1) Scrivere un semplice programma (una classe con il metodo main) che svolga le seguenti operazioni aritmetiche correttamente, scegliendo accuratamente i tipi di dati da utilizzare per immagazzinare i risultati di esse.
   - Una divisione (usare il simbolo /) tra due interi a = 5, e b = 3. Immagazzinare il risultato in una variabile r1, scegliendo il tipo di dato adeguato.
   - Una moltiplicazione (usare il simbolo *) tra un char c = 'a', ed uno short s = 5000. Immagazzinare il risultato in una variabile r2, scegliendo il tipo di dato adeguato.
   - Una somma (usare il simbolo +) tra un int i = 6 ed un float f = 3.14F. Immagazzinare il risultato in una variabile r3, scegliendo il tipo di dato adeguato.
   - Una sottrazione (usare il simbolo -) tra r1, r2 e r3. Immagazzinare il risultato in una variabile r4, scegliendo il tipo di dato adeguato.
   - Verificare la correttezza delle operazioni stampandone i risultati parziali ed il risultato finale. Tenere presente la promozione automatica nelle espressioni e utilizzare il casting propriamente.

2) All'interno del package it.unipa.prg.es02 creare la seguente classe:

```
public class NumeroIntero {
        public int numeroIntero;

        public void stampaNumero() {
                System.out.println(numeroIntero);
        }
}
```

   Questa classe definisce il concetto di numero intero come oggetto. In essa vengono dichiarati una variabile intera ed un metodo che stamperà la variabile stessa.
   Scrivere, compilare ed eseguire nello stesso package una nuova classe (contenente ovviamente un metodo main()) che:
   - istanzierà almeno due oggetti dalla classe NumeroIntero;
   - cambierà il valore delle relative variabili d'istanza e testerà la veridicità delle avvenute assegnazioni, sfruttando il metodo stampaNumero();
   - aggiungerà un costruttore alla classe NumeroIntero che inizializzi la variabile d'istanza.
   - Se istanziamo un oggetto della classe NumeroIntero, senza assegnare un nuovo valore alla variabile numeroIntero, quanto varrà quest'ultima?

3) Creare una classe Quadrato che dichiari una variabile d'istanza intera lato. Creare un metodo pubblico che si chiami perimetro() che ritorni il perimetro del quadrato, e un metodo pubblico area() che ritorni l'area del quadrato.
   - Creare una classe TestQuadrato contenente un metodo main() che istanzi un oggetto di tipo *Quadrato*, con lato di valore 5 (con una istruzione simile alla seguente: nomeOggetto.lato = 5;). Stampare poi il perimetro e l'area dell'oggetto appena creato.
   - Si crei un costruttore nella classe Quadrato che riceve in input il lato del quadrato. Fatto questo si compili la classe Quadrato.
   - Ricompilare la classe TestQuadrato e interpretare l'errore.
   - Modificare il codice della classe TestQuadrato in modo tale che compili e sia eseguita correttamente.

4) Creare una classe Rettangolo equivalente alla classe Quadrato. Prima di codificare la classe decidere che specifiche deve avere questa classe (variabili e metodi).
   - Si crei una classe TestRettangolo contenente un metodo main() che testi la classe Rettangolo, equivalentemente a come fatto nell'esercizio precedente. Istanziare almeno due rettangoli diversi.

5) Aggiungere commenti javadoc alla classe Rettangolo appena creata descrivendo brevemente lo scopo della classe, le variabili d'istanza usate, i costruttori e i metodi.
   Usate quindi l'istruzione ***javadoc -d docs Rettangolo.java*** per generare la documentazione in HTML e apritela nel browser il file *index.html* per verificare il risultato.

6) Analogamente a quanto fatto per la classe Rettangolo aggiungere commenti javadoc alla classe Quadrato. Usate quindi l'istruzione ***javadoc -d docs Quadrato.java*** per generare la documentazione in HTML. Siete soddisfatti del risultato? Riprovate con l'istruzione ***javadoc -d docs *.java***

7) Scrivere una classe SommaVettori dotata di un metodo main che: generi attraverso la Math.random() (restituisce un double fra 0.0 e 1.0) due array di double compresi fra 0 e 1000, vettore1 e vettore2, di dimensione 10; utilizzi un array di interi sommaVettori per memorizzare la somma di vettore1 e vettore2; stampi il risultato della somma.

8) Scrivere una classe Calcolatrice che contenga al suo interno i seguenti metodi:
   a. double[ ] generaVettore(int L) che genera un array di double casuali fra 0 e 1000 di dimensione L, con L parametro in input al metodo;
   b. int[ ] sommaVettori(int[ ] a, int[ ] b) che riceve in input due array di double e, una volta controllato che le dimensioni dei due array siano uguali, restituisca un array che contiene la somma dei vettori in input. Se le dimensioni di a e b sono diverse restituisce un vettore uguale ad a;
   c. int[ ] concatenaVettori(int[ ] a, int[ ] b) che riceve in input due array di double e restituisca un array che contiene la concatenazione dei vettori in input;
   d. void stampaVettore(int[ ] a) che riceve in input un vettore di double e lo stampa.

Utilizzare una classe TestMatematica con un metodo main che:
- istanzi un oggetto miaCalcolatrice della classe Calcolatrice;
- utilizzi l'oggetto miaCalcolatrice per generare 3 vettori: vettore1 e vettore2 di dimensione 3 e vettore3 di dimensione 5.
- utilizzando i metodi dell'oggetto miaCalcolatrice, sommi vettore1 e vettore2, sommi vettore1 e vettore3, concateni vettore1 e vettore3 e stampi i risultati ottenuti.

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:

javac -d . nomePackage.nomeClasse.java	*compila e genera il bytecode*
java nomePackage.nomeClasse	*esegue il bytecode sulla JVM*