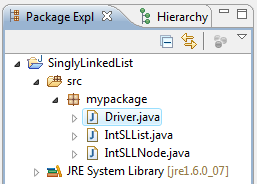
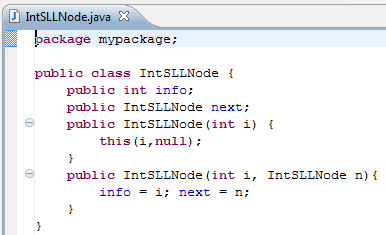
***Hands-On Exercise 3.2 [20-points]: Singly Linked List***

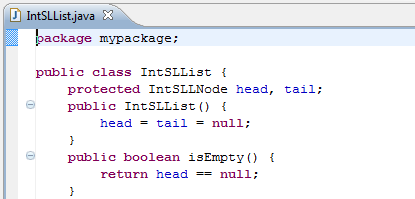
***IDE structure:***

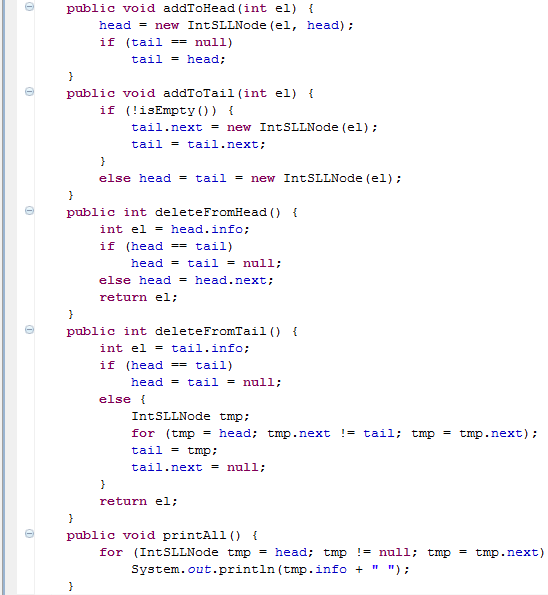


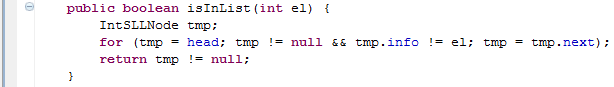
1.❑ Add an IntSLLNode class and type the code below

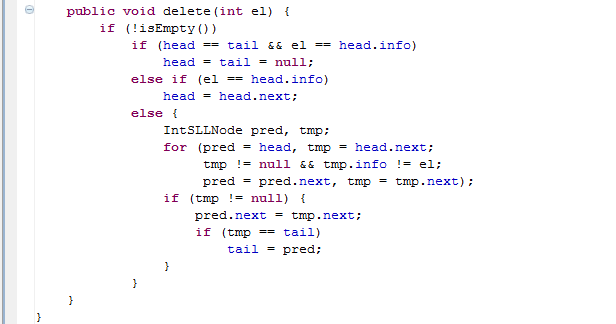


2.❑ Add an IntSLList class and type the code below

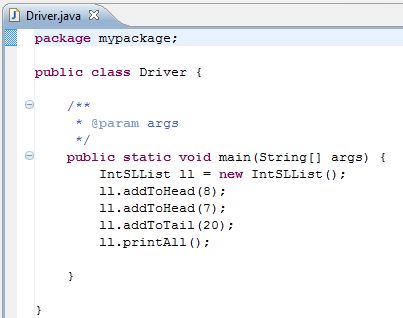








3.❑ Add Driver class and type the code below



4.❑ Paste your code here.

**package** inclass;

**public** **class** IntSLLNode {

**public** **int** info;

**public** IntSLLNode next;

**public** IntSLLNode(**int** i){

**this**(i, **null**);

}

**public** IntSLLNode(**int** i, IntSLLNode n){

info = i;

next = n;

}

**package** inclass;

**public** **class** IntSLList {

**protected** IntSLLNode head, tail;

**public** IntSLList(){

head = tail = **null**;

}

**public** **boolean** isEmpty(){

**return** head == **null**;

}

**public** **void** addToHead(**int** el){

head = **new** IntSLLNode(el, head);

**if**(tail == **null**){

tail = head;

}

}

**public** **void** addToTail(**int** el){

**if**(!isEmpty()){

tail.next = **new** IntSLLNode(el);

tail = tail.next;

}

**else**{

head = tail = **new** IntSLLNode(el);

}

}

**public** **int** deleteFromHead(){

**int** el=head.info;

**if**(head == tail){

head = tail = **null**;

}

**else**{

head = head.next;

}

**return** el;

}

**public** **int** deleteFromTail(){

**int** el = tail.info;

**if**(head == tail){

head = tail = **null**;

}

**else**{

IntSLLNode tmp;

**for**(tmp = head; tmp.next != tail; tmp = tmp.next);

tail = tmp;

tail.next = **null**;

}

**return** el;

}

**public** **void** printAll(){

**for**(IntSLLNode tmp = head; tmp != **null**; tmp = tmp.next){

System.***out***.println(tmp.info + " ");

}

}

**public** **boolean** isInList(**int** el){

IntSLLNode tmp;

**for**(tmp = head; tmp != **null** && tmp.info != el; tmp = tmp.next);

**return** tmp != **null**;

}

**public** **void** delete(**int** el){

**if**(!isEmpty())

**if**(head == tail && el == head.info)

head = tail = **null**;

**else** **if** (el == head.info)

head = head.next;

**else**{

IntSLLNode pred, tmp;

**for**(pred = head, tmp = head.next;

tmp != **null** && tmp.info != el;

pred = pred.next, tmp = tmp.next);

**if**(tmp != **null**){

pred.next = tmp.next;

**if**(tmp == tail)

tail = pred;

}

}

}

}

**package** inclass;

**public** **class** Driver {

**public** **static** **void** main(String[] args){

IntSLList li = **new** IntSLList();

li.addToHead(8);

li.addToHead(7);

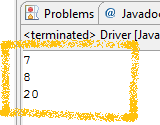
li.addToTail(20);

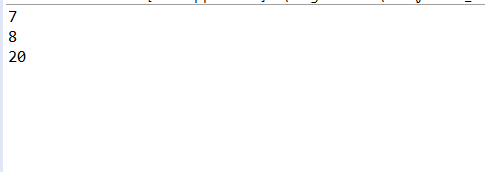
li.printAll();

}

}

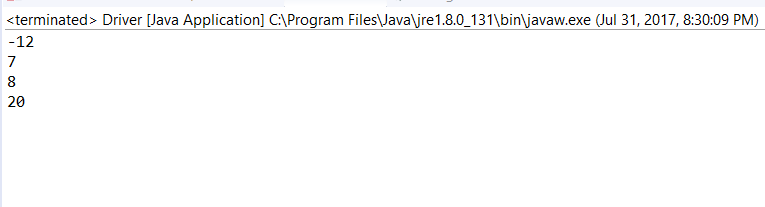
5.❑ Paste your screen shot output here [Ctrl] + [PrtScn]. Make sure you magnified it.

Similar to this output



6.❑ Add another node with int value of -12, paste below the code that you added and paste the new output.

li.addToHead(-12);



7.❑ Write your topmost question regarding this topic.

8.❑ **Critical Thinking:** If you are asked to make a test question based on this topic, what would be the question and what is your answer?

**Submission Procedure**

1. Write your **name** here: Antonio Isabella
2. Date: 7/31/17
3. **Backup** your work to your USB drive, this material may come out as part of your exam.
4. **Submit** to Blackboard at the link where you got it.

**Note:**

* Submit back to Blackboard where you get it.
* 2-points deduction if you submit it on the wrong place.
* 2-points deduction if you did not follow these instructions.
* Make sure you submit it at the correct location where you got it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GRADING RUBRIC | | | | |
| Grading Criteria | 3  **Exceeds**  *Excellent*  Epic Wow | 2  **Meets**  *Satisfactory*  O.K. | 1  **Partially Meets**  *Below Expectations*  Not Yet | 0  **Does Not Meet**  *Unacceptable*  Fail |
| **Completeness** | +5-Completed all the required work and added more examples. | +2-Completed all the work required. | +1-Partially completed the work required. | Unfortunately, did not complete the work required. |
| **Coding** | +10- Code is excellent, comments are added, and different techniques were used. | +7-Code is O.K., and program works. | +4-Code works, but still needs improvement. | Unfortunately, no coding. |
| **Output** | +5-Outputs are correct, and provided additional output cases. | +2-Output meets requirement and is readable. | +1-There is output, but not readable, and/or needs improvement. | Unfortunately, no output. |
| **Late** | Excellent, you submitted it before the deadline. | -5, unfortunately for submitting after the deadline. | -7, unfortunately for submitting several weeks after the deadline. | -10, unfortunately, for submitting very late. |