
15        int max = scanner.nextInt();
16
17        imprimirNumeros(cantidad, min, max);
18    }
19
20    static void imprimirNumeros(int cantidad, int min, int max) {
21        Random rand = new Random();
22        for (int i = 0; i < cantidad; i++) {
23            int numeroAleatorio = rand.nextInt((max - min) + 1) + min;
24            System.out.println("Número aleatorio: " + numeroAleatorio);
25        }
26    }
27 }
```

Output - Run (Aa4_18) ×

```
--- exec:3.1.0:exec (default-cli) @ Aa4_18 ---
Introduzca cuantos numeros aleatorios quiere:
6
Introduzca valor mínimo de los numeros aleatorios:
100
Introduzca valor máximo de los numeros aleatorios:
900
Número aleatorio: 833
Número aleatorio: 116
Número aleatorio: 761
Número aleatorio: 685
Número aleatorio: 293
Número aleatorio: 375
-----
BUILD SUCCESS
-----
```


4.19. Sobrecarga la función realizada en la Actividad de aplicación 4.18 para que el único parámetro sea la cantidad de números aleatorios que se muestra por consola. Los números aleatorios serán reales y estarán comprendidos entre 0 y 1.

```
1 package com.mycompany.aa4_19;
2
3 import java.util.Random;
4 import java.util.Scanner;
5
6 public class Aa4_19 {
7
8     public static void main(String[] args) {
9         Scanner scanner = new Scanner(System.in);
10        System.out.println(x: "Introduzca cuantos numeros aleatorios quiere: ");
11        int cantidad = scanner.nextInt();
12        imprimirNumeros(cantidad);
13    }
14
15    static void imprimirNumeros(int cantidad) {
16        for (int i = 0; i < cantidad; i++) {
17            double numeroAleatorio = Math.random();
18            System.out.println("Número aleatorio: " + numeroAleatorio);
19        }
20    }
21
22    static void imprimirNumeros(int cantidad, int min, int max) {
23        Random rand = new Random();
24        for (int i = 0; i < cantidad; i++) {
25            int numeroAleatorio = rand.nextInt((max - min) + 1) + min;
26            System.out.println("Número aleatorio: " + numeroAleatorio);
27        }
28    }
29 }
30
```

Introduzca cuantos numeros aleatorios quiere:

10

Número aleatorio: 0.7792266912305514

Número aleatorio: 0.13168843014891884

Número aleatorio: 0.11209444619948783

Número aleatorio: 0.3066494082718588

Número aleatorio: 0.6418141823840785

Número aleatorio: 0.5515492991387984

Número aleatorio: 0.7521650752059195

Número aleatorio: 0.7392043266980428

Número aleatorio: 0.5728902332994732

Número aleatorio: 0.9768557318783179

BUILD SUCCESS