Assignment . 13

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

template <typename T>

class PriorityItem {

public:

T data;

int priority;

PriorityItem(T d, int p) : data(d), priority(p) {}

bool operator<=(const PriorityItem& other) const {

return priority <= other.priority;

}

};

template <typename T>

class PriorityQueue {

vector<PriorityItem<T>> queue;

public:

void insert(T data, int priority) {

PriorityItem<T> newItem(data, priority);

queue.push\_back(newItem);

sort(queue.begin(), queue.end(), [](const PriorityItem<T>& a, const PriorityItem<T>& b) {

return a.priority > b.priority;

});

}

void deleteHighestPriority() {

if (queue.empty()) {

cout << "Queue is empty." << endl;

return;

}

cout << "Deleted item with highest priority: " << queue.front().data << endl;

queue.erase(queue.begin());

}

void displayQueue() {

if (queue.empty()) {

cout << "Queue is empty." << endl;

return;

}

cout << "Priority Queue: ";

for (const auto& item : queue) {

cout << "(" << item.data << ", Priority: " << item.priority << ") ";

}

cout << endl;

}

};

int main() {

PriorityQueue<string> pq;

int choice, priority;

string data;

do {

cout << "1. Insert Item\n2. Delete Highest Priority Item\n3. Display Queue\n4. Exit\n";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter data and priority: ";

cin >> data >> priority;

pq.insert(data, priority);

break;

case 2:

pq.deleteHighestPriority();

break;

case 3:

pq.displayQueue();

break;

}

} while (choice != 4);

return 0;

}

1. Insert Item

2. Delete Highest Priority Item

3. Display Queue

4. Exit

Enter data and priority: Task1 3

1. Insert Item

2. Delete Highest Priority Item

3. Display Queue

4. Exit

Enter data and priority: Task2 5

1. Insert Item

2. Delete Highest Priority Item

3. Display Queue

4. Exit

Enter data and priority: Task3 1

3

Priority Queue: (Task2, Priority: 5) (Task1, Priority: 3) (Task3, Priority: 1)

2

Deleted item with highest priority: Task2

3

Priority Queue: (Task1, Priority: 3) (Task3, Priority: 1)

1. Insert Item

2. Delete Highest Priority Item

3. Display Queue

4. Exit

Enter data and priority: Task4 4

3

Priority Queue: (Task1, Priority: 3) (Task4, Priority: 4) (Task3, Priority: 1)

2

Deleted item with highest priority: Task4

3

Priority Queue: (Task1, Priority: 3) (Task3, Priority: 1)

4s