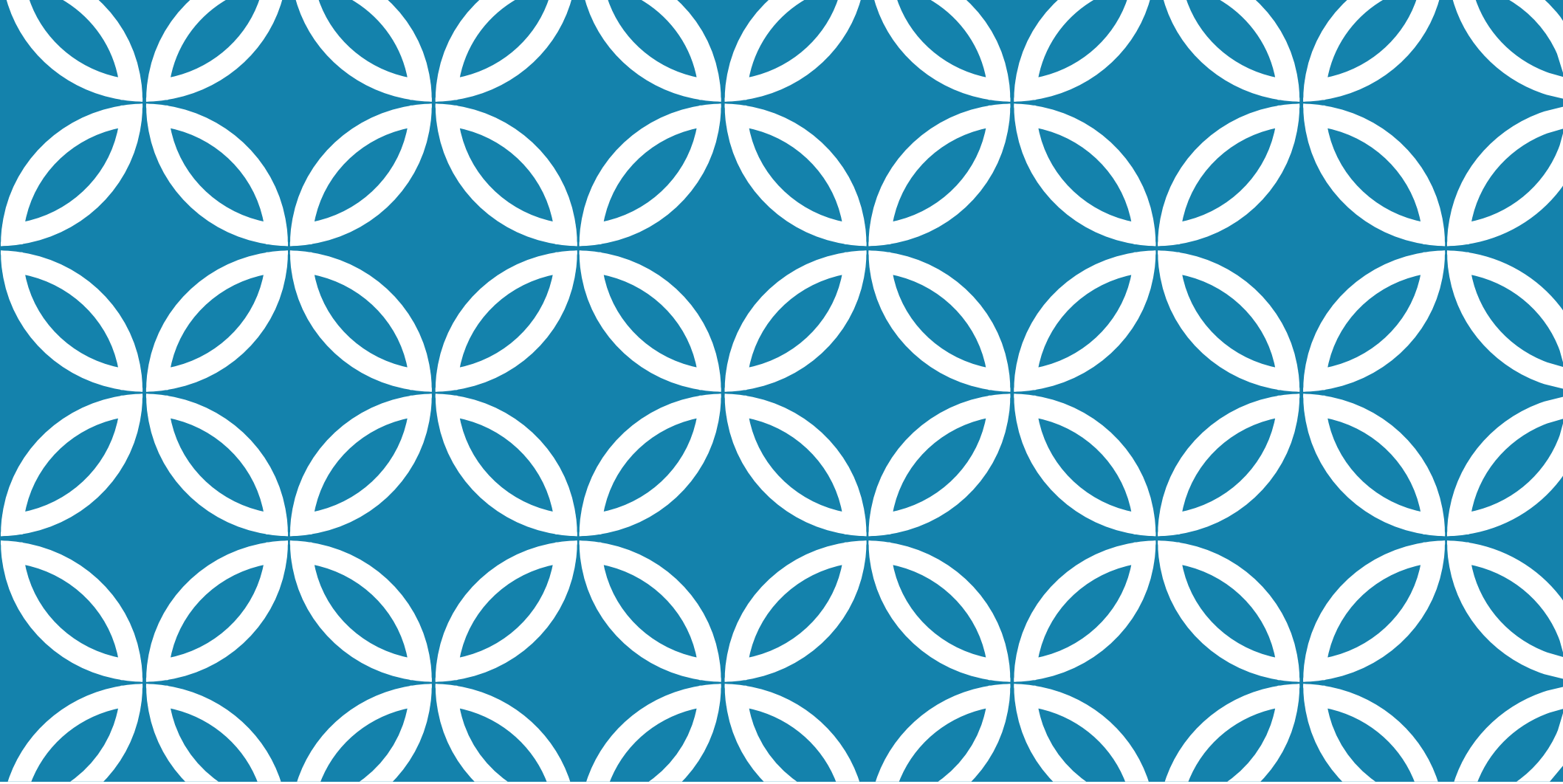

LABORATORIO DI PROGRAMMAZIONE

Corso di Laurea Ing. Gestionale 21/22

Ing. Antonio Luca Alfeo

luca.alfeo@ing.unipi.com

RECAP CLASSI IN JAVA (1/2)

- Una classe rappresenta una categoria di oggetti, entità informatiche definita da:
  - attributi, che definiscono le caratteristiche dell'oggetto;
  - metodi, che definiscono le operazioni che si possono fare sull'oggetto.
- Gli oggetti di una determinata classe prendono il nome di istanze della classe.

```
public class Persona {

     // Variabili (attributi)
     private String nome;
     private int eta;

     // Costruttore
     public Persona(String n, int e){
          nome = n;
          eta = e;
     }

     // Metodi
     public void presentati(){
          System.out.println("Ciao, sono " + nome
 + " ed ho " + eta + " anni");
     }

     public void invecchia(int nAnni){
          eta += nAnni;
     }

 }
```

```
public class TestPersona {

     public static void main(String[] args) {
          /* Creazione di un'istanza di Persona
           * tramite costruttore */
          Persona a = new Persona("Mario", 27);

          /* Invocazione di metodi */
          a.presentati();
          // Ciao, sono Mario ed ho 27 anni
          a.invecchia(3);
          a.presentati();
          // Ciao, sono Mario ed ho 30 anni

     }

 }
```

RECAP CLASSI IN JAVA (2/2)

- I modificatori *public* e *private* permettono di stabilire la visibilità di un membro (variabile o metodo) di una classe.
  - I membri *private* sono visibili solo alla classe di appartenenza.
  - I membri *public* sono visibili anche alle altre classi.
- Generalmente gli attributi di una classe sono privati. Per accedere in maniera controllata al valore di questi attributi si definiscono dei metodi che prendono il nome di *getter* (permettono di leggere un attributo) o *setter* (permettono di modificare un attributo).
- Se una *variabile* di una classe ha il modificatore *static*, non ne esiste una copia per ogni oggetto, ma è unica per tutti le istanze della classe.
- Se un *metodo* ha il modificatore *static* può essere riferito senza specificare un'istanza, ma non può accedere a variabili non statiche.
- Solo i metodi non *static* di una classe, oltre ai parametri dichiarati formalmente ha un parametro implicito: il riferimento *this*, che è un riferimento all'oggetto sul quale viene invocato il metodo grazie al quale è possibile accedere ai membri dell'oggetto.
- Il metodo *toString()* puo' essere definito per restituire la rappresentazione testuale dell'oggetto su cui è invocato e puo' essere usato assieme a *System.out.print()* per produrne la stampa a video.

ESERCIZIO "CONTOCORRENTE"

- Creare una classe *ContoCorrente* che permetta di gestire un conto corrente bancario caratterizzato dai seguenti attributi: *nome dell'intestatario* (tipo String), *cognome dell'intestatario* (tipo String), *numero di conto corrente* (tipo long), *saldo residuo* (tipo double).
- Il costruttore deve avere come parametri il nome, il cognome ed il saldo iniziale. Il numero di conto corrente deve essere calcolato automaticamente in maniera incrementale (si può realizzare usando una variabile statica).
- Realizzare i metodi:
  - *deposita*, che permette di depositare denaro sul conto corrente
  - *preleva*, che permette di prelevare denaro dal conto corrente
  - *toString*, che ritorna le informazioni sul conto corrente (numero, intestatario, saldo residuo)
- Per testare il corretto funzionamento della classe *ContoCorrente*, realizzare la classe *TestConto* ed il relativo metodo main che esegue le seguenti operazioni:
  - Crea due conti correnti intestati a due persone diverse
  - Effettua un'operazione di bonifico dal primo al secondo conto (utilizzano i metodi preleva e deposita)
  - Stampa a video le informazioni dei conti prima e dopo il bonifico

SOLUZIONE (1/3)

```
public class ContoCorrente {
     /* Dati personali intestatario */
     private String nome;
     private String cognome;
     /* Numero di conto corrente */
     private long numeroConto;
     /* Saldo attuale */
     private double saldoAttuale;

     /* Contatore interno per l'assegnazione del
      * numero di c/c per ogni nuovo conto */
     private static long prossimoNumeroConto = 1;

     /* Costruttore */
     public ContoCorrente(String n, String c,  double saldoIniz) {
          nome = n;
          cognome = c;
          numeroConto = prossimoNumeroConto;
          saldoAttuale = saldoIniz;
          /* Incremento il contatore statico: la prossima
           * invocazione del costruttore assegnerà
           * il valore successivo */
          prossimoNumeroConto++;
     }
```

SOLUZIONE (2/3)

```
    public boolean deposita(double ammontare) {
        boolean opValida = (ammontare > 0);
        if (opValida) {
            saldoAttuale += ammontare;
        }
        return opValida;
    }
    public boolean preleva(double ammontare) {
        boolean opValida = ((ammontare > 0 ) &&
                                (saldoAttuale >= ammontare ));
        if (opValida) {
            saldoAttuale -= ammontare;
        }
        return opValida;
    }
    public String toString() {
        return "Il signor " + cognome + " " + nome
                + " è titolare del c/c n. " + numeroConto
                + " con saldo pari a " + saldoAttuale;
    }
}
```

SOLUZIONE (3/3)

```
public class TestConto {
     public static void main(String[] args) {
          ContoCorrente cc1;
          cc1 = new ContoCorrente("Mario", "Rossi", 5000);

          ContoCorrente cc2;
          cc2 = new ContoCorrente("Fabio", "Pagani", 15000);

          System.out.println(cc1);
          System.out.println(cc2);

          if (cc1.preleva(500)) {
              cc2.deposita(500);

 System.out.println("Bonifico effettuato " + "con successo!");

          } else {

    System.out.println("Non è stato possibile " + "effettuare il bonifico!");

          }

          System.out.println(cc1);
           System.out.println(cc2);
     }
 }
```

ARRAY DI OGGETTI

```
public class Persona {

     // Variabili (attributi)
     private String nome;
     private String compleanno;

     // Costruttore
     public Persona(String n, String c){
         nome = n;
         compleanno = c;
     }

  public String toString(){

 return "Ciao, sono " + nome + " e compio
 gli anni il " + compleanno;

           }
     }

 }
```

```
public class TestPersona {

   public static void main(String[] args) {

       Persona[] partecipanti = new Persona[5];

       for (int i=0; i<5; i++) {

         String nome = Lettore.in.leggiString();

         String compleanno = Lettore.in.leggiString();

         partecipanti[i] = new Persona(nome, compleanno);

        }

    System.out.println(partecipanti[2]);

    System.out.println(partecipanti[4]);

    }

 }
```

ESERCIZIO "PERSONE"

- Creare una classe *Persona* come da slide precedente, e:
  - Modificare il metodo *toString()* affinche stampi il compleanno (stringa presa da tastiera nel formato MM/GG) nel formato «GG di MESE». MESE deve essere scritto in forma estesa a parole. Rivedere l'esercizio conversioneData della lezione 6.
- Per testare il corretto funzionamento della classe *Persona*, realizzare la classe *TestPersona* ed il relativo metodo main che esegue le seguenti operazioni:
  - Crea e popola un array di 5 istanze della classe *Persona*
  - Ordina l'array in base alla data di compleanno. Rivedere slides lezione 8.
  - Stampa a video le informazioni contenute nell'array con il metodo toString()

SOLUZIONE 1/2

```
public class Persona {
    private String nome;   private String compleanno;

    public Persona(String n, String c){
        nome = n;
        compleanno = c;
    }

    public String getCompleanno() {
        return compleanno;
    }

    public String toString(){
        return "Ciao, sono " + nome + " e compio gli anni il " + convertiData();
    }

    private String convertiData() {
        int gg_idx = compleanno.indexOf('/');
        String mm_str = compleanno.substring(0, gg_idx);
            int mm = Integer.parseInt(mm_str);
        String gg_str = compleanno.substring(gg_idx + 1);

        String mm_c_str;
            switch (mm) {
            case 1:
                mm_c_str = "Gennaio";
                break;
             case 2:
                mm_c_str = "Febbraio";
                break;
            case 3:
                mm_c_str = "Marzo";
                break;
            case 4:
                mm_c_str = "Aprile";
                break;
            case 5:
                mm_c_str = "Maggio";
                break;
            case 6:
                mm_c_str = "Giugno";
                break;
            case 7:
                mm_c_str = "Luglio";
                break;
            case 8:
                mm_c_str = "Agosto";
                break;
            case 9:
                mm_c_str = "Settembre";
                break;
            case 10:
                mm_c_str = "Ottobre";
                break;
            case 11:
                mm_c_str = "Novembre";
                break;
            case 12:
                mm_c_str = "Dicembre";
                break;
            default:
                mm_c_str = "caso non previsto";
            }

            return gg_str + "di" + mm_c_str;
    }

}
```

SOLUZIONE 2/2

```
import fiji.io.Lettore;
public class testPersona {

    public static void main(String[] args) {
           Persona[] partecipanti = new Persona[5];
           for (int i=0; i<5; i++) {
             System.out.println("INSERISCI NOME");
             String nome = Lettore.in.leggiString();
             System.out.println("INSERISCI COMPLEANNO");
             String compleanno = Lettore.in.leggiString();
             partecipanti[i] = new Persona(nome, compleanno);
            }

            for (int i = 0; i < partecipanti.length - 1; i++) {
                for (int j = i + 1; j < partecipanti.length; j++) {
                    if (partecipanti[i].getCompleanno().compareTo(partecipanti[j].getCompleanno())>0) {
                    // scambio degli elementi
                    Persona tmp = partecipanti[i];
                    partecipanti[i] = partecipanti[j];
                    partecipanti[j] = tmp;
                    }
                }

            }

            for (int i=0; i<5; i++) {
                System.out.println(partecipanti[i]);
            }
    }
}
```