**HW6**

**Atanas Dimitrov (100114943)**

bool searchOptions(string producer, string model, string searchPart, int price, int replacement, vector<Spare\_part\*>& sparePart, vector <Spare\_part\*>& subst, bool result)

{

stack<string> myStack;

stack<string> search;

string producers[200];

string models[200];

string parts[200];

bool ASP[2];

int positionProducer = 0, positionModel = 0, positionPart = 0, positionASP = 0;

bool repeat2 = false;

bool available = false;

bool found = false;

for (int i = 0; i < sparePart.size(); i++)

{

for (int n = 0; n < 200; n++)

{

if (sparePart.at(i)->getProducer() == producers[n])

repeat2 = true;

}

if (repeat2 == false)

{

producers[positionProducer] = sparePart.at(i)->getProducer();

myStack.push(sparePart.at(i)->getProducer());

positionProducer++;

}

repeat2 = false;

}

while (!myStack.empty())

{

if (myStack.top() == producer)

{

myStack.pop();

for (int i = 0; i < sparePart.size(); i++)

{

for (int n = 0; n < 200; n++)

{

if (sparePart.at(i)->getModel() == models[n])

repeat2 = true;

}

if (repeat2 == false && sparePart.at(i)->getProducer() == producer)

{

models[positionModel] = sparePart.at(i)->getModel();

myStack.push(sparePart.at(i)->getModel());

positionModel++;

}

repeat2 = false;

while (positionModel != 0)

{

if (myStack.top() == model)

{

myStack.pop();

for (int i = 0; i < sparePart.size(); i++)

{

for (int n = 0; n < 200; n++)

{

if (sparePart.at(i)->getName() == parts[n])

repeat2 = true;

}

if (repeat2 == false && sparePart.at(i)->getProducer() == producer && sparePart.at(i)->getModel() == model)

{

parts[positionPart] = sparePart.at(i)->getName();

myStack.push(sparePart.at(i)->getName());

positionPart++;

}

repeat2 = false;

}

while (positionPart != 0)

{

if (myStack.top() == searchPart)

{

myStack.pop();

for (int i = 0; i < sparePart.size(); i++)

{

if (sparePart.at(i)->getQuantity()>0)

{

available = true;

break;

}

}

if (available)

{

for (int i = 0; i < sparePart.size(); i++)

{

if (sparePart.at(i)->getProducer() == producer && sparePart.at(i)->getModel() == model && sparePart.at(i)->getName() == searchPart && sparePart.at(i)->getPrice()<price && sparePart.at(i)->getReplacement()<replacement)

{

cout << "\n" << producer << setw(15) << model << setw(15) << searchPart << setw(15) << sparePart.at(i)->getPrice() << setw(5) << sparePart.at(i)->getReplacement();

result = true;

}

}

found = true;

}

}

else

{

myStack.pop();

}

positionPart--;

}

}

else

{

myStack.pop();

}

positionModel--;

}

}

}

else

{

myStack.pop();

}

}

if (!found)

{

for (int i = 0; i < subst.size(); i++)

{

if (subst.at(i)->getProducer2() == producer && subst.at(i)->getModel2() == model && subst.at(i)->getName() == searchPart)

{

for (int j = 0; j < sparePart.size(); j++)

{

if (sparePart.at(j)->getProducer() == subst.at(i)->getProducer1() && sparePart.at(j)->getModel() == subst.at(i)->getModel1() && sparePart.at(j)->getName() == subst.at(i)->getName() && sparePart.at(j)->getQuantity()>0)

{

result=searchOptions(subst.at(i)->getProducer1(), subst.at(i)->getModel1(), searchPart, price, replacement, sparePart, subst, result);

}

}

}

}

}

if (!result)

{

cout << "\nNothing found.";

}

return result;

}

int main()

{

……………

case 12:

myStack.empty();

system("CLS");

if (sparePart.empty())

{

cout << "\nThere are no spare parts.";

Sleep(2000);

break;

}

header();

cout << " Search repair option";

cout << "\n";

cout << "Producer: ";

getline(cin, producer);

cout << "\nModel: ";

getline(cin, model);

cout << "\nSpare part: ";

getline(cin, searchPart);

do

{

cout << "\nPrice: ";

getline(cin, priceS);

price = atof(priceS.c\_str());

if (price <= 0)

{

cout << "\nWrong input. The price should be positive number.\n";

}

} while (price <= 0);

do

{

cout << "\nReplacement: ";

getline(cin, replacementS);

replacement = atoi(replacementS.c\_str());

if (replacement<1)

{

cout << "\nWrong input. The replacement price should be positive number.\n";

}

} while (replacement <1);

cout << "\nProducer" << setw(15) << "Model" << setw(15) << "Part" << setw(15) << "Price" << setw(5) << "Time";

searchOptions(producer, model, searchPart, price, replacement, sparePart, subst, false);

cout << "\n\nEnter 0 to go back to the menu\n";

do

{

getline(cin, exit);

} while (exit != "0");

break;

#pragma endregion

case 0:

break;

}

} while (choice != 0);

return 0;

}