Esercizio

- Progettare un metodo che estragga il carattere centrale da una stringa fornita in input
- Se la stringa ha un numero pari di caratteri, estrarre i due caratteri centrali (es. da “magico” deve estrarre “gi”)

• Fase 1: progettare l’interfaccia pubblica

```
public String estraiCarattereCentrale(String s)
{

}
```

• Fase 2: individuare la condizione per la diramazione:
  - La lunghezza della stringa è dispari?
  - s.length() % 2 == 1?

Esercizio

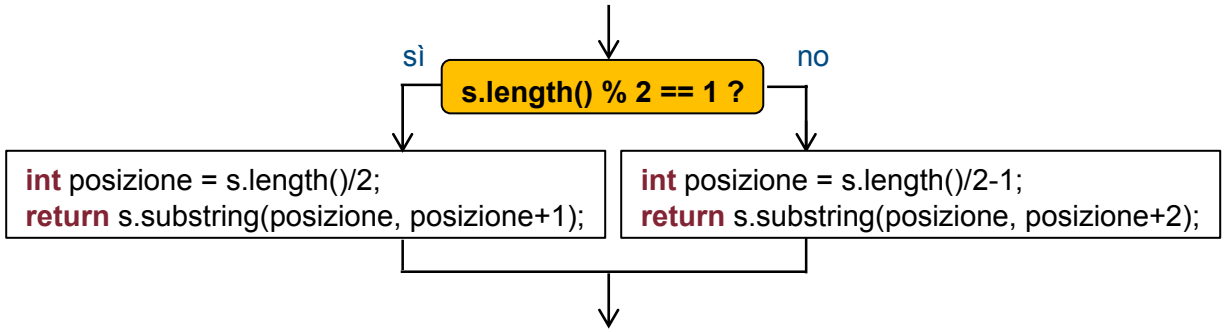
- Fase 3: progettare il codice (o lo pseudocodice) per le istruzioni da eseguire quando la condizione è soddisfatta

```
int posizione = s.length()/2;
return s.substring(posizione, posizione+1);
```

- Fase 4: progettare il codice (o lo pseudocodice) per le istruzioni
- da eseguire quando la condizione **NON** è soddisfatta

```
int posizione = s.length()/2-1;
return s.substring(posizione, posizione+2);
```

- Progetta il flusso di controllo:



Esercizio

- Una prima versione:

```
public String estraiCarattereCentrale(String s)
{
    if (s.length() % 2 == 1)
    {
        int posizione = s.length()/2;
        return s.substring(posizione, posizione+1);
    }
    else
    {
        int posizione = s.length()/2-1;
        return s.substring(posizione, posizione+2);
    }
}
```

Esercizio

- Una prima versione:

```
public String estraiCarattereCentrale(String s)
{
    if (s.length() % 2 == 1)
    {
        int posizione = s.length()/2;
        return s.substring(posizione, posizione+1);
    }
    else
    {
        int posizione = s.length()/2-1;
        return s.substring(posizione, posizione+2);
    }
}
```

- Fase 4: elimina le ripetizioni

```
public String estraiCarattereCentrale(String s)
{
    int posizione;
    int lunghezza;

    if (s.length() % 2 == 1)
    {
        posizione = s.length()/2;
        lunghezza = 1;
    }
    else
    {
        posizione = s.length()/2-1;
        lunghezza = 2;
    }

    return s.substring(posizione, posizione+lunghezza);
}
```

Esercizio: Saluto casuale

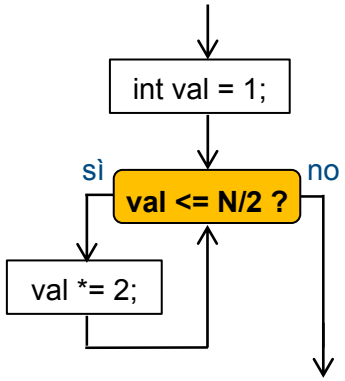
- Progettare un metodo che emetta sullo standard output un saluto scelto casualmente tra “ciao”, “hello”, “bella”, “salve” e “buongiorno” (rendere quest’ultimo doppiamente più probabile)

```
import java.util.Random;

public class SalutoCasuale
{
    public void sayHello()
    {
        String hello = null;

        // new Random().nextInt(6 equivalente a (int)(Math.random()*6)
        switch(new Random().nextInt(6))
        {
            case 0:     hello = "ciao"; break;          // informale
            case 1:     hello = "hello"; break;         // inglese
            case 2:     hello = "bella"; break;         // romanesco
            case 3:     hello = "salve"; break;         // salute a te
            default:    hello = "buongiorno"; break;    // formale
        }

        System.out.println(hello);
    }

    public static void main(String[] args)
    {
        new SalutoCasuale().sayHello();
    }
}
```

caso di default

Esercizio: la più grande potenza di 2 <= N

- Scrivere le istruzioni per calcolare la più grande potenza di 2 che sia <= un intero positivo N




```
// piu' grande potenza di 2 <= N
int val = 1;
while (val <= N/2) val *= 2;
```

Esercizio: for annidati

- Scrivere un metodo che, dato un intero N, stampi una matrice NxN il cui elemento (i, j) vale:
  - 1 se i è un divisore di j (o viceversa);
  - 0 altrimenti.

Variabili visibili solo all'interno del ciclo for

```
public void printMatrix(final int N)
{
    for (int i = 1; i <= N; i++)
    {
        for (int j = 1; j <= N; j++)
        {
            if ((i % j == 0) || (j % i == 0))
                System.out.print("1 ");
            else
                System.out.print("0 ");
        }

        System.out.println();
    }
}
```

printMatrix(10)

```
1 1 1 1 1 1 1 1 1 1
1 1 0 1 0 1 0 1 0 1
1 0 1 0 0 1 0 0 1 0
1 1 0 1 0 0 0 1 0 0
1 0 0 0 1 0 0 0 0 1
1 1 1 0 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0
1 1 0 1 0 0 0 1 0 0
1 0 1 0 0 0 0 0 1 0
1 1 0 0 1 0 0 0 0 1
```

Esercizio: ContaParola

- Scrivere un metodo che, presi in ingresso un testo sotto forma di stringa e una parola w, trasformi il testo in parole (token) e conti le occorrenze di w nel testo

Esercizio: StringaVerticale

- Scrivere un metodo che legge una stringa da console (ovvero da input args) e la stampa in verticale un carattere per linea
- Ad esempio, dato in input “ciao”, viene stampato:

```
c
i
a
o
```

Esercizio: StringheVerticali

- Scrivere un metodo che riceve tre stringhe e le stampa in verticale una accanto all'altra
- Ad esempio: date "ciao", "buondì", "hello", stampa:

```
cbh
iue
aol
onl
 do
 ì
```

Esercizio: ContaVocali

- Scrivere un metodo che riceve una stringa e stampa a video il conteggio delle vocali in essa contenute
- Ad esempio: data la stringa “le aiuole sono pulite”, il metodo stampa:

```
a=1 e=3 i=2 o=3 u=2
```

Esercizio: Divisori

- Scrivere un metodo che, dato un intero positivo n in ingresso, stampi i divisori propri di n (ovvero i divisori < n)
- Ad esempio, dato l'intero 20, il metodo stampa:

```
1, 2, 4, 5, 10
```

Esercizio: SommaNumeriPrecedenti

- Scrivere un metodo che, dati in ingresso due interi a e b e un terzo intero N, stampi a e b e gli N numeri della sequenza in cui ogni numero è la somma dei due precedenti
- Ad esempio, dati gli interi 2, 3 e 6, il metodo stampa:

```
2, 3, 5, 8, 13, 21, 34, 55
```

Esercizio: ConversioneNumeri

- Progettare una classe per la conversione di base dei numeri interi
- Ogni oggetto della classe viene costruito con un intero o con una stringa che contiene l'intero
- La classe è in grado di convertire l'intero nella base desiderata (restituito sotto forma di stringa)

Esercizio: FrasePalindroma

- Una stringa è palindroma se rimane uguale quando viene letta da sinistra verso destra o da destra verso sinistra
- Scrivere un metodo che dica che una stringa è palindroma
- Scrivere anche una classe di collaudo che testi il metodo in diverse situazioni
- Ad esempio, data in ingresso le stringhe “angelalavalalegna” o “itopinonavevanonipoti”, il metodo deve segnalare che la stringa è palindroma

Esercizio: Cornice

- Scrivere un metodo che, dato un intero N in ingresso, stampi una cornice NxN
- Ad esempio: dato l'intero 5 in ingresso il metodo stampa:

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

Esercizio: CorniceAvanzata

- Scrivere un metodo che, dato un intero N e una stringa in ingresso, stampi una cornice NxN all'interno della quale venga visualizzata la stringa (eventualmente suddivisa per righe)
- Ad esempio: dato l'intero 6 e la stringa "Cornici in Java" in ingresso il metodo stampa:

```
******
*Corn*
*ici *
*in J*
*ava *
******
```

Esercizio: TernePitagoriche

- Una terna pitagorica è una tripla di numeri interi a, b, c tali che
  1 <= a <= b <= c e $a^2+b^2 = c^2$
  - Ovvero a e b sono i lati di un triangolo rettangolo e c l'ipotenusa
- Scrivere un metodo che legge un intero N e stampa tutte le triple pitagoriche con c <= N
- Ad esempio: dato N=15 il metodo stampa:

a=3 b=4 c=5
a=6 b=8 c=10
a=5 b=12 c=13
a=9 b=12 c=15

Esercizio: StampaTriangoli

- Scrivere un metodo che, dato un intero positivo dispari N>1, stampi un triangolo isoscele la cui base è costituita da N caratteri e l'altezza da N-2
- Ad esempio, dato l'intero 5, il metodo stampa:

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

Esercizio: la classe RettangoloDiCaratteri

- Progettare una classe RettangoloDiCaratteri che rappresenta un rettangolo riempito con caratteri *
- Un oggetto della classe viene costruito fornendo la posizione x, y e la lunghezza e altezza del rettangolo
- Il metodo draw si occupa di stampare il rettangolo a video, partendo dalla posizione (0,0)
- Ad esempio, dato il rettangolo (2, 2, 4, 3), il metodo draw() stampa (␣ rappresenta una spazio):

```
␣
␣
␣****
␣****
␣****
```

Esercizio: ampliare la classe RettangoloDiCaratteri

- Aggiungere alla classe RettangoloDiCaratteri i seguenti metodi:

– setCarattere(): Permette di specificare il carattere da utilizzare per stampare i rettangoli

– drawVerticalStripes(): stampa il rettangolo a strisce verticali usando anche un secondo carattere, ad esempio:

```
*$*$

*$*$

*$*$
```

– drawHorizontalStripes(): stampa il rettangolo a strisce orizzontali

– drawChessboard(): stampa il rettangolo a mo' di scacchiera:

```
*$*$

$*$*

*$*$
```

– Una seconda versione di setCarattere() che permetta di specificare entrambi i caratteri da utilizzare per la stampa

– Un metodo che permetta di modificare la posizione del rettangolo

– Un metodo che permetta di accedere ai due caratteri usati per la stampa

Esercizio: calcolo di media, mediana e moda

- Progettare una classe Sequenza i cui oggetti si costruiscono con un array di interi
- La classe espone i seguenti metodi:
  - getMediana() che restituisce l'elemnto centrale dell'array ordinato (si utilizzi Arrays.sort per ordinare l'array e Arrays.copyOf per effettuarne una copia)
  - getMedia() che restituisce la media degli elementi dell'array
  - getModa() che restituisce il valore più frequente nella sequenza

Esercizio: da numeri romani a numeri interi

- Progettare una classe NumeroRomano i cui oggetti si costruiscono con una stringa contenente un numero romano
  - M = 1000, D = 500, C = 100, L = 50, X = 10, V = 5, I = 1
- La classe espone il metodo toInteger() che restituisce il valore intero corrispondente
- Ad esempio, new

  NumeroRomano(“MMXIX”).toInteger() restituisce 2019

Esercizio: stampa di un array

- Scrivere un metodo che, dato un array di stringhe, ne stampi i valori in sequenza
- Ad esempio, dato l'array { "son", "buoni", "almeno?", "assaggi", "il", "vino" } stampi: ["son", "buoni", "almeno", "assaggi", "il", "vino"]

```
public void stampaArray(String[] a)
{
    System.out.print("[ ");

    for (int k = 0; k < a.length; k++)
        System.out.print("\"" + a[k] + "\""
                         + (k == a.length-1 ? " ]" : ", "));

    System.out.println();
}
```

Lunghezza dell'array (**attenzione:** .length **SENZA PARENTESI**)

Esercizio: somma dei valori di un array

- Scrivere un metodo che, dato un array di interi, restituisca la somma dei suoi elementi
- Ad esempio, dato l'array { 10, 12, 12, 8 } il metodo deve restituire: 42

```
public int sommaArray(int[] a)
{
    int val = 0;

    for (int k = 0; k < a.length; k++) val += a[k];

    return val;
}
```

Esercizio: media dei valori di un array

- Utilizzando il metodo scritto per l'esercizio precedente, scrivere un metodo che, dato un array di interi, restituisca la media (double) dei suoi elementi
- Ad esempio, dato l'array { 10, 12, 12, 8 } il metodo deve restituire: 10.5

```
public double mediaArray(int[] a)
{
    return sommaArray(a)/a.length;
}
```

occhio: divide intero per intero (cosa serve?)

RIUSO DEL CODICE!!!

- Scrivete anche la versione per array di double!

Altri esercizi molto semplici con gli array

- Scrivere un metodo che, dato un array di stringhe e una stringa in input, restituisca true se l'array contiene la stringa, false altrimenti
- Scrivere una seconda versione del metodo che restituisca la posizione della stringa trovata, -1 altrimenti
- Scrivere un metodo che, dato un array di double, restituisca il valore massimo dell'array

Esercizio: reimplementa conta vocali

- Scrivere un metodo che riceve una stringa e stampa a video il conteggio delle vocali in essa contenute
  - Utilizzare un array per il conteggio separato delle 5 vocali
- Ad esempio: data la stringa “le aiuole sono pulite”, il metodo stampa: `a=1 e=3 i=2 o=3 u=2`
- Come l’avevate implementato **SENZA** array?

Esercizio: array con fattoriale

- Scrivere un metodo che, dato un intero n in ingresso, restituisca un array di dimensione n contenente k! nella k-esima cella, per ogni valore di k

```
public int[] getArrayFattoriali(final int n)
{
    int[] fatt = new int[n];

    fatt[0] = 1;
    for (int k = 1; k < n; k++) fatt[k] = fatt[k-1]*k;

    return fatt;
}
```

Esercizio: “l’array risponde”

Progettare una classe MioArray i cui oggetti vengono costruiti con un array di interi (int[ ])

- La classe implementa i seguenti metodi:

– contiene, che dati in ingresso una posizione e un intero, restituisce true o false se l’intero è contenuto in quella posizione nell’array

– somma2, che restituisce la somma dei primi due elementi dell’array. Se l’array è di lunghezza inferiore (**info**: la lunghezza dell’array a si ottiene con il campo speciale length, quindi a.length), restituisce il valore del primo elemento oppure 0 se l’array è vuoto

– scambia, che date in ingresso due posizioni intere, scambia i valori presenti nelle due posizioni dell’array (es. scambia(1, 3) trasforma l’array { 1, 2, 3, 4, 5 } in { 1, 4, 3, 2, 5 })

– maxTripla: che restituisce il valore massimo tra il primo, l’ultimo e il valore in posizione intermedia dell’array (es. restituisce 3 se l’oggetto è costruito con { **1**, 7, 5, **3**, 0, 2, **2** }, le posizione esaminate sono in grassetto)

– falloInDue: che restituisce un array di due interi, il primo è il primo elemento dell’array dell’oggetto, il secondo è l’ultimo elemento dell’array dell’oggetto

Esercizio: stampa di istogrammi

- Progettare una classe Istogramma che rappresenta la distribuzione di dati (es. voti degli studenti) in un intervallo da i a j fornito in input (es. da 0 a 31 (trenta e lode))
- La classe permette di incrementare il conteggio in corrispondenza di ciascun elemento dell'intervallo (es. memorizzando così un nuovo voto di uno studente)
- La classe può stampare a video l'istogramma corrispondente
  - Più facile in orizzontale
  - Provate a stampare in verticale!!!

Esercizio: mescolare e distribuire un mazzo di carte da gioco

- Progettare una classe Carta che rappresenti una singola carta da gioco (con seme e valore)
  - La classe deve restituire su richiesta la propria rappresentazione sotto forma di stringa
- Progettare quindi una classe MazzoDiCarte che rappresenti un intero mazzo da 52 carte
- La classe deve implementare i seguenti metodi:
  - mescola il mazzo di carte
  - distribuisci la prossima carta
- Infine si progetti una classe di collaudo che crea un mazzo, mescoli le carte e ne distribuisca carte fino ad esaurimento del mazzo

Esercizio: DaCifreALettere

- Scrivere un metodo che prenda in ingresso una stringa contenente cifre e restituisca una stringa in cui ciascuna cifra è stata trasformata nella parola corrispondente
- Ad esempio, data in input la stringa “8452”, il metodo restituisce “otto quattro cinque due”
- Viceversa, scrivere un metodo che prenda in ingresso una stringa contenente cifre scritte a lettere e restituisca una stringa contenente le cifre corrispondenti
- Ad esempio, data in input la stringa “otto quattro cinque due”, il metodo restituisce “8452”
- **Nota:** è conveniente utilizzare gli array di stringhe String[]!

Esercizio: da cifre a lettere e viceversa (avanzato!!!)

- Come l'esercizio precedente, ma stampando (o leggendo) a lettere tenendo conto della posizione delle cifre (occhio ai **casi speciali**: undici, dodici, ecc…)
- Ad esempio, "8452" viene trasformato in "ottomila quattrocento cinquanta due"

Esercizio: implementare un filtro

- Progettare una classe Filtro costruita con un array di interi
- La classe implementa operazioni che permettono di ottenere nuovi sotto-array dell'array iniziale:
  - passaBasso: restituisce tutti gli elementi <= k nell'ordine iniziale
  - passaAlto: restituisce tutti gli elementi >= k nell'ordine iniziale
  - filtra: restituisce l'array iniziale da cui sono state eliminate tutte le occorrenze dell'intero passato in input
  - filtra: una seconda versione del metodo che restituisce l'array iniziale da cui vengono eliminate tutte le occorrenze di interi presenti nell'array passato in input
- Se Filtro viene costruito con l'array { 1, 2, 10, 2, 42, 7, 8 }:
  - passaBasso(8) restituisce { 1, 2, 2, 7, 8 }
  - passaAlto(9) restituisce { 10, 42 }
  - filtra(2) restituisce { 1, 10, 42, 7, 8 }
  - filtra(new int[] { 2, 7, 42 }) restituisce { 1, 10, 8 }

Esercizio: reimplementare Arrays.copyOf() e Arrays.toString()

- Implementare un metodo statico copyOf che, analogamente a java.util.Arrays.copyOf, copi un array in un nuovo array delle dimensioni specificate (troncando l'array in input, se più grande)

Esercizio: sequenza di cifre estensibile

- Progettare una classe SequenzaDiCifre che espone un metodo che, data in input una stringa e un intero N, aggiunga alla sequenza inizialmente vuota (rappresentata mediate un array) le prime N cifre contenute nella stringa (si assuma che ne contenga comunque almeno N). La classe espone anche un metodo toString che fornisce una rappresentazione sotto forma di stringa della sequenza.
- Ad esempio:
  - SequenzaDiCifre s = new SequenzaDiCifre();
  - s.aggiungiCifre("abc1--23", 2);
  - s.aggiungiCifre("xx0a8b76543100", 4);
  - System.out.println(s.toString());
- stampa: [1,2,0,8,7,6]

Esercizio: implementare una lista mediante array

- Che cos'è una lista? E' una sequenza di oggetti
- Implementare una classe ListaDiInteri che permetta le seguenti operazioni:
  - Restituisce l'elemento i-esimo della lista
  - Restituisce l'indice della posizione dell'intero fornito in input
  - Restituisce una stringa formattata contenente la lista di interi
  - Restituisce la dimensione della lista
  - Contiene un determinato intero (true o false)?
  - Aggiungi un intero in coda alla lista
  - Aggiungi un intero nella posizione specificata
  - Elimina la prima occorrenza di un intero dalla lista
  - Elimina l'elemento i-esimo della lista

Esercizio: la tavola pitagorica

- Scrivere una classe che rappresenti la tavola pitagorica NxN (dove l'intero N è un parametro di costruzione della classe)
- La classe deve, su richiesta, restituire il valore della tabella in corrispondenza della posizione (i, j)

La classe deve poter stampare l'intera tavola

x	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	10
2	0	2	4	6	8	10	12	14	16	18	20
3	0	3	6	9	12	15	18	21	24	27	30
4	0	4	8	12	16	20	24	28	32	36	40
5	0	5	10	15	20	25	30	35	40	45	50
6	0	6	12	18	24	30	36	42	48	54	60
7	0	7	14	21	28	35	42	49	56	63	70
8	0	8	16	24	32	40	48	56	64	72	80
9	0	9	18	27	36	45	54	63	72	81	90
10	0	10	20	30	40	50	60	70	80	90	100

Esercizio: il gioco del tris

- Progettare una classe ScacchieraTris che implementi la scacchiera del gioco del tris
- La classe deve memorizzare la scacchiera i cui elementi possono essere:
  - " " (se non è stata ancora occupata la casella)
  - "X" oppure "O" (secondo il giocatore che ha occupato la casella)
- La classe deve stampare in qualsiasi momento la situazione della scacchiera
- Deve permettere di occupare una casella con un simbolo "X" o "O"
- Progettare quindi una classe Tris che implementi il gioco utilizzando la scacchiera appena progettata