|  |
| --- |
| #include<iostream>  #include<string>  #include<list>  #include<fstream>  class Question{  public:          Question();          Question(std::string *q*,std::string *a*);          std::string m\_getQuery();          std::string m\_getAnswer();          ~Question();  private:      std::string query;      std::string answer;};  Question::Question(){      query = "";      answer = "";}  Question::Question(std::string q,std::string a){      query = q;  *this*->answer = a;}  std::string Question::m\_getQuery(){      return query;}  std::string Question::m\_getAnswer(){      return answer;}  Question::~Question()  {}  class chat{  public:          chat();          bool isQuestion(std::string *input*);          bool check(std::string *input*);          void reply(std::string *input*);          void storeNewQuestion(std::string *input*);          ~chat();  private:          std::list<Question> questionList;};  chat::chat(){      std::ifstream inputFile;      inputFile.open("QA.txt");      std::string tempQ;      std::string tempA;      while(!inputFile.eof()){          std::getline(inputFile,tempQ, ';');          std::getline(inputFile,tempA);          Question question(tempQ,tempA);          questionList.push\_back(question);}}  bool chat::isQuestion(std::string *input*){      if(input.empty()){          return false;}      if(input.back() == '?'){          return true;}      else if(input.back() != '?'){          return false;}      return false;}  bool chat::check(std::string *input*){      std::list<Question>::iterator it;      it = questionList.begin();      for(it; it != questionList.end(); it++){          if(input == it->m\_getQuery()){              return true;}} return false;}  void chat::reply(std::string *input*){      std::list<Question>::iterator it;      it = questionList.begin();      for (it; it != questionList.end(); it++){          if (input == it->m\_getQuery()){              std::cout << it->m\_getAnswer() << std::endl;}}}  void chat::storeNewQuestion(std::string *input*){      std::string tmp;      std::cout << "Hmmm . . . that's a good question! Maybe you could help me? " << std::endl << input << std::endl;      std::getline(std::cin, tmp);      Question new\_query(input, tmp);      questionList.push\_back(new\_query);      std::ofstream outputFile;      outputFile.open("QA.txt", std::ios::app);      if (outputFile.is\_open()){          outputFile << std::endl << input << ";" << tmp;}      else{          std::cout << "Error, file didn't open." << std::endl;}      outputFile.close();}  chat::~chat()  {}  int main(int *argc*,char\* *argv*[]){      chat chat;      bool loop = true;      std::string input;      while (loop){          std::cout<<"Hello ! You Can Ask Me Any Question"<<std::endl;          std::getline(std::cin,input);          for(int i = 0;i != input.length(); i++)          {input.at(i) = tolower(input.at(i));}          if(chat.isQuestion(input) == true){              if(chat.check(input) == true)              {chat.reply(input);}              else              {chat.storeNewQuestion(input);}}          else{              std::cout<<"Wrong Question Format !!! End Question with ?"<<std::endl;}  *//system("pause");*  *//system("cls");*}  *//system("pause");*      return 0;} |

Text

Description automatically generated