// Dalton Wright

// CSCI 301 01

// Project #6

// UML Charts

|  |
| --- |
| Class Event |
| -eventType: char  -eventArrivalTime: int  -eventTransactionTime: int |
| +Event();  +Event(const char& type, const int& time);  +Event(const char& type, cons tint& time, cons tint& length);  +getEventTime(): int  +getEventType(): char  +getEventLength(): int  +setEventTime(): void  +setEventType(): void  +setEventLength(): void |

|  |
| --- |
| Class LinkedSortedList |
| -headPtr: Node<ItemType>\*  -itemCount: int  -getNodeBefore(const ItemType& anEntry) const: Node<ItemType>\*  -getNodeAt(int position) const: Node<ItemType>\*  -copyChain(const Node<ItemType>\* origChainPtr): Node<ItemType>\* |
| +LinkedSortedList();  +LinkedSortedList(const LinkedSortedList<ItemType>& aList);  +virtual ~LinkedSortedList();  +insertSorted(const ItemType& newEntry): void  +removeSorted(const ItemType& anEntry): bool  +getPosition(const ItemType& newEntry) const: int  +isEmpty() const: bool  +getLength() const: int  +remove(int position): bool  +clear(): void  +getEntry(innt position): ItemType |

|  |
| --- |
| Class SL\_PriorityQueue |
| -slistPtr: LinkedSortedList<ItemType>\* |
| +SL\_PriorityQueue();  +SL\_PriorityQueue(const SL\_PriorityQueue& pq);  +~SL\_PriorityQueue();  +isEmpty() const: bool  +add(const ItemType& newEntry): bool  +remove(): bool  +peek(): ItemType |
| Class PriorityQueueInterface |
|  |
| +virtual isEmpty() const=0: bool  +virtual add(const ItemType& newEntry)=0: bool  +virtual remove()=0: bool  +virtual peek() const= 0: ItemType |

|  |
| --- |
| Class Node |
| -nodeItem: ItemType  -previous: Node<ItemType>\*  -next: Node<ItemType>\* |
| +Node();  +Node(const ItemType& anItem);  +Node(const ItemType& anItem, Node<ItemType>\* nextNodePtr);  +Node(const ItemType& anItem, Node<ItemType>\* nextNodePtr, Node<ItemType>\* prevNodePtr);  +setItem(const ItemType& anItem): void  +setNext(Node<ItemType>\* nextNodePtr): void  +setPrev(Node<ItemType>\* prevNodePtr): void  +getItem() const: ItemType  +getNext() const: Node<ItemType>\*  +getPrev() const: Node<ItemType>\* |

|  |
| --- |
| Class PrecondViolatedExcep : public logic\_error |
|  |
| +PrecondViolatedExcep(const string& message = “ “); |