

# Relazione del Progetto di Informatica

Abu Shahid Islam & Giovanni Sebastiani

26 novembre 2024

# Indice

<b>1</b>	<b>Struttura</b>	<b>3</b>
1.1	Menu Principale . . . . .	3
1.2	Prestiti . . . . .	3
<b>2</b>	<b>Menu Principale</b>	<b>4</b>
2.1	Gestione Array . . . . .	4
2.2	Visual . . . . .	5
2.2.1	MenuArrow.cs . . . . .	6
2.3	AddBook.cs . . . . .	8
2.3.1	Step 1 . . . . .	8
2.3.2	Step 2 . . . . .	9
2.3.3	Step 3 . . . . .	10
2.4	ShowBook.cs . . . . .	11
2.4.1	Step 1 . . . . .	11
2.5	RemoveBook.cs . . . . .	12
2.5.1	Step 1 . . . . .	12
2.5.2	Step 2 . . . . .	12
2.5.3	Step 3 . . . . .	13
2.6	SearchByAuthor.cs . . . . .	14
2.6.1	Step 1 . . . . .	14
2.6.2	Step 2 . . . . .	15
2.6.3	Step 3 . . . . .	15
2.7	MostExpeBook.cs . . . . .	16
2.7.1	Step 1 . . . . .	16
2.8	Reorder.cs . . . . .	17
2.8.1	Step 1 . . . . .	17
2.8.2	Step 2 . . . . .	17
2.8.3	Step 3 . . . . .	18
2.9	PriceRange.cs . . . . .	19
2.9.1	Step 1 . . . . .	19
<b>3</b>	<b>Prestiti</b>	<b>20</b>
3.1	MenuPrestiti.cs . . . . .	20
3.1.1	Step 1 . . . . .	20
3.2	TakeBorrow.cs . . . . .	22
3.2.1	Step 1 . . . . .	22
3.2.2	Step 2 . . . . .	23
3.2.3	Step 3 . . . . .	24
3.3	ReturnBook.cs . . . . .	25
3.3.1	Step 1 . . . . .	25
3.3.2	Step 2 . . . . .	26
3.3.3	Step 3 . . . . .	27
3.4	ShowBorrowedBooks.cs . . . . .	28
3.4.1	Step 1 . . . . .	28
<b>4</b>	<b>Utils e Logo</b>	<b>29</b>
4.1	ToLower.cs . . . . .	29
4.2	LogoBiblioteca.cs — LogoPrestito.cs . . . . .	29
<b>5</b>	<b>Github</b>	<b>30</b>
5.1	Come funziona? . . . . .	30

# 1 Struttura

Il programma è diviso in 2 **Menu** principali:

## 1.1 Menu Principale

- Qui l'utente può scegliere tra 9 opzioni:
  - **Aggiungi un libro**
  - **Visualizza i libri**
  - **Rimuovi un libro**
  - **Cerca un libro per autore**
  - **Libro più costoso**
  - **Ordina libri in base al prezzo**
  - **Visualizza per fascia di prezzo**
  - **Prestiti**
  - **Esci**

## 1.2 Prestiti

- Qui l'utente può scegliere tra 4 opzioni:
  - **Prendi in prestito**
  - **Restituisci il prestito**
  - **Visualizza i libri in prestito**
  - **Esci**
- Questo programma anche se fatto con Array statici (Array a dimensione fissa) dá la possibilità all'utente illimitati libri

## 2 Menu Principale

### 2.1 Gestione Array

Funzionamento del Menu Principale:

- Chiamato **Program.cs** contiene il *main*.
- Il main contiene tutti gli array per la gestione della libreria

```
1 string[] Titoli = new string[2];  
2 string[] Autori = new string[2];  
3 double[] prezzo = new double[2];  
4 string[] categoria = new string[2];  
5 string[] casa_editrice = new string[2];  
6 int[] copie = new int[2];
```

Listing 1: Array della libreria

- Contiene gli array per la gestione dei prestiti

```
1 int prestiti = 0;  
2 string[] libri_prestito = new string[2];  
3 string[] utenti_prestito_nome = new string[2];  
4 string[] utenti_prestito_cognome = new string[2];  
5 string[] giorno_preso = new string[2];  
6 string[] tempo_trattenuto = new string[2];  
7 int[] codice_prestito = new int[2];  
8 int returns = 0;
```

Listing 2: Array dei prestiti

- Alcune funzioni necessitano il numero di libri presenti nella libreria

```
1 int libri_unici = 0;  
2 int libri_unici_copia = 0;
```

Listing 3: Funzione per il numero di libri

## 2.2 Visual

- Tutta la parte visuale è gestita da un menu comandabile con le frecce *MenuArrow.cs*. Questa funzione ha come *return* la scelta dell'utente.
- In base alla scelta viene scelta la funzione da eseguire.

```
1      //Menu
2  while (true)
3  {
4      Console.Clear();
5
6      //Menu con le frecce
7      int scelta = MenuArrow.menuArrow();
8
9      switch (scelta)
10     {
11         case 1:
12
13             AddBook.addBook(ref Titoli, ref Autori, ref prezzo, ref categoria, ref casa_editrice,
14                             ref copie, ref libri_unici, ref libri_unici_copia);
15             break;
16
17         case 2:
18
19             ShowBook.showBook(Titoli, copie);
20             break;
21
22         case 3:
23
24             RemoveBook.removeBook(ref Titoli, ref Autori, ref prezzo, ref categoria, ref
25                                     casa_editrice, ref copie, ref libri_unici, ref libri_unici_copia);
26             break;
27
28         case 4:
29
30             SearchByAuthor.searchByAuthor(Titoli, Autori, copie);
31             break;
32
33         case 5:
34
35             MostExpeBook.mostExpeBook(Titoli, prezzo);
36             break;
37
38         case 6:
39
40             Reorder.reorder(ref Titoli, ref Autori, ref prezzo, ref categoria, ref casa_editrice,
41                             ref copie);
42             break;
43
44         case 7:
45
46             PriceRange.priceRange(Titoli, prezzo, copie);
47             break;
48
49         case 8:
50
51             MenuPrestiti.menuPrestiti(Titoli, Autori, prezzo, categoria, casa_editrice, copie, ref
52                                     libri_unici_copia, ref prestiti, ref libri_prestito,
53                                     ref utenti_prestito_nome, ref utenti_prestito_cognome, ref giorno_preso, ref
54                                     tempo_trattenuto, ref codice_prestito, ref retuns);
55             break;
56
57         case 9:
58
59             Console.Clear();
60             Console.WriteLine("Arrivederci!");
61             Console.ReadKey();
62
63             return;
64     }
65 }
```

Listing 4: Funzione per il menu

### 2.2.1 MenuArrow.cs

Questa funzione permette di navigare nel menu con le frecce. Le frecce funzionano tramite un array `strings` che contiene le scelte del menu. Un intero `pos` che indica la posizione attuale nel menu; e leggendo il tasto premuto, se è una freccia, cambia la posizione.

Nel seguente codice si può vedere come funziona la chat colorata e la freccia che indica la scelta dell'utente.

```
1 //Stampa il menu
2 static public void Menu(int pos)
3 {
4     // Menu
5     string[] strings =
6     {
7         "Inserisci un libro",
8         "Consulta tutti i libri della biblioteca",
9         "Rimuovi un libro",
10        "Ricerca libri per autore",
11        "Stampa libro piu costoso",
12        "Ordina i libri in base al prezzo",
13        "Stampa i libri per fascia di prezzo",
14        "Prestito",
15        "Esci"
16    };
17
18    // Calcola la larghezza della console
19
20    int consoleWidth = Console.WindowWidth;
21
22    for (int i = 0; i < strings.Length; i++)
23    {
24        // Calcola il numero di spazi iniziali per centrare la stringa
25        int padding = (consoleWidth - strings[i].Length) / 2;
26
27        // Assicurati che il padding sia almeno 0
28        if(padding < 0)
29        {
30            padding = 0;
31        }
32
33
34        // Colore verde per l'opzione selezionata
35        if (i == pos - 1)
36        {
37            Console.ForegroundColor = ConsoleColor.Green;
38            Console.WriteLine(new string(' ', padding) + $"> {strings[i]}");
39        }
40        else
41        {
42            Console.ForegroundColor = ConsoleColor.White;
43            Console.WriteLine(new string(' ', padding) + $" {strings[i]}");
44        }
45    }
46
47    // Resetta il colore
48    Console.ResetColor();
49 }
```

Listing 5: MenuArrow.cs

- `Console.WindowWidth` restituisce la larghezza della console.
- `Console.ForegroundColor` cambia il colore del testo.
- `Console.ResetColor` resetta il colore del testo.
- `padding` calcola il numero di spazi iniziali per centrare la stringa.
- `new string(' ', padding)` crea una stringa di spazi in base al padding.

Invece qui si può vedere come funziona la freccia e la scelta dell'utente.

```
1 //Stampa il menu con la freccia
2
3 static public int menuArrow()
4 {
5     //Leggo il tasto premuto
6     ConsoleKeyInfo key;
7     int pos = 1;
8     do
9     {
10         Console.Clear();
11         Console.WriteLine();
12         Biblioteca.Logo();
13         Menu(pos);
14         Console.WriteLine();
15         key = Console.ReadKey();
16         Se premo freccia giu incremento la posizione quindi va all opzione dopo
17         if (key.Key == ConsoleKey.DownArrow)
18         {
19             if (pos < 9)
20             {
21                 pos++;
22             }
23         }
24         //Se premo freccia su decremento la posizione quindi va all'opzione prima
25         else if (key.Key == ConsoleKey.UpArrow)
26         {
27             if (pos > 1)
28             {
29                 pos--;
30             }
31         }
32     } while (key.Key != ConsoleKey.Enter);
33     return pos;
34 }
```

Listing 6: MenuArrow.cs

- **ConsoleKeyInfo** legge il tasto premuto.
- **Console.Clear** pulisce la console.
- **Console.ReadKey** legge il tasto premuto.
- **ConsoleKey.DownArrow** controlla se è stata premuta la freccia giù.
- **ConsoleKey.UpArrow** controlla se è stata premuta la freccia su.
- **ConsoleKey.Enter** controlla se è stato premuto il tasto invio.
- **pos** indica la scelta corrente.

## 2.3 AddBook.cs

- Questa funzione permette di aggiungere un libro alla libreria.
- **2.3.1 Step 1**

Primo step è controllare se ci sono posti disponibili dove mettere i libri. In caso contrario, il programma crea un nuovo spazio tramite *Array.Resize*.

```
1 Console.Clear();
2 //Se abbiamo raggiunto il massimo
3 if (libri_unici == Titoli.Length)
4 {
5     //Resize degli array
6
7     Array.Resize(ref Titoli, Titoli.Length + 1);
8     Array.Resize(ref Autori, Autori.Length + 1);
9     Array.Resize(ref prezzo, prezzo.Length + 1);
10    Array.Resize(ref categoria, categoria.Length + 1);
11    Array.Resize(ref casa_editrice, casa_editrice.Length + 1);
12    Array.Resize(ref copie, copie.Length + 1);
13 }
```

Listing 7: Controllo spazio disponibile



### 2.3.2 Step 2

- Inserimento dei dati del libro.
- Il programma controlla all’inserimento se il libro è già presente.
- in caso sia un libro nuovo chiede le altre informazioni.
- Se avviene qualche errore, Es. inserimento di una **stringa** al posto di un **double**, il programma annulla l’inserimento.

```
1 Console.WriteLine("Inserisci il titolo del libro");
2 string titolo = Console.ReadLine();
3
4 //Libro gia presente o meno
5
6 for (int i = 0; i < Titoli.Length; i++)
7 {
8     if (Titoli[i] == null)
9     {
10         continue;
11     }
12     if (ToLower.ToLowerString(Titoli[i]) == ToLower.ToLowerString(titolo))
13     {
14         copie[i]++;
15         Console.WriteLine("Il libro e gia presente nei nostri archivi, non servono le altre informazioni");
16         Console.ReadKey();
17         return;
18     }
19 }
20
21 try{
22     Titoli[libri_unici] = titolo;
23
24     Console.WriteLine("Inserisci l'autore del libro");
25     Autori[libri_unici] = Console.ReadLine();
26
27     Console.WriteLine("Inserisci il prezzo");
28     prezzo[libri_unici] = double.Parse(Console.ReadLine());
29
30     Console.WriteLine("Inserisci la categoria");
31     categoria[libri_unici] = Console.ReadLine();
32
33     Console.WriteLine("Inserisci la casa editrice");
34     casa_editrice[libri_unici] = Console.ReadLine();
35
36     copie[libri_unici] = 1;
37 }
38 catch (Exception e)
39 {
40     Console.WriteLine("Errore, Qualcosa e andato storto");
41     Console.ReadKey();
42     Array.Resize(ref Titoli, Titoli.Length - 1);
43     Array.Resize(ref Autori, Autori.Length - 1);
44     Array.Resize(ref prezzo, prezzo.Length - 1);
45     Array.Resize(ref categoria, categoria.Length - 1);
46     Array.Resize(ref casa_editrice, casa_editrice.Length - 1);
47     Array.Resize(ref copie, copie.Length - 1);
48     return;
49 }
```

Listing 8: Inserimento dei dati

- **Array.Resize** aumenta/diminuisce la dimensione dell'array.

### 2.3.3 Step 3

- Ultimo step è la stampa dei dati inseriti

```
1 Console.WriteLine("Libro aggiunto con successo");
2 Console.ReadKey();
3
4 string[] strings =
5 {
6     $"Libro: ",
7     Titoli[libri_unici],
8     $"Autore: ",
9     Autori[libri_unici],
10    $"Prezzo: ",
11    prezzo[libri_unici].ToString(),
12    $"Categoria: ",
13    categoria[libri_unici],
14    $"Casa Editrice",
15    casa_editrice [libri_unici],
16 };
17
18 Console.Clear();
19 int padding = 15;
20
21 for(int i=0; i < strings.Length; i+=2)
22 {
23
24     Console.Write(new string(' ', padding) + strings[i]);
25     int padding2 = Console.WindowHeight - (padding - strings[i].Length)/2;
26
27     Console.Write(new string(' ', padding2-i) + strings[i+1]);
28
29     if(i == 4)
30     {
31         Console.Write(" Eur");
32     }
33     Console.WriteLine();
34 }
35
36 //Aumento libri unici
37 libri_unici++;
38 libri_unici_copia++;
39
40 Console.ReadKey();
41
```

Listing 9: Stampa dei dati inseriti

## 2.4 ShowBook.cs

### 2.4.1 Step 1

Lo scopo di questa funzione è di visualizzare tutti i libri presenti nella libreria.

Questa funzione mostra un menu con tutti i libro, e alla selezione di ogni libro, mostra le informazioni di quel libro. Il modo di funzionare è simile al menu principale, ma con l'obbiettivo di visualizzare i libri.

```
1 public static void Menu(int pos, string[] Titoli)
2 {
3     Console.Clear();
4     Console.WriteLine("Indice Libri");
5
6     for(int i=0; i<Titoli.Length; i++)
7     {
8         if (i == pos-1)
9         {
10             Console.ForegroundColor = ConsoleColor.Green;
11             Console.WriteLine($"> [{i+1}] {Titoli[i]}");
12             Console.ResetColor();
13         }
14         else
15         {
16             Console.WriteLine($" [{i+1}] {Titoli[i]}");
17         }
18     }
19 }
20
21
22 public static void showBook(string[] Titoli, int[] copie, string[] Autori, double[] prezzo, string[]
23 categoria, string[] casa_editrice)
24 {
25     ConsoleKeyInfo key;
26     int pos = 1;
27     do{
28         Menu(pos, Titoli);
29         key = Console.ReadKey();
30
31         if(key.Key == ConsoleKey.DownArrow)
32         {
33             if(pos == Titoli.Length)
34             {
35                 pos = 1;
36             }
37             else
38             {
39                 pos++;
40             }
41         }
42         else if(key.Key == ConsoleKey.UpArrow)
43         {
44             if(pos == 1)
45             {
46                 pos = Titoli.Length;
47             }
48             else
49             {
50                 pos--;
51             }
52         }
53     }while(key.Key != ConsoleKey.Enter);
54
55     Console.Clear();
56
57     //Stampa libri
58     Console.WriteLine("Titolo: " + Titoli[pos-1]);
59     Console.WriteLine("Autore: " + Autori[pos-1]);
60     Console.WriteLine("Prezzo: " + prezzo[pos-1] + "Eur");
61     Console.WriteLine("Categoria: " + categoria[pos-1]);
62     Console.WriteLine("Casa Editrice: " + casa_editrice[pos-1]);
63     Console.WriteLine("Copie: " + copie[pos-1]);
64
65     Console.WriteLine("Premi un tasto per tornare al menu");
66     Console.ReadKey();
67
68     return;
69 }
```

## 2.5 RemoveBook.cs

### 2.5.1 Step 1

Lo scopo di questa funzione è di rimuovere un libro dalla libreria. Funziona con lo stesso identico schema di Menu visto precedentemente. Ogni libro ha un numero associato, e alla selezione di quel numero, il libro viene rimosso. In caso ci sia più di una copia elimina prima una copia, e poi il libro.

```
1 public static void removeBook(ref string[] Titoli, ref string[] Autori, ref double[] prezzo, ref string[]  
   categoria, ref string[] casa_editrice, ref int[] copie, ref int libri_unici, ref int  
   libri_unici_copia)  
2 {  
3     ConsoleKeyInfo key;  
4     int lettura = 1;  
5     do{  
6         Menu(lettura, Titoli, copie);  
7         key = Console.ReadKey();  
8  
9         if(key.Key == ConsoleKey.DownArrow)  
10        {  
11            if(lettura == Titoli.Length)  
12            {  
13                lettura = 1;  
14            }  
15            else  
16            {  
17                lettura++;  
18            }  
19        }  
20        else if(key.Key == ConsoleKey.UpArrow)  
21        {  
22            if(lettura == 1)  
23            {  
24                lettura = Titoli.Length;  
25            }  
26            else  
27            {  
28                lettura--;  
29            }  
30        }  
31    }while(key.Key != ConsoleKey.Enter);  
32 }
```

- **Menu** è la funzione che stampa il menu.
- **ConsoleKey.DownArrow** controlla se è stata premuta la freccia giù.
- **ConsoleKey.UpArrow** controlla se è stata premuta la freccia su.
- **ConsoleKey.Enter** controlla se è stato premuto il tasto invio.

### 2.5.2 Step 2

Eliminazione copia

```
1 if (copie[lettura - 1] > 1)  
2 {  
3     copie[lettura - 1]--;  
4     Console.Clear();  
5     Console.WriteLine("Copia eliminata con successo");  
6     Console.ReadKey();  
7     Console.Clear();  
8     return;  
9 }  
10
```

- **copie[lettura - 1] > 1** controlla se ci sono più di una copia.
- **copie[lettura - 1]--** elimina una copia.
- **Console.Clear** pulisce la console.

### 2.5.3 Step 3

#### Eliminazione libro

```
1  Titoli[lettura - 1] = "";
2  Autori[lettura - 1] = "";
3  prezzo[lettura - 1] = 0;
4  categoria[lettura - 1] = "";
5  casa_editrice[lettura - 1] = "";
6  copie[lettura - 1] = 0;
7
8  for (int i = lettura - 1; i < Titoli.Length - 1; i++)
9  {
10     string temp;
11     int tempint;
12     double tempdouble;
13
14     temp = Titoli[i + 1];
15     Titoli[i] = temp;
16
17     temp = Autori[i + 1];
18     Autori[i] = temp;
19
20     tempdouble = prezzo[i + 1];
21     prezzo[i] = tempdouble;
22
23     temp = categoria[i + 1];
24     categoria[i] = temp;
25
26     temp = casa_editrice[i + 1];
27     casa_editrice[i] = temp;
28
29     tempint = copie[i + 1];
30     copie[i] = tempint;
31 }
32
33 //Resize degli array
34 Array.Resize(ref Titoli, Titoli.Length - 1);
35 Array.Resize(ref Autori, Autori.Length - 1);
36 Array.Resize(ref prezzo, prezzo.Length - 1);
37 Array.Resize(ref categoria, categoria.Length - 1);
38 Array.Resize(ref casa_editrice, casa_editrice.Length - 1);
39 Array.Resize(ref copie, copie.Length - 1);
40
41 libri_unici--;
42 libri_unici_copia--;
43
44 Console.Clear();
45 Console.WriteLine("Il libro e stato eliminato");
46 Console.ReadKey();
47 Console.Clear();
```

- `Titoli[lettura - 1] = ""` elimina il titolo.
- `Array.Resize` diminuisce la dimensione dell'array.
- `libri_unici--` diminuisce il numero di libri unici.
- `libri_unici_copia--` diminuisce il numero di libri unici copia.

## 2.6 SearchByAuthor.cs

### 2.6.1 Step 1

Menu (guardare sopra per spiegazione) e ricerca per autore.

Il seguente codice, crea un array con tutti gli autori presenti, e permette di scegliere un autore.

```
1 static public void searchByAuthor(string[] Titoli, string[] Autori, int[] copie)
2 {
3     Console.Clear();
4     Console.WriteLine("Autori Presenti");
5
6     string[] autori_unici = new string[1];
7
8     //Metto dentro l'array tutti i autori
9     for (int i = 0; i < Autori.Length; i++)
10    {
11        if(Autori[i] == null) {
12            continue;
13        }
14        //Controllo che non ci siano doppioni
15        bool doppione = false;
16        for (int j = 0; j < autori_unici.Length; j++)
17        {
18            if (autori_unici[j] == null)
19            {
20                autori_unici[j] = Autori[i];
21                break;
22            }
23
24            if (ToLower.ToLowerString(Autori[i]) == ToLower.ToLowerString(autori_unici[j]))
25            {
26                doppione = true;
27                break;
28            }
29        }
30
31        if (!doppione)
32        {
33            Array.Resize(ref autori_unici, autori_unici.Length + 1);
34            autori_unici[autori_unici.Length - 1] = Autori[i];
35        }
36    }
37
38
39    if(Autori[0] == null)
40    {
41        Console.WriteLine("Nessun autore presente");
42        Console.ReadKey();
43        return;
44    }
```

- **autori\_unici** contiene tutti gli autori presenti.
- **ToLower.ToLowerString** trasforma la stringa in minuscolo.
- **Array.Resize** aumenta la dimensione dell'array.

## 2.6.2 Step 2

Gestore della scelta dell'autore

```
1 //Stampo
2 ConsoleKeyInfo key;
3 int scelta = 1;
4 do{
5     Menu(scelta, autori_unici);
6     key = Console.ReadKey();
7
8     if(key.Key == ConsoleKey.DownArrow)
9     {
10         if(scelta == autori_unici.Length)
11         {
12             scelta = 1;
13         }
14         else
15         {
16             scelta++;
17         }
18     }
19     else if(key.Key == ConsoleKey.UpArrow)
20     {
21         if(scelta == 1)
22         {
23             scelta = autori_unici.Length;
24         }
25         else
26         {
27             scelta--;
28         }
29     }
30 }while(key.Key != ConsoleKey.Enter);
```

- **Menu** è la funzione che stampa il menu.
- **ConsoleKey.DownArrow** controlla se è stata premuta la freccia giù.
- **ConsoleKey.UpArrow** controlla se è stata premuta la freccia su.
- **ConsoleKey.Enter** controlla se è stato premuto il tasto invio.
- **scelta** indica la scelta dell'utente.

## 2.6.3 Step 3

Stampa dei libri dell'autore scelto, tramite un ciclo **for** che controlla gli autori di tutti i libri e in caso affermativo stampa

```
1 //Stampo i libri dell'autore scelto
2 Console.Clear();
3 Console.WriteLine($"Libri di {autori_unici[scelta - 1]}");
4 int pos = 0;
5 for (int i = 0; i < Autori.Length; i++)
6 {
7     if (ToLower.ToLowerString(Autori[i]) == ToLower.ToLowerString(autori_unici[scelta - 1]))
8     {
9         Console.WriteLine($"[{pos + 1}] {Titoli[i]} \t x{copie[i]}");
10        pos++;
11    }
12
13 }
14
15 Console.ReadKey();
16 return;
17 }
```

## 2.7 MostExpeBook.cs

### 2.7.1 Step 1

Lo scopo di questa funzione è di trovare il libro più costoso. Cerca la posizione del prezzo più alto e stampa il libro.

```
1 public static void mostExpeBook(string[] Titoli, double[] prezzo)
2 {
3     Console.Clear();
4     double max = -1;
5     int pos = 0;
6
7     for (int i = 0; i < prezzo.Length; i++)
8     {
9         if (prezzo[i] > max)
10         {
11             max = prezzo[i];
12             pos = i;
13         }
14     }
15
16     Console.WriteLine($"Il libro piu costoso: {Titoli[pos]} \t {prezzo[pos]} Eur");
17     Console.ReadKey();
18     return;
19 }
```



## 2.8 Reorder.cs

### 2.8.1 Step 1

Lo scopo di questa funzione è di ordinare i libri in base al prezzo. Viene usato un algoritmo di ordinamento [Bubble Sort](#). Si ha una scelta tra ordinamento crescente e decrescente.

Menu

```
1 Console.Clear();
2 Console.WriteLine("Scegli come riordinare");
3 Console.WriteLine("[1] Crescente \t [2] Decrescente");
4 int scelta;
5 try
6 {
7     scelta = int.Parse(Console.ReadLine());
8 }
9 catch (Exception e)
10 {
11     Console.WriteLine("Errore: Inserire un numero valido");
12     Console.ReadKey();
13     return;
14 }
15
16 if(scelta != 1 && scelta != 2)
17 {
18     Console.WriteLine("Errore: Inserire un numero valido");
19     Console.ReadKey();
20     return;
21 }
```

### 2.8.2 Step 2

Ordinamento crescente

```
1 if (scelta == 1)
2 {
3     for(int i = 0; i < prezzo.Length-1; i++)
4     {
5         for (int j = i + 1; j < prezzo.Length; j++)
6         {
7             if (prezzo[i] > prezzo[j])
8             {
9
10                 string temp = Titoli[i];
11                 Titoli[i] = Titoli[j];
12                 Titoli[j] = temp;
13
14                 temp = Autori[i];
15                 Autori[i] = Autori[j];
16                 Autori[j] = temp;
17
18                 double temp2 = prezzo[i];
19                 prezzo[i] = prezzo[j];
20                 prezzo[j] = temp2;
21
22                 temp = categoria[i];
23                 categoria[i] = categoria[j];
24                 categoria[j] = temp;
25
26                 temp = casa_editrice[i];
27                 casa_editrice[i] = casa_editrice[j];
28                 casa_editrice[j] = temp;
29
30                 int temp3 = copie[i];
31                 copie[i] = copie[j];
32                 copie[j] = temp3;
33             }
34         }
35     }
36 }
```

## Ordinamento Decrescente

```
1  if (scelta == 1)
2  {
3      for(int i = 0; i < prezzo.Length-1; i++)
4      {
5          for (int j = i + 1; j < prezzo.Length; j++)
6          {
7              if (prezzo[i] < prezzo[j])
8              {
9
10                 string temp = Titoli[i];
11                 Titoli[i] = Titoli[j];
12                 Titoli[j] = temp;
13
14                 temp = Autori[i];
15                 Autori[i] = Autori[j];
16                 Autori[j] = temp;
17
18                 double temp2 = prezzo[i];
19                 prezzo[i] = prezzo[j];
20                 prezzo[j] = temp2;
21
22                 temp = categoria[i];
23                 categoria[i] = categoria[j];
24                 categoria[j] = temp;
25
26                 temp = casa_editrice[i];
27                 casa_editrice[i] = casa_editrice[j];
28                 casa_editrice[j] = temp;
29
30                 int temp3 = copie[i];
31                 copie[i] = copie[j];
32                 copie[j] = temp3;
33             }
34         }
35     }
36 }
```

### 2.8.3 Step 3

#### Stampa dei libri ordinati

```
1  Console.WriteLine("Libri riordinati con successo. Premere un tasto per continuare");
2  Console.ReadKey();
3  Console.Clear();
4  Console.WriteLine("Indice - Titolo - Autore - Prezzo - Categoria - Casa Editrice - Copie");
5  for (int i = 0; i < Titoli.Length; i++)
6  {
7      Console.WriteLine($"{i+1} - {Titoli[i]} - {Autori[i]} - {prezzo[i]} - {categoria[i]} - {
8          casa_editrice[i]} - x{copie[i]}");
9  }
10 Console.ReadKey();
11 return;
```

## 2.9 PriceRange.cs

### 2.9.1 Step 1

Lo scopo di questa funzione è di visualizzare i libri in base ad una fascia di prezzo. Viene chiesto all'utente di inserire un prezzo minimo e massimo, e vengono stampati i libri in quella fascia.

```
1 public static void priceRange(string[] Titoli, double[] prezzo, int[] copie)
2 {
3     Console.Clear();
4     Console.WriteLine("Inserisci la fascia di prezzo");
5     double min, max;
6     try
7     {
8         Console.WriteLine("Minimo");
9         min = double.Parse(Console.ReadLine());
10        Console.WriteLine("Massimo");
11        max = double.Parse(Console.ReadLine());
12    }
13    catch (Exception e)
14    {
15        Console.WriteLine("Errore: Inserire un numero valido");
16        Console.ReadKey();
17        return;
18    }
19
20
21    if (min > max)
22    {
23        Console.WriteLine("Errore: Il minimo non puo essere maggiore del massimo");
24        Console.ReadKey();
25        return;
26    }
27
28    for(int i = 0; i < prezzo.Length; i++)
29    {
30        if (prezzo[i] >= min && prezzo[i] <= max)
31        {
32            Console.WriteLine($"[{i}] {Titoli[i]} \t {copie[i]} Eur \t x{copie[i]}");
33
34            Console.WriteLine();
35        }
36    }
37
38    Console.ReadKey();
39    return;
40
41 }
```

## 3 Prestiti

### 3.1 MenuPrestiti.cs

#### 3.1.1 Step 1

Lo scopo di questa funzione è di gestire i prestiti. Viene chiesto all'utente di scegliere tra 4 opzioni:

- Prendi in prestito
- Restituisci il prestito
- Visualizza i libri in prestito
- Esci

Viene tutto salvato negli array citati all'inizio. Essendo il menu uguale a quello principale, non verrà spiegato.

```
1 ConsoleKeyInfo key;
2 int pos = 1;
3 do
4 {
5     Console.Clear();
6     Console.WriteLine();
7     LogoPrestito.logoPrestito();
8     Menu(pos);
9     key = Console.ReadKey();
10    //Se premo freccia giu incremento la posizione quindi va all'opzione dopo
11    if (key.Key == ConsoleKey.DownArrow)
12    {
13        if (pos < 4)
14        {
15            pos++;
16        }
17        if(pos == 5)
18        {
19            pos = 1;
20        }
21    }
22    //Se premo freccia su decremento la posizione quindi va all'opzione prima
23    else if (key.Key == ConsoleKey.UpArrow)
24    {
25        if (pos > 1)
26        {
27            pos--;
28        }
29        if (pos == 0)
30        {
31            pos = 4;
32        }
33    }
34 } while (key.Key != ConsoleKey.Enter);
```

switch per la scelta dell'utente

```
1  switch (pos)
2  {
3      case 1:
4
5          TakeBorrow.takeBorrow(Titoli,copie, ref libri_prestito, ref prestiti, ref
            utenti_prestito_nome, ref utenti_prestito_cognome, ref giorno_preso, ref tempo_trattenuto,
            ref codice_prestito, ref returns);
6
7          break;
8
9      case 2:
10
11          ReturnBook.returnBook(ref libri_prestito,ref utenti_prestito_nome, ref
            utenti_prestito_cognome, ref giorno_preso, ref tempo_trattenuto, ref codice_prestito, ref
            returns);
12
13          break;
14
15      case 3:
16
17          ShowBorrowesBooks.showBorrowedBooks(libri_prestito, utenti_prestito_nome,
            utenti_prestito_cognome, giorno_preso, tempo_trattenuto, codice_prestito, prestiti);
18
19          break;
20
21      case 4:
22
23          Console.Clear();
24          Console.WriteLine("Premi per ritornare al menu principale");
25
26          break;
27
28  }
```

## 3.2 TakeBorrow.cs

### 3.2.1 Step 1

In questo primo step andiamo a rimuovere dall'array dei libri e copie tutti i libri già in prestito. Per poi controllare la presenza di libri o meno.

```
1  Console.Clear();
2  string[] strings = Titoli;
3  int[] ints = copie;
4
5
6  for (int i=0; i < libri_prestito.Length; i++)
7  {
8      for(int j = 0; j < strings.Length; j++)
9      {
10         if (libri_prestito[i] == null)
11         {
12             continue;
13         }
14         if (libri_prestito[i] == strings[j])
15         {
16             if (ints[j] > 1)
17             {
18                 ints[j]--;
19             }
20             else
21             {
22                 strings[j] = null;
23             }
24         }
25     }
26 }
27
28 //controllo se ci sono libri disponibili
29 bool flag = false;
30 for (int i = 0; i < strings.Length; i++)
31 {
32     if (strings[i] != null)
33     {
34         flag = true;
35         break;
36     }
37 }
38 if (!flag)
39 {
40     Console.WriteLine("Non ci sono libri disponibili");
41     Console.ReadKey();
42     return;
43 }
```

### 3.2.2 Step 2

Come step 2 l'utente seleziona tramite il menu quale libro prendere in prestito

```
1 //Stampo i libri disponibili e faccio il menu per il prestito
2 ConsoleKeyInfo key;
3 int pos = 0;
4 do
5 {
6
7     Menu(strings, pos);
8     key = Console.ReadKey();
9
10    //Freccie
11    if (key.Key == ConsoleKey.DownArrow)
12    {
13        pos++;
14        if (pos >= strings.Length)
15        {
16            pos = 0;
17        }
18        while (strings[pos] == null)
19        {
20            pos++;
21            if (pos >= strings.Length)
22            {
23                pos = 0;
24            }
25        }
26    }
27    if (key.Key == ConsoleKey.UpArrow)
28    {
29        pos--;
30        if (pos < 0)
31        {
32            pos = strings.Length - 1;
33        }
34        while(strings[pos] == null)
35        {
36            pos--;
37            if (pos < 0)
38            {
39                pos = strings.Length - 1;
40            }
41        }
42    }
43 }
44 } while (key.Key != ConsoleKey.Enter);
45
```

### 3.2.3 Step 3

Come step 3 l'utente inserisce il proprio nome, cognome e la data di restituzione, e il programma genera un codice prestito. Il codice infine stampa i dati completi del prestito.

```
1 //Aggiungo il libro preso in prestito
2 string nome, cognome, giorno_restituisce;
3
4 Console.WriteLine("Inserisci il tuo nome");
5 nome = Console.ReadLine();
6 Console.WriteLine("Inserisci il tuo cognome");
7 cognome = Console.ReadLine();
8 Console.WriteLine("Inserisci il giorno in cui devi restituire il libro, Inserisci GG/MM/AAAA");
9 giorno_restituisce = Console.ReadLine();
10
11
12
13 if (prestiti >= 2)
14 {
15     Array.Resize(ref libri_prestito, libri_prestito.Length + 1);
16     Array.Resize(ref utenti_prestito_nome, utenti_prestito_nome.Length + 1);
17     Array.Resize(ref utenti_prestito_cognome, utenti_prestito_cognome.Length + 1);
18     Array.Resize(ref giorni_preso, giorni_preso.Length + 1);
19     Array.Resize(ref tempo_trattenuto, tempo_trattenuto.Length + 1);
20     Array.Resize(ref codice_prestito, codice_prestito.Length + 1);
21
22
23 }
24 libri_prestito[prestiti - returns] = strings[pos ];
25 utenti_prestito_nome[prestiti - returns] = nome;
26 utenti_prestito_cognome[prestiti - returns] = cognome;
27 giorni_preso[prestiti - returns] = DateTime.Now.ToString("dd/MM/yyyy");
28 tempo_trattenuto[prestiti - returns] = giorno_restituisce;
29 codice_prestito[prestiti - returns] = prestiti;
30 prestiti++;
31
32 Console.WriteLine("Libro preso in prestito con successo");
33
34 Console.WriteLine("Libro : " + libri_prestito[prestiti - 1 - returns]);
35 Console.WriteLine("Utente : " + utenti_prestito_nome[prestiti - 1 - returns] + " " +
    utenti_prestito_cognome[prestiti - 1 - returns]);
36 Console.WriteLine("Giorno preso : " + giorni_preso[prestiti - 1 - returns]);
37 Console.WriteLine("Giorno da restituire : " + tempo_trattenuto[prestiti - 1 - returns]);
38 Console.WriteLine("Codice prestito : " + codice_prestito[prestiti - 1 - returns]);
39
40 Console.ReadKey();
41
42 return;
```



## 3.3 ReturnBook.cs

### 3.3.1 Step 1

In questo primo step l'utente inserisce il proprio codice e il programma controlla la sua validità.

```
1 Console.Clear();
2 Console.WriteLine("Inserisci il codice del prestito da restituire");
3 int codice;
4 try
5 {
6     codice = int.Parse(Console.ReadLine());
7 }
8 catch (Exception e)
9 {
10    Console.WriteLine("Errore: Inserire un numero");
11    return;
12 }
13
14 int pos;
15 //Flag codice non presente
16 bool flag = false;
17 for (pos = 0; pos < codice_prestito.Length; pos++)
18 {
19     if (codice_prestito[pos] == codice)
20     {
21         break;
22     }
23     if(pos == codice_prestito.Length - 1)
24     {
25         flag = true;
26     }
27 }
28
29 if(flag)
30 {
31     Console.WriteLine("Codice non presente");
32     Console.ReadKey();
33     return;
34 }
```

(Nota: I codici non sono stati descritti in quanto sono molto simili a quelli visti precedentemente)

### 3.3.2 Step 2

Step 2, stampare le informazioni del prestito e chiedere conferma.

```
1 //Prestito restituito si/no menu
2 ConsoleKeyInfo key;
3 int pos3 = 0;
4 do
5 {
6     Console.Clear();
7
8     string[] strings =
9     {
10         "Nome: " + utenti_prestito_nome[pos],
11         "Cognome: " + utenti_prestito_cognome[pos],
12         "Libro: " + libro_prestito[pos],
13         "Giorno preso: " + giorni_preso[pos],
14         "Tempo trattenuto: " + tempo_trattenuto[pos],
15         "Codice prestito: " + codice_prestito[pos],
16         "Confermi la restituzione? "
17     };
18
19     int consoleWidth = Console.WindowWidth;
20     for (int i = 0; i < strings.Length; i++)
21     {
22         int padding = (consoleWidth - strings[i].Length) / 2;
23         if (padding < 0)
24         {
25             padding = 0;
26         }
27         Console.WriteLine(new string(' ', padding) + $" {strings[i]}");
28     }
29     Menu(pos3);
30     key = Console.ReadKey();
31     if (key.Key == ConsoleKey.LeftArrow)
32     {
33         if (pos3 == 1)
34             pos3--;
35
36     }
37     else if (key.Key == ConsoleKey.RightArrow)
38     {
39         if (pos3 == 0)
40             pos3++;
41     }
42 } while (key.Key != ConsoleKey.Enter);
```

(Nota: I codici non sono stati descritti in quanto sono molto simili a quelli visti precedentemente)

### 3.3.3 Step 3

Data la conferma, il programma rimuove il prestito e stampa un messaggio di conferma.

```
1  if (pos3 == 1)
2  {
3      Console.WriteLine("Azione annullata");
4      Console.ResetColor();
5      Console.ReadKey();
6      return;
7  }
8  else
9  {
10     //Restituzione libro
11     libro_prestito[pos] = null;
12     utenti_prestito_nome[pos] = null;
13     utenti_prestito_cognome[pos] = null;
14     giorni_preso[pos] = null;
15     tempo_trattenuto[pos] = null;
16     codice_prestito[pos] = 0;
17
18     for (int i = pos; i < libro_prestito.Length - 1; i++)
19     {
20         libro_prestito[i] = libro_prestito[i + 1];
21         utenti_prestito_nome[i] = utenti_prestito_nome[i + 1];
22         utenti_prestito_cognome[i] = utenti_prestito_cognome[i + 1];
23         giorni_preso[i] = giorni_preso[i + 1];
24         tempo_trattenuto[i] = tempo_trattenuto[i + 1];
25         codice_prestito[i] = codice_prestito[i + 1];
26     }
27
28     Array.Resize(ref libro_prestito, libro_prestito.Length - 1);
29     Array.Resize(ref utenti_prestito_nome, utenti_prestito_nome.Length - 1);
30     Array.Resize(ref utenti_prestito_cognome, utenti_prestito_cognome.Length - 1);
31     Array.Resize(ref giorni_preso, giorni_preso.Length - 1);
32     Array.Resize(ref tempo_trattenuto, tempo_trattenuto.Length - 1);
33     Array.Resize(ref codice_prestito, codice_prestito.Length - 1);
34     returns++;
35     Console.WriteLine("Libro restituito con successo");
36 }
37
38 Console.ReadKey();
39 return;
```

(Nota: I codici non sono stati descritti in quanto sono molto simili a quelli visti precedentemente)

## 3.4 ShowBorrowedBooks.cs

### 3.4.1 Step 1

Lo scopo di questa funzione è di visualizzare i libri in prestito. Vengono stampati tutti i prestiti attivi al momento.

```
1 Console.Clear();
2 Console.WriteLine("Prestiti:");
3 for (int i = 0; i < prestiti; i++)
4 {
5     if(libri_prestito.Length <= i)
6     {
7         break;
8     }
9     Console.WriteLine("Codice prestito: " + codice_prestito[i]);
10    Console.WriteLine($"  \tLibro: {libri_prestito[i]}");
11    Console.WriteLine($"  \tUtente: {utenti_prestito_nome[i]} {utenti_prestito_cognome[i]}");
12    Console.WriteLine($"  \tGiorno preso: {giorni_preso[i]}");
13    Console.WriteLine($"  \tTempo trattenuto: {tempo_trattenuto[i]}");
14    Console.WriteLine();
15 }
16 Console.WriteLine("Premi un tasto per tornare al menu");
17 Console.ReadKey();
```

## 4 Utils e Logo

### 4.1 ToLower.cs

Una funzione già esistente in C# che trasforma una stringa in minuscolo, ma per alcuni bug ne abbiamo creato una noi.

```
1 public static string ToLowerString(string str)
2 {
3     //Converte la stringa in minuscolo
4     string lower = "";
5     for(int i = 0; i < str.Length; i++)
6     {
7         //Le lettere maiuscole vengono convertite in minuscole
8         if (str[i] >= 'A' && str[i] <= 'Z')
9         {
10             lower += str[i].ToString().ToLower();
11         }
12         else
13         {
14             lower += str[i];
15         }
16     }
17     return lower;
18 }
19 }
```

### 4.2 LogoBiblioteca.cs — LogoPrestito.cs

Logo della biblioteca e Prestito fatte tramite un sito TopSecret

(Nota: Il codice dei loghi sono reperibili su github!)



Figura 1: Logo biblioteca.



Figura 2: Logo Prestito.

## 5 Github

Per questo progetto abbiamo deciso di utilizzare github per la condivisione del progetto, abitando tra noi distanti, (Io e Giovanni) ciò ci ha permesso di continuare il codice anche a distanza

### 5.1 Come funziona?