



Esercizi

Esercizi

- Calcolo dell'n-esimo numero di Fibonacci**
 - Verifica se un numero e' primo**
 - Verifica se un anno è bisestile**
 - Quanti giorni in un mese?**
 - Il giorno dopo**
-

Fibonacci

- Scrivere un'applicazione che calcoli l'n-esimo numero di Fibonacci
 - Scrivere la classe con
 - il metodo **fibonacci**
 - il metodo **main** per leggere il numero n da tastiera, invocare il metodo fibonacci e stampare il numero di Fibonacci calcolato.

nota. I primi numeri di Fibonacci sono:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

ovvero $\text{Fib}(1) = 1$; $\text{Fib}(2) = 1$; $\text{Fib}(N) = \text{Fib}(N-1) + \text{Fib}(N-2)$

esempio. il sesto numero di Fibonacci è 8 (ovvero $5+3$).

Fibonacci – il metodo

```
public static int fibonacci(int n){
    int fib, fibPrec, fibPrecPrec;
    int i;

    if ((n==1)|| (n==2))
        fib = 1;
    else{
        fibPrec=1;
        fibPrecPrec=1;
        fib = fibPrecPrec + fibPrec;
        i=3;
        while(i<n){
            fibPrecPrec = fibPrec;
            fibPrec = fib;
            fib = fibPrecPrec + fibPrec;
            i=i+1;
        }
    }
    return fib;
}
```

Fibonacci – l'applicazione

```
import java.util.*;
class Fib{

    public static int fibonacci(int n){
        ... (vedere la slide precedente)
    }

    public static void main(String[] args){
        int num;
        Scanner in = new Scanner(System.in);

        //pre: n>=0
        System.out.println("Scrivi un numero (>=1)");
        num = in.nextInt();
        System.out.print("il numero di Fibonacci richiesto e'=");
        System.out.println(fibonacci(num));
    }
}
```

Verifica se un numero è primo

- Scrivere un'applicazione che verifica se un numero è primo
 - Scrivere la classe con
 - il metodo `primo`
 - il metodo `main` per leggere il numero n da tastiera, invocare il metodo `primo` e stampare un messaggio che dica se il numero è primo o non lo è.

Algoritmo per la verifica se un numero è primo

Leggi **n** da input

Inizializza **div** a 2

Inizia a provare se **div** divide **n**:

se **div** divide **n** fine - perché **n** non è primo

Finché (**div** non divide **n**) e (**div** non è arrivato a $n/2$)

incremento **div** di 1

se esci dal ciclo perché **n** è divisibile per **div**

allora **n** è primo

altrimenti **n** non è primo

verifica se un numero è primo – il metodo

```
public static boolean primo(int n){
    int div; //divisori per la verifica
    boolean trovato; //true se è stato trovato un divisore

    div = 2;
    trovato = false;

    while (!trovato && (div<=n/2)){
        if (n%div == 0)
            trovato = true;
        div = div +1;
    }
    /* esco dal ciclo o quando trovato è true (il num è
       divisibile) o quando div > n/2 li ho provati tutti */

    return !trovato;
}
```

verifica se un numero è primo – l'applicazione

```
import java.util.*;
/* Applicazione che verifica se un numero è primo */
class Primi{

    public static boolean primo(int n){ ... }

    public static void main(String[] args){
        int n; //numero letto da input
        Scanner in = new Scanner(System.in);
        //pre: n>0
        /*lettura del numero da input*/
        System.out.println("Scrivi un numero (>0)");
        n = in.nextInt();
        if (primo(n))
            System.out.print("il numero "+n+" e' primo");
        else
            System.out.print("il numero "+n+" non e' primo");
    }
}
```

Verifica se un anno è bisestile

- Scrivere un'applicazione che verifichi se un certo anno è bisestile
 - Scrivere la classe con
 - il metodo **bisestile**
 - il metodo **main** per leggere i dati di ingresso da tastiera, invocare il metodo bisestile e stampare i dati di uscita

Condizioni

□ Un anno (>1600) per essere bisestile deve:

- essere divisibile per 4 e inoltre
- qualora sia divisibile per 100 deve essere anche divisibile per 400

Esempi

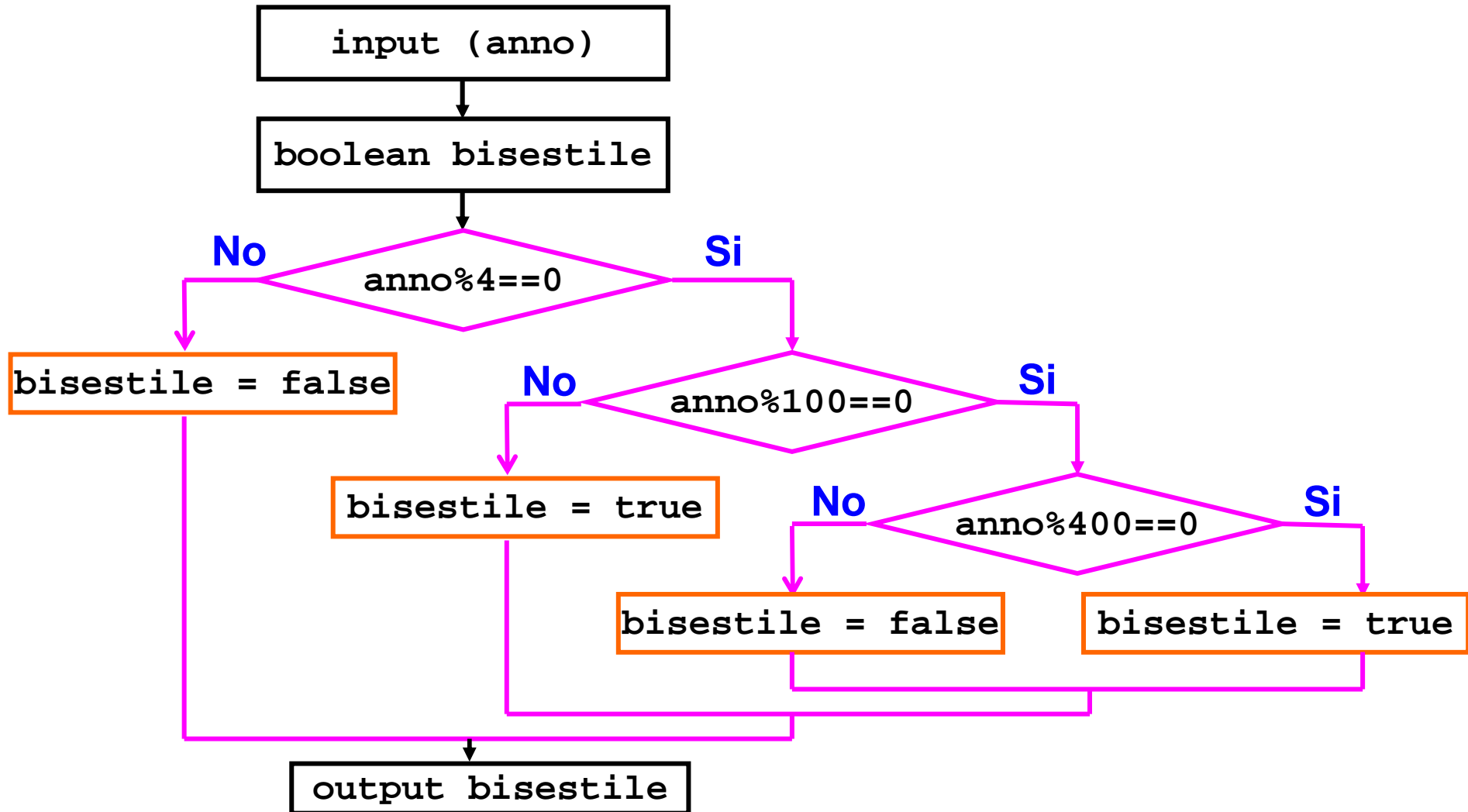
- L'anno 2004 è bisestile
- L'anno 1900 non è bisestile
- L'anno 2000 è bisestile

Algoritmo per verificare se un anno è bisestile

- ❑ Dato un numero intero **data**, verifico che tutte le condizioni siano verificate:

- ❑ verifica che **data** sia divisibile per 4
- ❑ se true allora
 - verifica che **data** sia divisibile per 100
 - se true allora
 - verifica che **data** è anche divisibile per 400
 - se true allora restituisci “bisestile”
 - altrimenti restituisci “non bisestile”
 - altrimenti restituisci “bisestile”
- ❑ altrimenti restituisci “non bisestile”

Diagramma a blocchi



Bisestile – il metodo

□ Realizziamo un metodo **bisestile** che ha come unico parametro di input un intero **data**, che rappresenta un anno, e restituisce **true** se l'anno è bisestile, altrimenti **false**

```
public static boolean bisestile(int data){
    boolean bis;    //vero se l'anno e' bisestile

    if(data%4==0){ // se è divisibile per 4
        if(data%100==0){
            /* se è divisibile per 100          */
            /* allora per essere bisestile     */
            /* deve essere divisibile per 400 */
            if(data%400==0)
                bis = true;
            else
                bis = false;
        }
        /* è divisibile per 4 ma non per 100 */
        else bis = true;
    }
    /* non e' divisibile per 4 */
    else bis = false;

    return bis;
}
```

Bisestile – l'applicazione

```
import java.util.*;
/* Applicazione che verifica se un anno e' bisestile
 */
class Bisestile{

    public static boolean bisestile(int data){...}

    public static void main(String[] args){
        //precondizione: anno >1600
        int anno; //INPUT
        Scanner in = new Scanner(System.in);

        /*lettura dei parametri di input */
        System.out.println("Scrivi una data corrisp. ad un anno (>1600");
        anno = in.nextInt();

        if (bisestile(anno))
            System.out.println("e' bisestile");
        else
            System.out.println("NON e' bisestile");

    } //end main
}
```

Quanti giorni in un mese?

- ❑ Scrivere un metodo che ricevuto in input un mese e un anno, restituisca il numero dei giorni del mese (verificando, per febbraio, se l'anno è bisestile)

NOTA. Modalità per inserire la data:

Esempio se il mese e' dicembre e l'anno 2004

Input: 12 2004

Output: 31

I giorni di un mese

- Il numero di giorni per un dato mese è calcolato dal seguente metodo, che tiene conto se un anno è bisestile o meno:

```
public static int giorniMese(int mese,int anno){
    int giorni;
    if(mese==2)
        if(Bisestile.bisestile(anno))
            giorni = 29;
        else
            giorni = 28;
    else
        if(mese==4 || mese==6 || mese==9 || mese==11)
            giorni = 30;
        else giorni = 31;
    return giorni;
}
```

Il giorno dopo

- Scrivere un'applicazione che calcoli la data del giorno successivo ad una data ricevuta in input
 - Scrivere la classe con
 - il metodo `giorniMese`
 - il metodo `main` per leggere i dati di ingresso da tastiera, calcolare il risultato e stampare i dati uscita

Condizioni

□ Una *data* è rappresentata da tre diversi input interi: **giorno, mese, anno**.

□ Generalmente la data del giorno dopo viene calcolata incrementando di uno il valore del giorno. **13/1/2005 \Rightarrow 14/1/2005**

□ Dobbiamo fare attenzione alle **condizioni limite**:

- Se il giorno corrisponde all'ultimo del mese.

$29/2/2004 \Rightarrow 1/3/2004$

- Se il giorno è l'ultimo del mese ed il mese è dicembre.

$31/12/2004 \Rightarrow 1/1/2005$

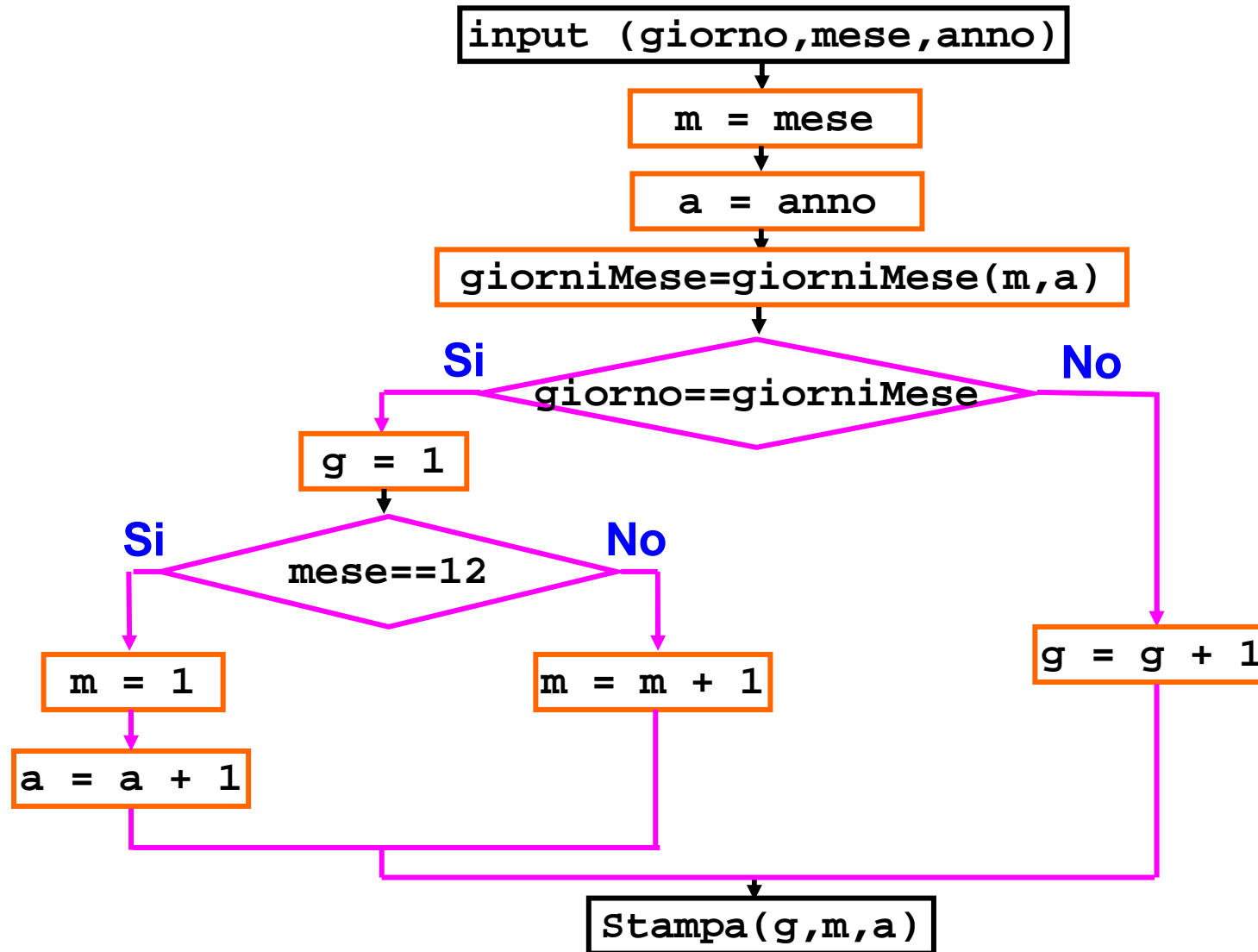
L'algoritmo

- ❑ Letti **giorno, mese e anno** verifico che non mi trovo in una condizione limite e incremento il **giorno**.
- ❑ Impiego la variabile intera **giorni_del_mese** per rappresentare il massimo numero dei giorni per il **mese**.

...L'algoritmo

- ❑ Verifico che **giorno** sia uguale a **giorni_del_mese**
- ❑ Se **true**, allora
 - **giorno** viene posto a 1
 - verifico che **mese** sia uguale a 12 (dicembre)
 - se **true**, allora
 - **mese** viene posto a 1
 - incremento **anno** di 1
 - Altrimenti, incremento **mese** di 1
- ❑ Altrimenti incremento **giorno** di 1

Il giorno dopo



IlGiornoDopo

```
import java.util.*;
/* Applicazione che calcola la data del giorno dopo */
class IlGiornoDopo{

public static int giorniMese(int mese,int anno){...}

public static void main(String[] args){

    int giorno, mese, anno; //INPUT
    int g,m,a;//OUTPUT
    int giorni_del_mese; //quanti giorni ha il mese m
    Scanner in = new Scanner(System.in);

    /*lettura della data di partenza senza controllo
        sulla correttezza dei dati */
    giorno = in.nextInt();
    mese = in.nextInt();
    anno = in.nextInt();
```

La classe: IlGiornoDopo

```
a=anno; //nel caso in cui non vengano modificate
m=mese;
/* calcolo dei giorni del mese */
giorni_del_mese=giorniMese(mese, anno);

/* calcolo della data successiva */
/* se il giorno e' l'ultimo del mese */
if (giorno==giorni_del_mese){
    g=1;
    if (mese==12){ //ultimo giorno dell'anno
        m=1;
        a=anno+1;}
    else
        m=mese+1;
}
else //nel caso generale basta incrementare il giorno
    g=giorno+1;

System.out.println(g + " " +m+ " " + a);
} }
```