*//----------------Perfomance test with Integer---------------------*

*//-------------Appends element to the end of list--------------------*

20070 JavaList add time

1733 MyList add time

*//----------------------Inserts element in the list------------------*

161998 JavaList insert time

335359 MyList insert time

*//---------------------Search element in the list--------------------*

1475934 JavaList search time

1195601 MyList search time

*//---------------------Remove element from list----------------------*

1296766 JavaList remove time

3310295 MyList remove time

*//----------------perfomance test with MyTriangle--------------------*

*//-------------Appends element to the end of list--------------------*

20963 JavaList add time

13483 MyList add time

*//----------------------Inserts element in the list------------------*

69226 JavaList insert time

77329 MyList insert time

*//---------------------Search element in the list--------------------*

1543070 JavaList search time

1373865 MyList search time

*//---------------------Remove element from list----------------------*

1805605 JavaList remove time

4099625 MyList remove time

Вывод: так как в двусвязном списке есть возможность идти с конца, поиск в нем осуществляется быстрее, а соответственно удаление и вставка в середину происходит быстрее, что видно из полученных результатов. Вставка следующего узла(вставка в конец) осуществляется быстрее в односвязном списке, так как нет необходимости в указании ссылки на предыдущий элемент.