namespace LinkedList;

interface

uses

System.Text;

type

List<T> = public class

where T is Object;

private

method AppendToString(aBuilder: StringBuilder);

public

constructor(aData: T);

constructor(aData: T; aNext: List<T>);

property Next: List<T>;

property Data: T;

method ToString: string; override;

end;

implementation

constructor List<T>(aData: T);

begin

Data := aData;

end;

constructor List<T>(aData: T; aNext: List<T>);

begin

constructor(aData);

Next := aNext;

end;

method List<T>.ToString: string;

begin

with lBuilder := new StringBuilder do begin

AppendToString(lBuilder);

result := lBuilder.ToString();

end;

end;

method List<T>.AppendToString(aBuilder: StringBuilder);

begin

if assigned(Data) then

aBuilder.Append(Data.ToString)

else

aBuilder.Append('nil');

if assigned(Next) then begin

aBuilder.Append(', ');

Next.AppendToString(aBuilder);

end;

end;

end.