［初期化処理］（空ポインタを-1 とする）

```
public struct ElementData
{   public long 番号;  public string 氏名; public int 点数;
}
public struct ElementSet
{   public ElementData Element; public long Next;
}
public ElementSet[] DataArea = new ElementSet[500];
public long ErasedP;  //  未使用領域の先頭領域
private void 初期化( )
{   for(int i=0;i<DataArea.Length-1;i++) DataArea[i].Next=i+1;
    DataArea[DataArea.Length - 1].Next = -1;
    ErasedP=0;
}
```

(3)領域確保

領域確保では，現在の未使用ポインタが指している要素を使うものとします。すなわち，以下の手順で処理します。

ⅰ)未使用ポインタを関数値とする。

ⅱ)未使用ポインタが指しているポインタを未使用ポインタとする。

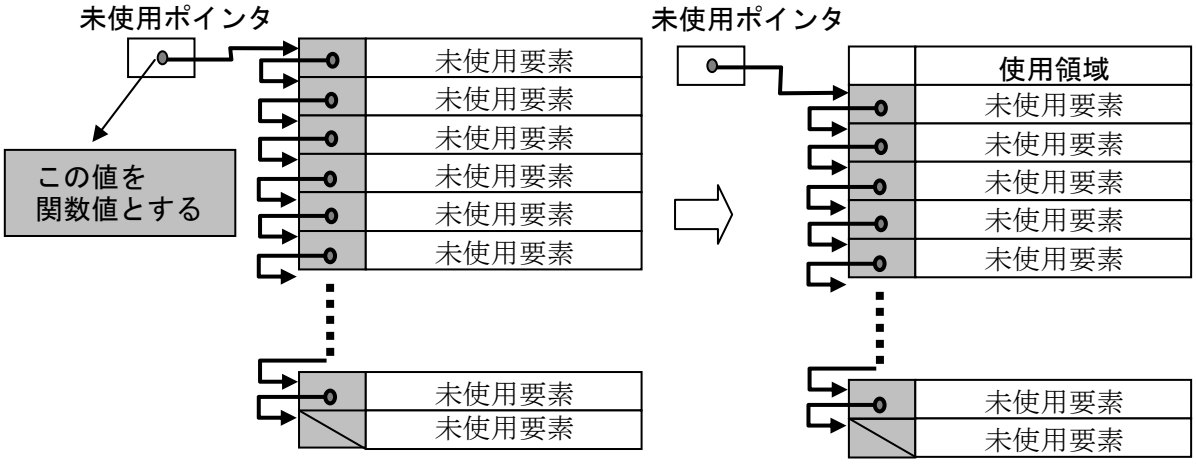

図 3-4　領域確保の処理

[領域確保の処理]

```
private long GetArea()
{    long R = ErasedP;
     ErasedP = DataArea[ErasedP].Next;
     return R;
}
```