**ҚАЗАҚСТАН РЕСПУБЛИКАСЫНЫҢ БІЛІМ ЖӘНЕ ҒЫЛЫМ МИНИСТРЛІГІ**

«М.Х. ДУЛАТИ АТЫНДАҒЫ ТАРАЗ ӨҢІРЛІК УНИВЕРСИТЕТ»

КОММЕРЦИЯЛЫҚ ЕМЕС АКЦИОНЕРЛІК ҚОҒАМЫ

Технологиялық **факультеті**

Қолданбалы информатика және бағдарламалау **кафедрасы**

Оқу практика бойынша есебі

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тобы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/аты-жөні/ /қолы/

Жетекші **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

/қызметі/ /аты-жөні/ /қолы/

Қорғалды «\_\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_ж. бағасы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/жазбаша/

Комиссия мүшелері: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/аты-жөні/ /қолы/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/аты-жөні/ /қолы/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/аты-жөні/ /қолы/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

/аты-жөні/ /қолы/

Өндіріс жетекшісі **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

/қызметі, аты-жөні/ /қолы/

Өндіріс М.О.

##### Тараз 2024

**Мазмұны**

Кіріспе………………………………………………………………………….......3

Теориялық бөлім …………...…………………………………………..….4

Бағдарламада қолданылған С++ тілінің технологиялары…......................……………...….......................5

Жеке жұмыс.........................................................................6

**КІРІСПЕ**

***Менің тәжірибелік жұмысымның тақырыбы С++ тілінде кеңінен қолданылатын технологиялармен танысу***

*Жұмыстың өзектілігі* : Технологиялық прогрестің күннен күнге дамып келе жатқан біздің заманымызда білімді бағалау өте өзекті болып табылады. Күн сайын әлемде тез әрі тиімді бағалауды қажет ететін ақпараттар пайда болып жатыр. Ең қолжетімді бағалау әдістерінің ішінде тестілеуге арналған программалар танымалы және ыңғайлылардың бірі саналады. Бұл бағдарлама әртүрлі салаларда қолданылады және білімді тиімді бақылап қана қоймай сонымен қатар тесттерді пайдаланушылардың әртүрлі қажеттіліктермен мақсаттарына қарай бейімдеу мүмкіндігін береді. Бұл құралдың арқасында мұғалімдер студенттердің білім деңгейін дәлірек бағалай алады, ал студенттер емтиханға тиімдірек дайындалып, үлгерімін өз бетінше бақылай алады және HR мамандары кандидаттардың білімімен дағдыларын бағалай отырып кадрларды сапалы және объективті іріктеу жүргізе алады.

*Жұмыстың мақсаты*:

* Бірнеше жауабы бар таңдау тесттерін әзірлеу және басқару үшін бағдарламалық жасақтаманы құру.
* Әртүрлі пайдаланушы қажеттіліктеріне сәйкес жүйенің икемділігін қамтамасыз ету.
* Тесттерді жасау және тапсыру үшін ыңғайлы және интуитивті интерфейсті қамтамасыз ету.

*Міндеттері*:

1. Берілген техникалық талаптар негізінде бағдарлама кодын әзірлеу.
2. Бағдарламаның сенімді жұмысын қамтамасыз ету үшін тестілеу және дебаггинг жасау .
3. Бағдарламаны қолдау және қажет болған жағдайда өзгерістер мен жақсартулар енгізу.
4. Жасалған бағдарламалық өнімді тестілеу

Теориялық бөлім

1. std::vector — элементтердің динамикалық массивін қамтамасыз ететін C++ стандартты кітапханасындағы контейнерлердің бірі. Кәдімгі массивтерден айырмашылығы, std::veсtor бағдарламаның орындалу барысында өзінің өлшемін динамикалық түрде өзгерте алады. Ол массивтің соңындағы элементтерді қосуды және жоюды, сонымен қатар элементтерге индекс бойынша қол жеткізуді қамтамасыз етеді.
2. std::map — элементтерді кілт-мән жұбы түрінде сақтайтын контейнер. Әрбір кілт std::map ішінде бірегей болып табылады, бұл кілт арқылы мәнді жылдам табуға мүмкіндік береді. std::map ағаш деректер құрылымын (әдетте қызыл-қара ағаш) пайдалану арқылы элементтерге жылдам қол жеткізуді қамтамасыз етеді.
3. std::random\_device — кездейсоқ сандар генераторларына (бар болса) қол жеткізуді қамтамасыз ететін кездейсоқ сандар генераторы. Ол көбінесе генераторларды инициализациялау үшін пайдаланылатын ерекше кілттерді жасау үшін кездейсоқ сандар генераторларымен бірге қолданылады.
4. OOP (Object Oriented Programming): OOP - бұл объектілер мен класстар тұжырымдамаларына негізделген бағдарламалау әдістемесі. Объектілер – бұл деректермен (өрістер) және осы деректермен өзара әрекеттесуді қамтамасыз ететін әдістерді (функцияларды) қамтуы мүмкін класстар. Код OOP негізгі принциптерін көрсететін Answer, MultipleChoiceQuiz және Exam класстарын пайдаланады.
5. Ағынды басқару бағдарламада нұсқаулардың орындалу ретін басқару әдістерін білдіреді. Код шарттар негізінде орындалу ағынын басқару үшін циклдарды, шартты if-else операторларын және exit() пәрмендерін пайдаланады.
6. Интернационалдандыру және локализация: Интернационалдандыру (i18n) және локализациялау (l10n) әртүрлі тілдер мен тілдерді қолдауға арналған бағдарламалық жасақтаманы әзірлеу процестері болып табылады. Код әртүрлі тілдер үшін интерфейс пен мәтінді локализациялауды қамтамасыз ету үшін std::map құрылымын қолданатын аударма механизмін пайдаланады.
7. Көрсеткіштер мен сілтемелерді пайдалану: Көрсеткіштер мен сілтемелер жадты өңдеуге және функциялар мен нысандар арасында деректерді беруге мүмкіндік береді. Олар деректердің үлкен көлемін тиімді тасымалдау және қажетсіз көшіруді болдырмау үшін қолданылады.
8. Қателерді өңдеу: бұл бағдарламадағы күтпеген жағдайларды өңдеу тәсілдері. Код std::cerr көмегімен енгізу қателерін және қате туралы хабарларды шығару үшін std::cin.fail() сияқты ерекше жағдайларды өңдеу механизмдерін пайдаланады.

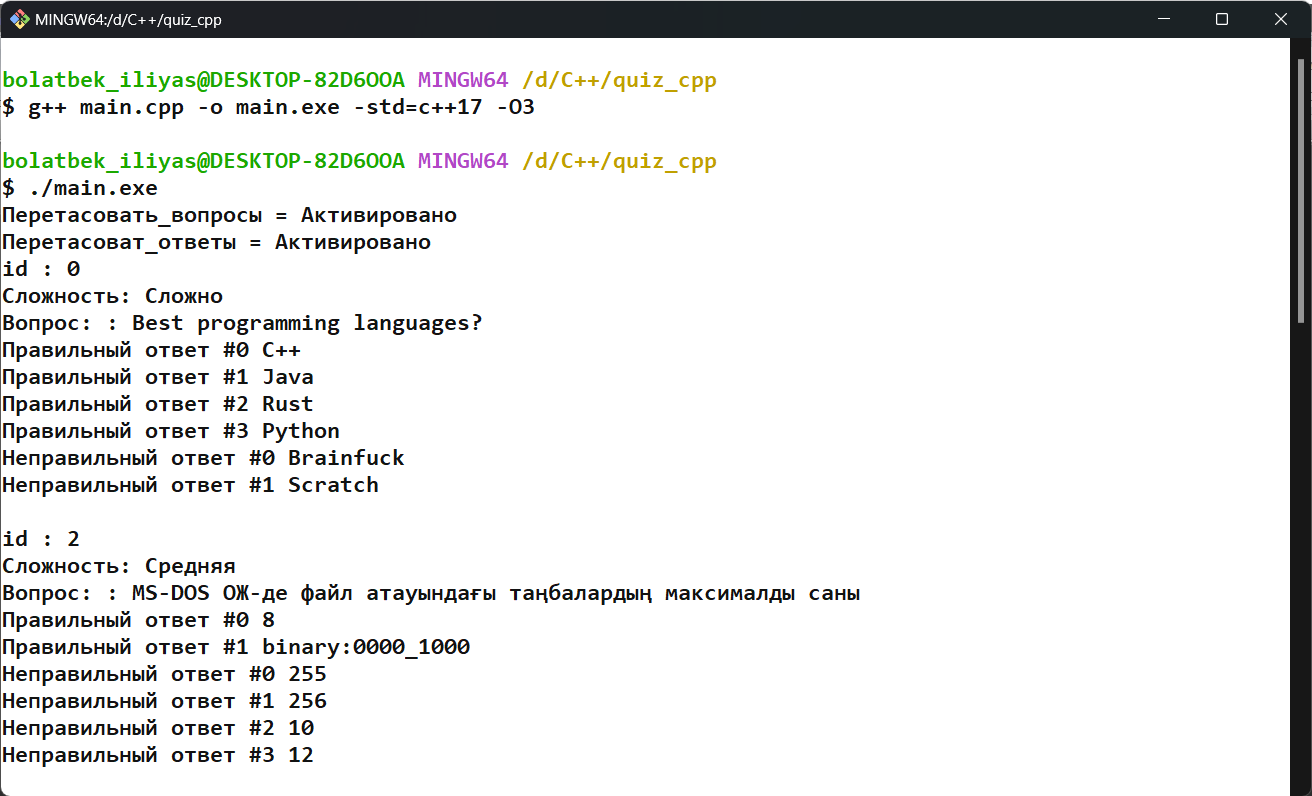
Бұл тұжырымдамалар мен технологиялар бірнеше тілдер мен параметрлерді қолдайтын қуатты және икемді тестілеу қосымшасын жасау үшін бірге жұмыс істейді.

*Жеке жұмыс :* ***Бір немесе біренеше дұрыс жауабы бар тестілеуге арналған программа жазу.***

|  |
| --- |
| **#include <iostream>**  **#include <vector>**  **#include <map>**  **#include <algorithm>**  **#include <set>**  **#include <random>**    **enum Languages**  **{**  **KZ,**  **RU,**  **EN,**  **};**    **enum Translate**  **{**  **USER\_ANSWER\_NUM,**  **WRONG\_INDEX,**  **ERROR\_WHILE\_CHECKING\_ID,**  **CFG\_SHUFFLE\_QUESTION,**  **CFG\_SHUFFLE\_ANSWERS,**  **COMPLEXITY\_EASY,**  **COMPLEXITY\_MEDIUM,**  **COMPLEXITY\_DIFFICULT,**  **COMPLEXITY\_NOT\_SET,**  **QUESTION,**  **CORRECT\_ANSWER,**  **CORRECT\_ANSWERS,**  **WRONG\_ANSWER,**  **CORRECT\_ANSWER\_COUNT,**  **YOUR\_ANSWERS,**  **YOUR\_SCORE,**  **TYPE\_C\_OR\_Q\_TO,**  **BOOL\_TRUE,**  **BOOL\_FALSE,**  **};**    **// Configs start**    **const bool SHUFFLE\_ANSWERS = true;**  **const bool SHUFFLE\_QUESTIONS = true;**  **const Languages UI\_LANG = RU;**    **std::map<Languages, std::map<Translate, std::string>> translate = {**  **{**  **Languages::KZ,**  **std::map<Translate, std::string>{**  **{Translate::USER\_ANSWER\_NUM, "Жауап индексін таңдаңыз # "},**  **{Translate::WRONG\_INDEX, "Индекс жарамсыз. Қайталап көріңіз!"},**  **{Translate::ERROR\_WHILE\_CHECKING\_ID, "Қате: Викторина идентификаторларын тексеріңіз!"},**  **{Translate::CFG\_SHUFFLE\_QUESTION, "Сұрақтарды араластыру = "},**  **{Translate::CFG\_SHUFFLE\_ANSWERS, "Жауаптарды араластыру = "},**  **{Translate::COMPLEXITY\_EASY, "Күрделелігі : Oңай"},**  **{Translate::COMPLEXITY\_MEDIUM, "Күрделелігі : Орташа"},**  **{Translate::COMPLEXITY\_DIFFICULT, "Күрделелігі : Қиын"},**  **{Translate::COMPLEXITY\_NOT\_SET, "The Күрделелігі орнатылмаған"},**  **{Translate::QUESTION, "Сұрақ: "},**  **{Translate::CORRECT\_ANSWER, "Дұрыс жауап #"},**  **{Translate::WRONG\_ANSWER, "Қате жауап #"},**  **{Translate::CORRECT\_ANSWER\_COUNT, "Дұрыс жауаптар саны: "},**  **{Translate::YOUR\_ANSWERS, "Сіздің жауаптарыңыз: "},**  **{Translate::CORRECT\_ANSWERS, "Дұрыс жауаптар: "},**  **{Translate::YOUR\_SCORE, "Сіздің ұпайыңыз: "},**  **{Translate::TYPE\_C\_OR\_Q\_TO, "Жалғастыру үшін `c` немесе шығу үшін [`q` немесе кез келген таңба] теріңіз: "},**  **{Translate::BOOL\_TRUE, "Қосылған"},**  **{Translate::BOOL\_FALSE, "Өшірілген"},**  **},**  **},**  **{**  **Languages::RU,**  **std::map<Translate, std::string>{**  **{Translate::USER\_ANSWER\_NUM, "Выберите индекс ответа # "},**  **{Translate::WRONG\_INDEX, "Неверный индекс. Попробуйте еще раз!"},**  **{Translate::ERROR\_WHILE\_CHECKING\_ID, "Ошибка: Проверьте свои идентификаторы викторины!"},**  **{Translate::CFG\_SHUFFLE\_QUESTION, "Перетасовать\_вопросы = "},**  **{Translate::CFG\_SHUFFLE\_ANSWERS, "Перетасоват\_ответы = "},**  **{Translate::COMPLEXITY\_EASY, "Сложность: Легко"},**  **{Translate::COMPLEXITY\_MEDIUM, "Сложность: Средняя"},**  **{Translate::COMPLEXITY\_DIFFICULT, "Сложность: Сложно"},**  **{Translate::COMPLEXITY\_NOT\_SET, "Сложность не установлена"},**  **{Translate::QUESTION, "Вопрос: : "},**  **{Translate::CORRECT\_ANSWER, "Правильный ответ #"},**  **{Translate::WRONG\_ANSWER, "Неправильный ответ #"},**  **{Translate::CORRECT\_ANSWER\_COUNT, "Количество правильных ответов: "},**  **{Translate::YOUR\_ANSWERS, "Ваши ответы: "},**  **{Translate::CORRECT\_ANSWERS, "Правильные ответы: "},**  **{Translate::YOUR\_SCORE, "Ваша оценка: "},**    **{Translate::TYPE\_C\_OR\_Q\_TO, "Введите `c`, чтобы продолжить, или [`q` или любой символ], чтобы выйти: "},**  **{Translate::BOOL\_TRUE, "Активировано"},**  **{Translate::BOOL\_FALSE, "Неактивировано"},**  **},**  **},**  **{**  **Languages::EN,**  **std::map<Translate, std::string>{**  **{Translate::USER\_ANSWER\_NUM, "Choose answer index : # "},**  **{Translate::WRONG\_INDEX, "Wrong index. Try again!"},**  **{Translate::ERROR\_WHILE\_CHECKING\_ID, "Error : Check your quizzes IDs!"},**  **{Translate::CFG\_SHUFFLE\_QUESTION, "SHUFFLE\_QUESTION = "},**  **{Translate::CFG\_SHUFFLE\_ANSWERS, "SHUFFLE\_ANSWERS = "},**  **{Translate::COMPLEXITY\_EASY, "Complexity : Easy"},**  **{Translate::COMPLEXITY\_MEDIUM, "Complexity : Medium"},**  **{Translate::COMPLEXITY\_DIFFICULT, "Complexity : Difficult"},**  **{Translate::COMPLEXITY\_NOT\_SET, "The Complexity:: is not set"},**  **{Translate::QUESTION, "Question : "},**  **{Translate::CORRECT\_ANSWER, "Correct answer #"},**  **{Translate::WRONG\_ANSWER, "Wrong answer #"},**  **{Translate::CORRECT\_ANSWER\_COUNT, "Correct answer count : "},**  **{Translate::YOUR\_ANSWERS, "Your answers : "},**  **{Translate::CORRECT\_ANSWERS, "Correct answers : "},**  **{Translate::YOUR\_SCORE, "Your score : "},**  **{Translate::TYPE\_C\_OR\_Q\_TO, "Type `c` to continue or [`q` or any char] to quit : "},**  **{Translate::BOOL\_TRUE, "Active"},**  **{Translate::BOOL\_FALSE, "Inactive"},**  **},**  **},**  **};**    **// Configs end**    **enum Color**  **{**  **RED,**  **GREEN,**  **YELLOW,**  **BLUE,**  **ORANGE,**  **CYAN,**  **RESET,**  **};**    **std::map<Color, std::string> termColor = {**  **{Color::RED, "\033[91m"},**  **{Color::GREEN, "\033[92m"},**  **{Color::YELLOW, "\033[93m"},**  **{Color::BLUE, "\033[34m"},**  **{Color::ORANGE, "\u001b[38;5;166m"},**  **{Color::CYAN, "\033[36m"},**  **{Color::RESET, "\033[0m"},**  **};**    **template <typename T>**  **void shuffleVector(std::vector<T> &v)**  **{**  **std::random\_device rd;**  **std::mt19937 gen(rd());**  **std::uniform\_int\_distribution<> dis(0, v.size() - 1);**  **for (int i = 0; i < v.size(); ++i)**  **{**  **int randIndex = dis(gen);**  **T temp = v[i];**  **v[i] = v[randIndex];**  **v[randIndex] = temp;**  **}**  **}**    **class Answer**  **{**  **public:**  **std::vector<std::string> correctAnswers;**  **std::vector<std::string> wrongAnswers;**    **std::vector<std::string> getAnswers()**  **{**  **std::vector<std::string> merged;**  **merged.insert(merged.end(), correctAnswers.begin(), correctAnswers.end());**  **merged.insert(merged.end(), wrongAnswers.begin(), wrongAnswers.end());**  **return merged;**  **}**    **std::vector<std::string> getShuffled()**  **{**  **std::vector<std::string> merged;**  **merged.insert(merged.end(), correctAnswers.begin(), correctAnswers.end());**      **merged.insert(merged.end(), wrongAnswers.begin(), wrongAnswers.end());**  **shuffleVector(merged);**  **return merged;**  **}**  **};**    **enum Complexity**  **{**  **EASY,**  **MEDIUM,**  **DIFFICULT,**  **};**    **struct MultipleChoiceQuiz**  **{**  **int id;**  **std::string question;**  **Answer answers;**  **Complexity complexity;**  **};**    **class Exam**  **{**  **private:**  **std::vector<MultipleChoiceQuiz> quizzes;**    **std::string to\_str(int a)**  **{**  **return std::to\_string(a);**  **}**    **std::string bool\_2\_str(bool b)**  **{**  **return (b) ? translate[UI\_LANG][Translate::BOOL\_TRUE] : translate[UI\_LANG][Translate::BOOL\_FALSE];**  **}**    **std::vector<int> getAnswersFromUser(int answerCount, int exclusiveHighBound)**  **{**  **std::set<int> result;**  **for (int i = 0; i < answerCount; ++i)**  **{**  **int input;**  **std::cout << translate[UI\_LANG][Translate::USER\_ANSWER\_NUM] << i + 1 << ": ";**  **std::cin >> input;**  **if (input < 0 or input >= exclusiveHighBound or std::cin.fail())**  **{**  **std::cin.clear();**  **std::cin.ignore(32767, '\n');**  **--i;**  **std::cout << translate[UI\_LANG][Translate::WRONG\_INDEX] << "\n";**  **continue;**  **}**  **result.insert(input);**  **}**  **return std::vector<int>(result.begin(), result.end());**  **}**    **public:**  **Exam(std::vector<MultipleChoiceQuiz> \_quizzes)**  **{**  **addQuizzes(\_quizzes);**  **}**    **void addQuizzes(std::vector<MultipleChoiceQuiz> \_quizzes)**  **{**  **quizzes.insert(quizzes.end(), \_quizzes.begin(), \_quizzes.end());**  **checkDuplicatedIDs();**  **}**    **auto getQuizzes()**  **{**  **return quizzes;**  **}**    **void checkDuplicatedIDs()**  **{**  **std::set<int> IDs;**  **for (const auto &quiz : quizzes)**  **IDs.insert(quiz.id);**    **if (quizzes.size() != IDs.size())**  **{**  **std::cout << translate[UI\_LANG][Translate::ERROR\_WHILE\_CHECKING\_ID] << "\n";**  **for (const auto &quiz : quizzes)**  **std::cout << quiz.id << "   ";**  **exit(1);**  **}**      **}**    **void debug(bool printable)**  **{**  **std::string log = "";**    **log.append(translate[UI\_LANG][Translate::CFG\_SHUFFLE\_QUESTION] + bool\_2\_str(SHUFFLE\_QUESTIONS) + "\n");**  **log.append(translate[UI\_LANG][Translate::CFG\_SHUFFLE\_ANSWERS] + bool\_2\_str(SHUFFLE\_ANSWERS) + "\n");**    **for (const auto &quiz : quizzes)**  **{**  **// id**  **log += std::string("id : " + to\_str(quiz.id) + "\n");**    **// complexity**  **switch (quiz.complexity)**  **{**  **case Complexity::EASY:**  **log += std::string(translate[UI\_LANG][Translate::COMPLEXITY\_EASY] + "\n");**  **break;**  **case Complexity::MEDIUM:**  **log += std::string(translate[UI\_LANG][Translate::COMPLEXITY\_MEDIUM] + "\n");**  **break;**  **case Complexity::DIFFICULT:**  **log += std::string(translate[UI\_LANG][Translate::COMPLEXITY\_DIFFICULT] + "\n");**  **break;**  **default:**  **std::cerr << translate[UI\_LANG][Translate::COMPLEXITY\_NOT\_SET];**  **exit(1);**  **}**  **// question**  **log += std::string(translate[UI\_LANG][Translate::QUESTION] + quiz.question + "\n");**    **// answers**  **for (int i = 0; i < quiz.answers.correctAnswers.size(); ++i)**  **log += std::string(translate[UI\_LANG][Translate::CORRECT\_ANSWER] + to\_str(i) + " " + quiz.answers.correctAnswers[i] +**  **"\n");**  **for (int i = 0; i < quiz.answers.wrongAnswers.size(); ++i)**  **log += std::string(translate[UI\_LANG][Translate::WRONG\_ANSWER] + to\_str(i) + " " + quiz.answers.wrongAnswers[i] + "\n");**  **log += std::string("\n");**  **}**  **if (printable)**  **std::cout << log;**  **}**    **void conduct()**  **{**  **// std::cout << "Conducting exam..." << "\n";**  **while (true)**  **{**  **// Generate quiz index for default or shuffled**  **std::vector<int> quiz\_indexes;**  **quiz\_indexes.reserve(quizzes.size());**  **for (int i = 0; i < quizzes.size(); ++i)**  **quiz\_indexes.push\_back(i);**  **if (SHUFFLE\_QUESTIONS)**  **shuffleVector(quiz\_indexes);**    **// Start exam...**  **for (const auto &quiz\_it : quiz\_indexes)**  **{**  **MultipleChoiceQuiz quiz = quizzes[quiz\_it];**    **std::cout << termColor[Color::BLUE] << "id: " << quiz.id << ") " << quiz.question << "    " << termColor[Color::CYAN];**  **switch (quiz.complexity)**  **{**  **case Complexity::EASY:**  **std::cout << translate[UI\_LANG][Translate::COMPLEXITY\_EASY] << "\n";**  **break;**  **case Complexity::MEDIUM:**  **std::cout << translate[UI\_LANG][Translate::COMPLEXITY\_MEDIUM] << "\n";**  **break;**  **case Complexity::DIFFICULT:**  **std::cout << translate[UI\_LANG][Translate::COMPLEXITY\_DIFFICULT] << "\n";**  **break;**  **default:**  **std::cerr << translate[UI\_LANG][Translate::COMPLEXITY\_NOT\_SET];**  **exit(1);**  **}**    **std::vector<std::string> answers;**  **if (SHUFFLE\_ANSWERS)**  **answers = quiz.answers.getShuffled();**  **else**  **answers = quiz.answers.getAnswers();**    **std::cout << termColor[Color::RESET] << "\n";**    **for (int ans\_i = 0; ans\_i < answers.size(); ++ans\_i)**  **std::cout << termColor[Color::ORANGE] << ans\_i << ") " << answers[ans\_i] << "\n"**  **<< termColor[Color::RESET];**  **std::cout << "\n";**    **std::vector<std::string> correctAnswers = quiz.answers.correctAnswers;**    **std::cout << translate[UI\_LANG][Translate::CORRECT\_ANSWER\_COUNT] << correctAnswers.size() << "\n";**  **std::vector<int> userAnswers = getAnswersFromUser(correctAnswers.size(), answers.size());**  **std::cout << "\n";**  **std::cout << termColor[Color::YELLOW] << translate[UI\_LANG][Translate::YOUR\_ANSWERS];**  **for (const auto &i : userAnswers)**  **std::cout << answers[i] << "   ";**  **std::cout << "\n";**    **std::cout << translate[UI\_LANG][Translate::CORRECT\_ANSWERS];**  **for (const auto &e : correctAnswers)**  **std::cout << e << "   ";**  **std::cout << "\n";**    **int score = 0;**  **for (const auto &userInput : userAnswers)**  **{**  **if (std::find(correctAnswers.begin(), correctAnswers.end(), answers[userInput]) != correctAnswers.end())**  **score += 1;**  **}**    **std::cout << "\n";**  **std::cout << termColor[Color::GREEN] << translate[UI\_LANG][Translate::YOUR\_SCORE] << score << "/" << correctAnswers.size()**  **<< "\n";**  **std::cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";**  **std::cout << "\n\n\n";**  **}**  **char isExit;**  **std::cout << termColor[Color::RESET] << translate[UI\_LANG][Translate::TYPE\_C\_OR\_Q\_TO];**  **std::cin >> isExit;**  **if (isExit == 'c')**  **continue;**  **exit(1);**  **}**  **}**  **};**  **int main()**  **{**  **Exam exam(**  **std::vector<MultipleChoiceQuiz>{**  **MultipleChoiceQuiz{**  **0,**  **"Best programming languages?",**  **Answer{**  **std::vector<std::string>{"C++", "Java", "Rust", "Python"},**  **std::vector<std::string>{"Brainfuck", "Scratch"},**  **},**  **Complexity::DIFFICULT},**  **MultipleChoiceQuiz{**  **2,**  **"MS-DOS ОЖ-де файл атауындағы таңбалардың максималды саны",**  **Answer{**  **std::vector<std::string>{"8", "binary:0000\_1000"},**  **std::vector<std::string>{"255", "256", "10", "12"},**  **},**  **Complexity::MEDIUM},**  **MultipleChoiceQuiz{**  **3,**  **"Операциялық жүйелер құрамына мыналар кіреді:",**  **Answer{**  **std::vector<std::string>{"жүйелік бағдарламалық қамтамасыз ету"},**  **std::vector<std::string>{"қолданбалы бағдарламалық қамтамасыз ету",**  **"бағдарламалау жүйелері",**  **"деректер базасын басқару жүйелері",**  **"бірегей бағдарламалық қамтамасыз ету"},**  **},**  **Complexity::EASY},**  **MultipleChoiceQuiz{**  **4,**  **"WINDOWS операциялық жүйесі келесі функцияны орындамайды",**  **Answer{**  **std::vector<std::string>{"бағдарламалық жүйелерді құру"},**  **std::vector<std::string>{"қолданбалы бағдарламаларды іске қосу",**  **"файлдық жүйемен жұмыс істеуді қамтамасыз ету",**  **"компьютердің аппараттық құралдарын басқару",**  **"бірнеше қосымшалардың бір уақытта жұмыс істеуін қамтамасыз етеді"},**  **},**  **Complexity::MEDIUM},**  **MultipleChoiceQuiz{**  **5,**  **"Сommand.com бағдарламасы ...",**    **Answer{**  **std::vector<std::string>{"пайдаланушы енгізген командаларды өңдейді"},**  **std::vector<std::string>{"операциялық жүйенің барлық командаларын сақтайды",**  **"командалар мен бағдарламаларды өңдейді",**  **"барлық командаларды өз жұмысында сақтайды",**  **"дискіні пішімдейді"},**  **},**  **Complexity::EASY},**  **MultipleChoiceQuiz{**  **6,**  **"Жедел жадтың жылдамдығы немен өлшенеді",**  **Answer{**  **std::vector<std::string>{"наносекундпен", "nanosec"},**  **std::vector<std::string>{"миллисекундпен",**  **"секундтармен",**  **"мегагерцпен",**  **"гигагерцпен"},**  **},**  **Complexity::EASY},**    **});**  **exam.debug(true);**  **exam.conduct();**  **}** |

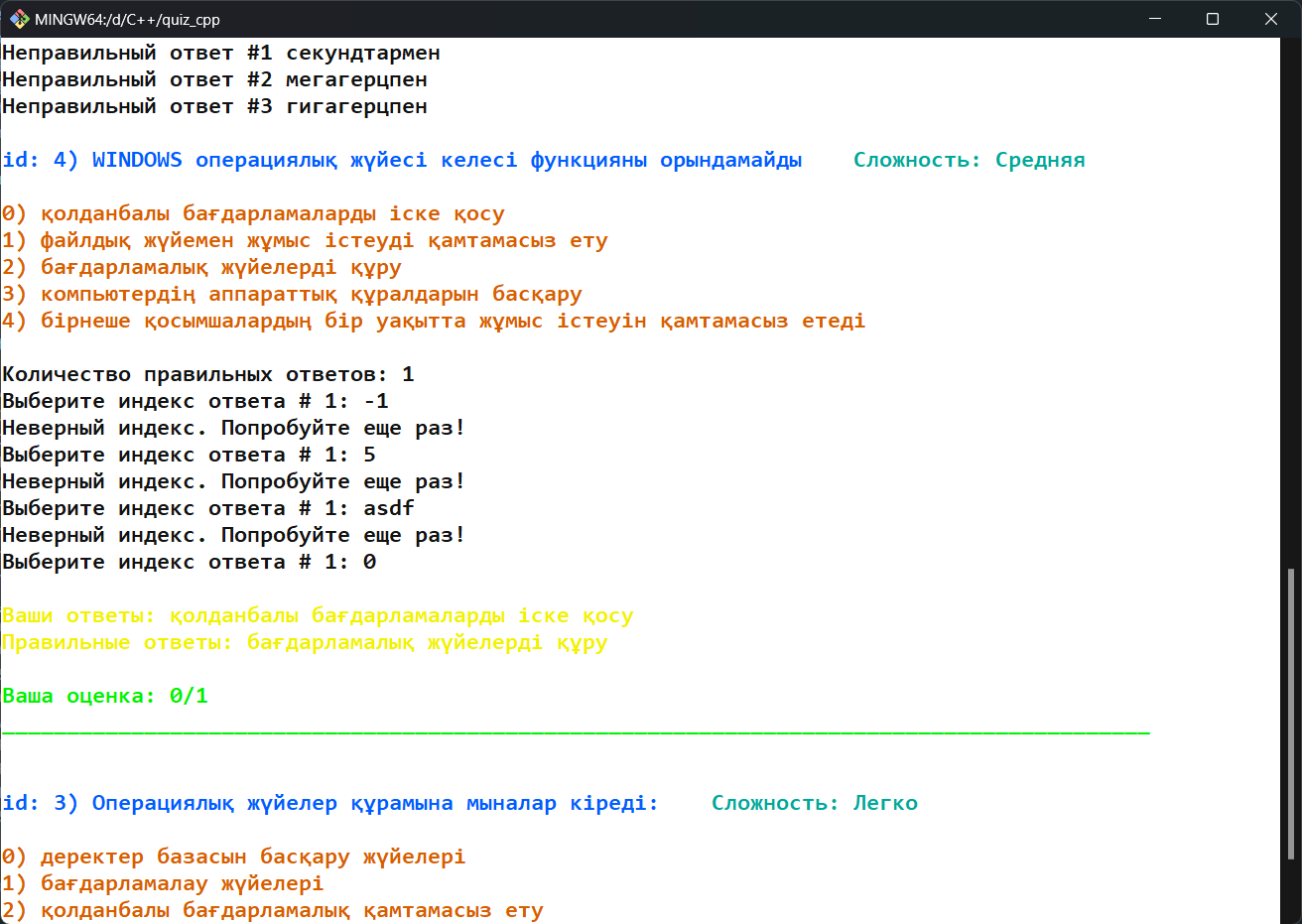
*Нәтижесі:*

Скриншот-1

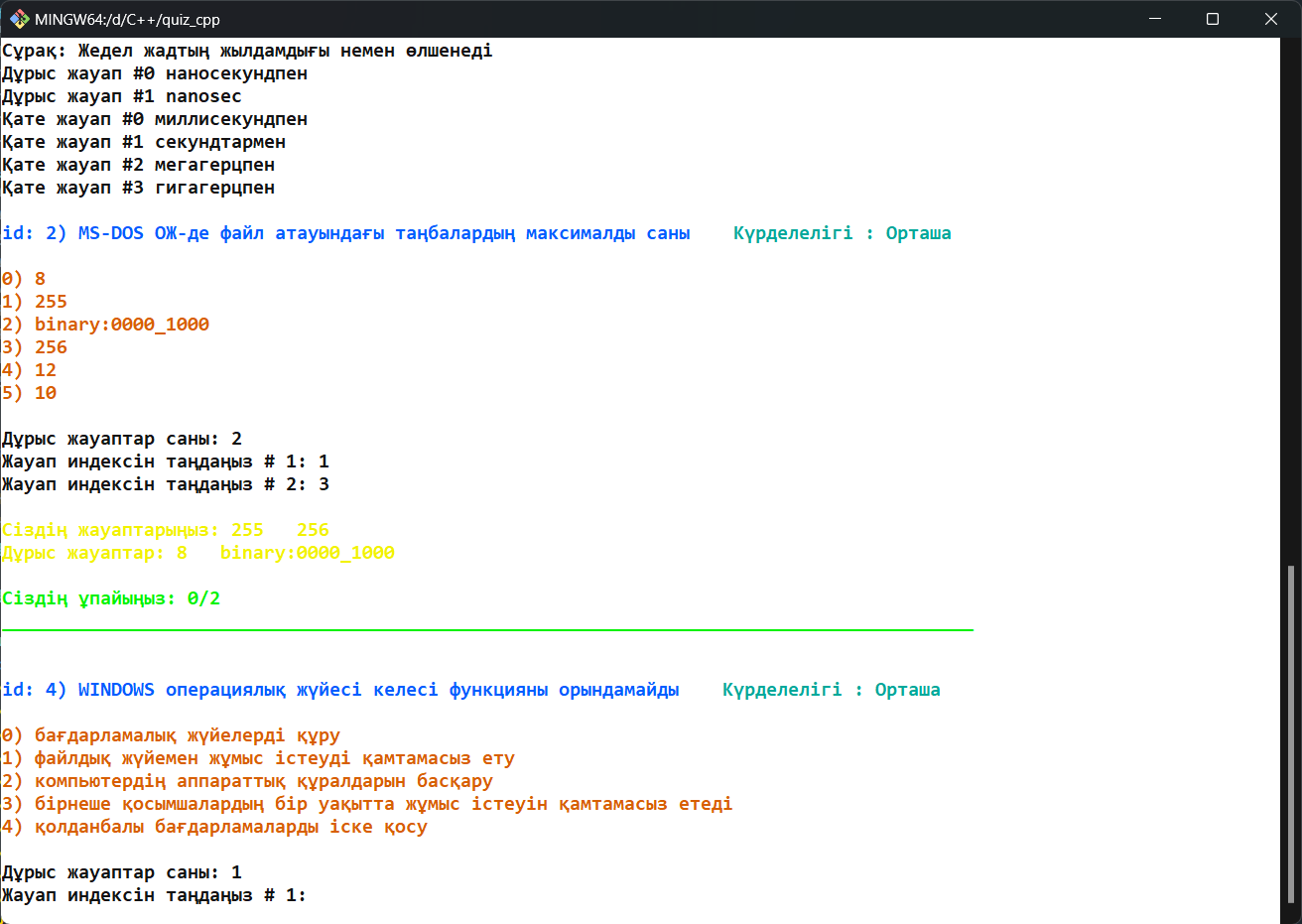


Бұл скриншотта exam.debug(true); функциясының шақырылғандағы экранға шығарылған қосымша ақпарат көрсетілген. Қаласаңыздар false мәнімен шақырып тек қана тексеру жасауға болады. Бағдарламада қолданушының қате енгізген ақпаратын өңдейтін және қолданушының ыңғайына қарай өзгертуге болатын және тілді өзгертетін функциялар бар.

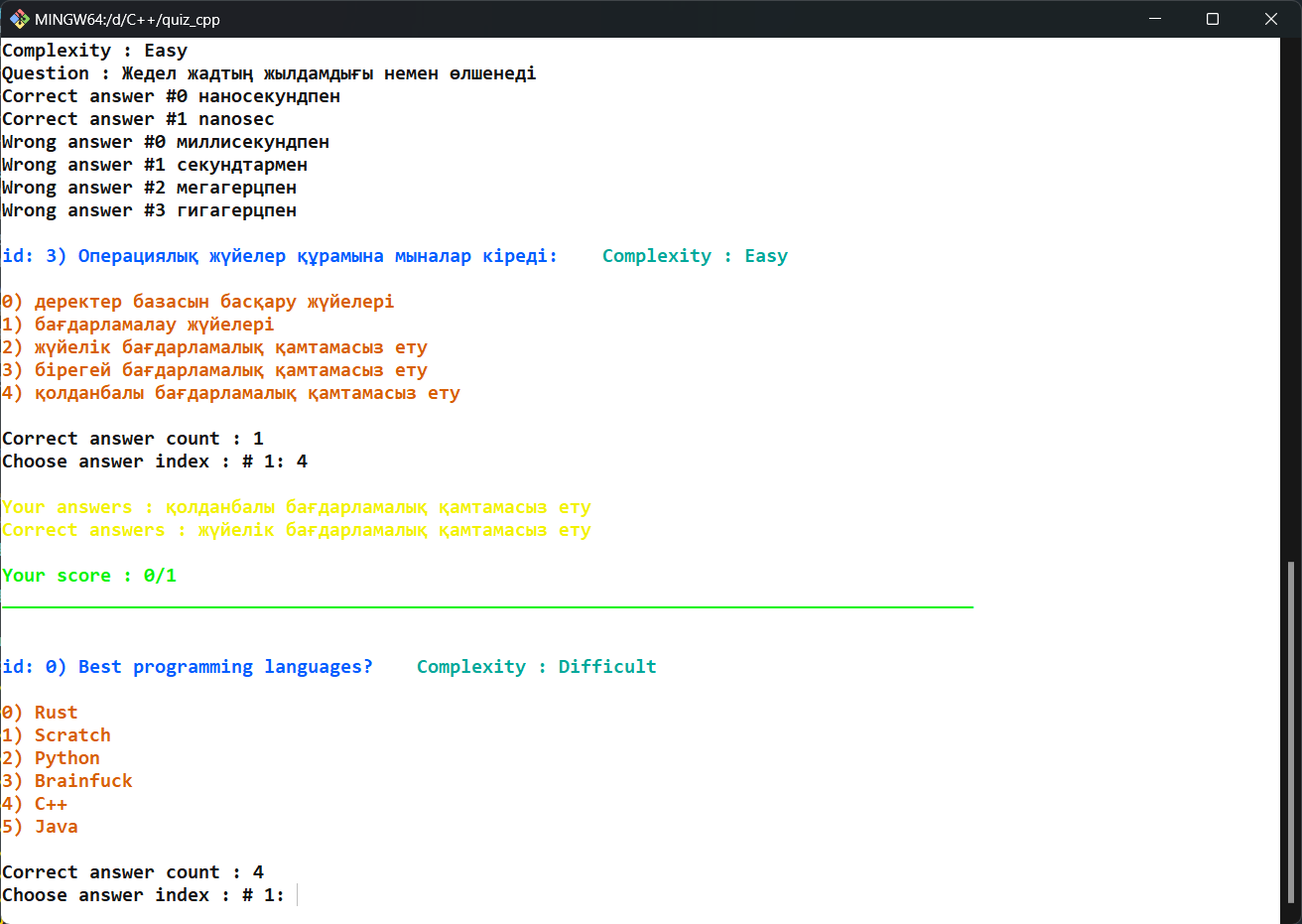
Скриншот-2



Скриншот-3



Скриншот-4



Скриншот-5

