



Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Астраханский государственный технический университет»
Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS
по международному стандарту ISO 9001:2015

Институт Информационных технологий и коммуникаций
Направление 09.03.04 Программная инженерия
Профиль Разработка программно-информационных систем
Кафедра Автоматизированные системы обработки информации и управления
от ____ . ____ .20 ____ № ____

ОТЧЕТ ПО ПРАКТИКЕ

Вид практики: учебная

Место прохождения практики:

Астраханский государственный технический университет,
кафедра «Автоматизированные системы обработки информации и управления»

Отчет выполнил:
обучающийся группы ДИПР6-21

_____ Катунин Сергей Сергеевич

Руководитель практики от Университета
доцент

_____ Лаптев Валерий Викторович

Результаты защиты отчета

Оценка, полученная на защите

« _____ »

Члены комиссии:

_____ Толасова В.В.

_____ Лаптев В.В.

« ____ » _____ 201__ г.

Астрахань 2020

ИНДИВИДУАЛЬНЫЙ ПЛАН/ЗАДАНИЕ

Вид практики: учебная / производственная / преддипломная

Способ проведения практики: выездная / стационарная

Обучающийся Катунин Сергей Сергеевич, группа ДИПР6-21.

Направление 09.03.04 Программная инженерия.

Профиль Разработка программно-информационных систем.

Место проведения практики Астраханский государственный технический университет, кафедра «Автоматизированные системы обработки информации и управления»

Задание: произвести реинжиниринг и рефакторинг программного продукта, разработанного в рамках курсового проекта по дисциплине «Программирование и информатика» (1 семестр).
Тема курсового проекта: программа-тренажер «**Разложение натурального числа на простые множители**».

Требования к программному проекту:

- Код программы должен быть переписан на языке C++ (стандарт C++ 11 и выше).
- Вместо консольного интерфейса должен быть разработан оконный интерфейс, в котором использованы необходимые элементы управления.
- Интерфейс должен быть реализован с помощью WinAPI, программа должна работать под управлением операционной системы Windows XP и более поздних.
- Программа должна обеспечивать многократное выполнение тренировочных упражнений, завершение работы – по требованию пользователя.
- Программа должна собирать статистику по всем выполненным упражнениям и подсчитывать:
 - ✓ количество выполненных упражнений;
 - ✓ количество использованных подсказок;
 - ✓ количество повторенных заданий;
 - ✓ количество и процент правильно выполненных упражнений;
 - ✓ количество и процент неправильно выполненных упражнений;
 - ✓ серию правильных ответов максимальной длины;
 - ✓ серию неправильных ответов максимальной длины;
 - ✓ средний размер серии правильных ответов;
 - ✓ при наличии неправильных ответов – отношение количества правильных ответов к количеству неправильных.

Объем и краткое содержание (виды работ) практики

№ п/п	Раздел практики	Формы текущего контроля успеваемости
1	1 этап (подготовительный) 1. Получение индивидуального задания на практику. 2. Инструктаж по технике безопасности при работе на персональном компьютере и в компьютерном классе	Регистрация в журнале по технике безопасности, собеседование
2	2 этап (теоретический) Выполнение индивидуального задания: анализ предметной области 1. Проведение анализа предметной области в соответствии с утвержденным планом; 2. Поиск и анализ информации по индивидуальному заданию; сбор данных, необходимых для решения поставленных задач. 3. Разработка технического задания, технического проекта, а также программы и методики испытаний программного продукта	Техническое задание. Разделы отчёта по практике «Технический проект» и «Программа и методика испытаний» для разрабатываемого программного продукта
3	3 этап (экспериментальный) Выполнение индивидуального задания: обработка и анализ полученной информации 1. Разработка и тестирование программного продукта. 2. Обработка и анализ полученной информации, сравнение полученных результатов с известными данными, при необходимости – внесение изменений в программный код. 3. Обоснование полученных выводов.	Раздел отчёта по практике – «Рабочий проект» для разработанного программного продукта
4	4 этап: подготовка отчета по практике Подготовка отчета, в котором должны быть отражены результаты работы. Составление пояснительной записки в виде файла и твердой копии. Подготовка презентации для выступления на защите в комиссии кафедры	Материал по результатам работы: программа, отвечающая поставленным требованиям, письменный отчет, электронная презентация
5	Заключительный этап: Защита отчета по практике на кафедре	Отчет по результатам практики
	Форма отчетности по практике	Зачет с оценкой

Руководитель практики от Университета:

доцент Лаптев Валерий Викторович

« ____ » _____ 201__ г. _____
подпись

Задание получил:

Катунин Сергей Сергеевич

« ____ » _____ 201__ г. _____
подпись

РАБОЧИЙ ГРАФИК (ПЛАН) ПРОВЕДЕНИЯ ПРАКТИКИ

2019/2020 учебный год

Направление 09.03.04 Программная инженерия.

Профиль Разработка программно-информационных систем.

Курс 2, группа ДИПР6-21.

Место прохождения практики Астраханский государственный технический университет,
кафедра «Автоматизированные системы обработки информации и управления»

Руководитель практики от Университета доцент Лаптев В.В..

Вид практики: учебная / производственная / преддипломная

Способ проведения практики: выездная/стационарная

Срок прохождения практики: с 27.01.2020 по 08.02.2020.

Дата (сроки)	Планируемые формы работы (раздел практики)
27.01.2020	Знакомство с правилами внутреннего распорядка, изучение структуры организации, инструктаж по технике безопасности. Получение индивидуального задания на практику
28.01.2020 - 29.01.2020	Анализ предметной области, сбор информации, необходимой для разработки технического задания, технического проекта и программы испытаний программного продукта
30.01.2020- 04.02.2020	Разработка и тестирование программного продукта. Обработка и анализ полученной информации, сравнение полученных результатов исследований с известными данными, при необходимости – внесение изменений в программный код
05.02.2020- 07.02.2020	Подготовка отчета. Составление пояснительной записки в виде файла и твердой копии. Подготовка презентации для выступления на защите в комиссии кафедры
08.02.2020	Защита практики

Руководитель практики от университета доцент Лаптев Валерий Викторович

«___» _____ 201__ г. _____
подпись

Обучающийся Катунин Сергей Сергеевич

«___» _____ 201__ г. _____
подпись

ДНЕВНИК ПРАКТИКИ

Вид практики: учебная / производственная/ преддипломная

Способ проведения практики: выездная/стационарная

Студент Катунин Сергей Сергеевич, группа ДИПР6-21

Направление 09.03.04 Программная инженерия

Профиль Разработка программно-информационных систем.

Место проведения практики Астраханский государственный технический университет,
кафедра «Автоматизированные системы обработки информации и управления»

Дата начала практики «27» января 2020 г.

Дата окончания практики «08» февраля 2020 г.

Дата	Наименование и ход работ,	Краткое описание работы	Оценка, замечания и предложения руководителя практики
27.01.2020	Цели и задачи практики, техника безопасности в компьютерном классе, постановка задачи	Изучена инструкция по технике безопасности в компьютерном классе. Получено задание на практику.	
28.01.2020	Обследование предметной области. Разработка системы тестов. Разработка технического задания.	Собран и проанализирован материал по предметной области. Написано техническое задание. Разработана система тестов.	
29.01.2020	Обследование предметной области. Разработка форматов данных, алгоритмов решения задачи.	Разработаны форматы данных. Спроектирован пользовательский интерфейс. Реализовано главное окно интерфейса, а также переключение дочерних окон внутри главного с помощью соответствующих кнопок.	
30.01.2020	Разработка и тестирование программного продукта	Разработан класс Trenirovka для генерации и представления задания в виде составного числа, которое необходимо разложить на множители. Внесены корректирующие изменения в интерфейс, начато добавление функционала для дочернего окна с тренажером.	
31.01.2020	Разработка и тестирование программного продукта	Разработаны базовые функции тренажера: генерация и представление задания, ввод и проверка ответа на корректность. Внесены различные улучшения в интерфейс, минимизирующие лишние действия от пользователя во время работы с тренажером.	
01.02.2020	Разработка и тестирование программного продукта	Спроектирован интерфейс для окна со статистикой и для окна с инструкцией.	
02.02.2020	Выходной день	—	—

Дата	Наименование и ход работ,	Краткое описание работы	Оценка, замечания и предложения руководителя практики
03.02.2020	Разработка и тестирование программного продукта	Разработан класс Stats для сбора статистики и сохранения её в файл. Также реализован вывод статистики на экран в окне статистики из файла.	
04.02.2020	Разработка и тестирование программного продукта	Реализована кнопка «Повтор» для повторного выполнения неправильно выполненного задания по требованию пользователя. Тестирование программного продукта.	
05.02.2020	Оформление отчёта	Оформлены разделы 1 и 2, введение, заключение, список источников.	
06.02.2020	Оформление отчёта	Оформлен раздел 3, приложения. Оформлено техническое задание на разработку учебно-демонстрационной программы.	
07.02.2020	Оформление отчёта	Подготовлена презентация к докладу	
08.02.2020	Защита практики		

Руководитель практики от Университета
доцент

_____ Лаптев Валерий Викторович

«___» _____ 201__ г.

Обучающийся Катунин Сергей Сергеевич

«___» _____ 201__ г. _____
подпись

СОДЕРЖАНИЕ

Введение.....	8
1 Технический проект.....	9
1.1 Анализ предметной области.....	9
1.1.1 Простые числа.....	9
1.1.2 Проверка знаний и сбор статистики.....	9
1.2 Технология обработки информации.....	10
1.2.1 Диаграмма классов.....	11
1.2.2 Форматы данных.....	13
1.2.3 Алгоритм генерации числа.....	14
1.2.4 Алгоритм генерации и проверки задания.....	14
1.2.5 Основной алгоритм.....	15
1.3 Входные и выходные данные.....	15
1.4 Системные требования.....	15
2 Рабочий проект.....	17
2.1 Общие сведения о работе системы.....	17
2.2 Функциональное назначение программного продукта.....	17
2.3 Установка и выполнение программного продукта.....	17
2.4 Описание программы.....	17
2.5 Разработанные меню и интерфейсы.....	22
2.6 Сообщения системы.....	26
3 Программа и методика испытаний.....	28
Заключение.....	29
Список использованных источников.....	30
Приложение 1 Техническое задание на разработку программы-тренажера.....	31
Приложение 2 Диаграмма классов.....	34
Приложение 3 Файл со статистикой.....	35

ВВЕДЕНИЕ

Разложение натурального числа на простые множители является стандартной задачей в средних классах школы на уроках математики. Умение раскладывать число на простые множители необходимо школьникам, к примеру, чтобы найти НОД (наибольший общий делитель) и НОК (наименьшее общее кратное чисел), а также для сокращения различных дробей. Факторизация (разложение чисел на простые множители) для больших чисел является задачей большой сложности. Не существует никакого известного способа, чтобы решить эту задачу быстро. Её сложность лежит в основе некоторых алгоритмов шифрования с открытым ключом, таких как RSA. Из-за этой сложности при большом количестве учеников преподавателю будет необходимо большое количество времени для проверки всех заданий, решенных учениками.

Целью создания программы «Разложение натурального числа на простые множители» является автоматизация процесса изучения школьниками 5-6 классов простых чисел и работы с ними.

Назначение программы – повышение успеваемости и качества знаний школьников, снижение нагрузки на преподавателя. Используя данную программу-тренажер на нескольких компьютерах, время, затрачиваемое на проверку, сокращается в несколько раз соответственно (благодаря статистике решенных заданий, которую ведет программа). К тому же не стоит забывать о «человеческом факторе» – преподаватель может что-то упустить или не заметить. С программой данная проблема исключается.

1 ТЕХНИЧЕСКИЙ ПРОЕКТ

1.1 Анализ предметной области

1.1.1 Простые числа

Простое число – натуральное (целое положительное) число, имеющее ровно два различных натуральных делителя — единицу и самого себя. Другими словами, число x является простым, если оно больше 1 и при этом делится без остатка только на 1 и на x . К примеру, 5 – простое число, а 6 является составным числом, так как, помимо 1 и 6, также делится на 2 и на 3. Основная теорема арифметики устанавливает основную роль простых чисел в теории чисел: любое целое число, больше, чем 1, является простым или может быть выражено как произведение простых чисел, которое является уникальным с точностью до порядка. В таблице 1.1 приведены первые двадцать простых чисел.

Таблица 1.1 – Начало таблицы простых чисел (первые 20 элементов).

2	3	5	7	11
13	17	19	23	29
31	37	41	43	47
53	59	61	67	71

Таким образом, разложение числа на простые множители является представлением составного числа в виде произведения простых чисел-множителей. Например, чтобы разложить число 30 на простые множители, ученику необходимо разделить число 30 на самое маленькое простое число, которое является его делителем (число 2). А затем полученное число 15 аналогично необходимо разделить на наименьший простой делитель (число 3), в результате чего получается число 5, которое является простым. Таким образом, получаем, что составное число 30 представимо как произведение простых множителей 2, 3 и 5. Соответственно, ученик в процессе решения должен перебирать делители от 2 до квадратного корня числа.

1.1.2 Тренировка и сбор статистики

После того, как ученик ознакомился с теоретическим материалом по теме «Разложение натурального числа на простые множители», он может воспользоваться тренажером, для того,

чтобы выяснить, насколько хорошо был усвоен полученный теоретический материал, при условии, что ученик владеет навыками умножения и деления.

Во время работы тренажера, ученик может воспользоваться подсказкой, которая сообщит ему один из простых множителей числа или повторить неправильно выполненное задание. Использованное количество подсказок и повторов отмечаются в статистике. Задания генерируются, пока не будет прекращена тренировка по желанию обучаемого.

По завершении тренировки должна быть сохранена статистика по всем выполненным упражнениям. Таким образом ученик или преподаватель могут ознакомиться с результатом тренировки в статистике. Процент правильных и неправильных ответов в статистике высчитывается по формуле:

$$P = \frac{A}{N} * 100\% , \quad (1.1)$$

где

P – процент правильных / неправильных ответов,

A – количество правильных / неправильных ответов,

N – общее количество выполненных упражнений.

Средний размер серии правильных ответов высчитывается по формуле:

$$Avg = \frac{Summa}{Count} , \quad (1.2)$$

где

Avg – средний размер серии правильных ответов,

Summa – сумма всех серий правильных ответов,

Count – общее количество серий правильных ответов.

Отношение количества правильных ответов к количеству неправильных можно вычислить по формуле (при наличии неправильных ответов):

$$Ratio = \frac{R}{W} , \quad (1.3)$$

где

Ratio – отношение количества правильных ответов к количеству неправильных,

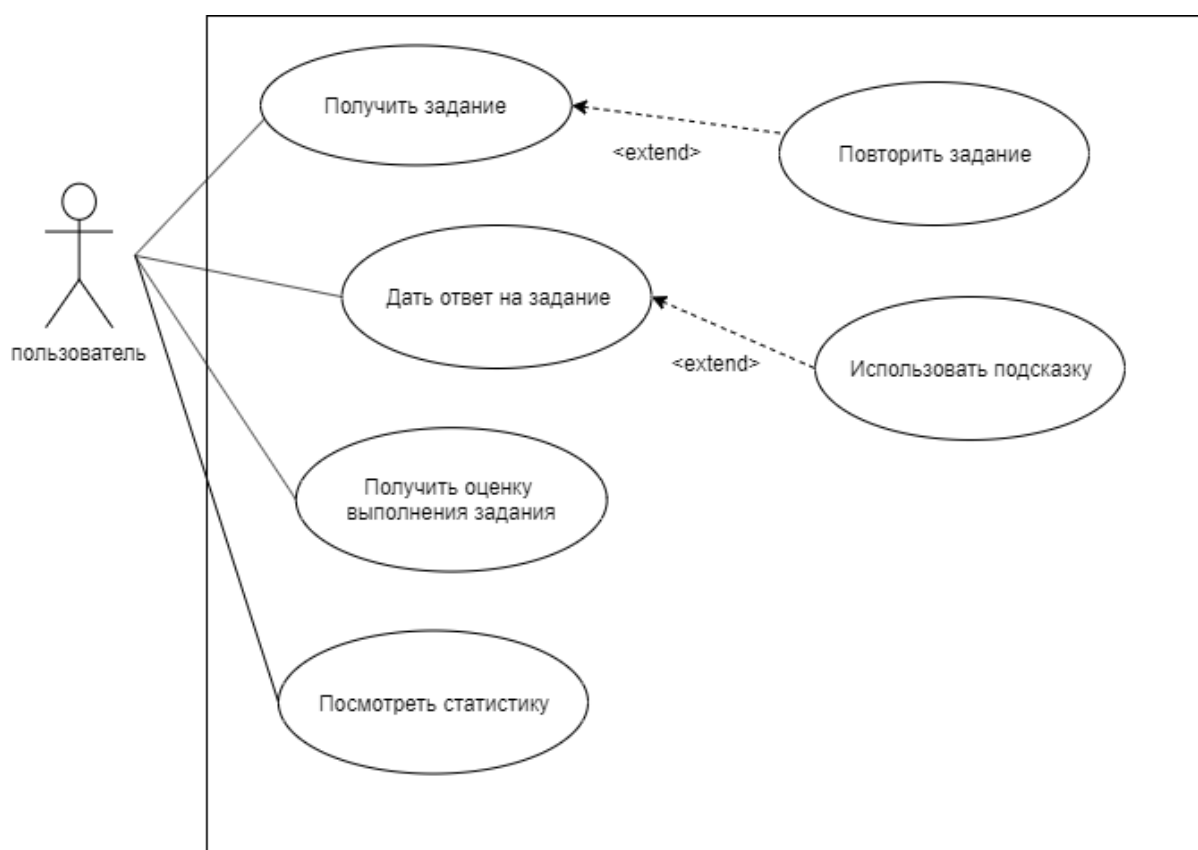
Summa – количество правильных ответов,

Count – количество неправильных ответов.

1.2 Технология обработки информации

Анализ предметной области показал, что программа рассчитана на одного пользователя, который может воспользоваться тренажером и просмотреть статистику. Во время тренировки пользователь имеет возможность повторить неправильно выполненное задание, а также использовать подсказку.

В ходе анализа предметной области была построена диаграмма вариантов использования (рис. 1.1).



Вариант использования «Получить задание» расширен функцией «Повторить задание».

Вариант использования «Дать ответ на задание» расширен функцией «Использовать подсказку».

1.2.1 Форматы данных

Задание представляет собой строковую переменную, содержащую целое число, над которым следует выполнить указанное действие. Числа, которые необходимо разложить на простые множители, для заданий генерируются в диапазоне от 30 до 1000, так как задания с числами больше этого диапазона будут слишком сложны для решения. Поскольку числа в заданиях генерируются в диапазоне от 30 до 1000, то для составления такого числа достаточно использовать простые множители в диапазоне от 2 до 19 (первые восемь простых чисел). Соответственно для генерации задания используется массив, состоящий из первых восьми

простых чисел и массив из целых элементов, куда заносятся простые числа (множители) в порядке неубывания, произведение элементов этого массива предлагается пользователю в качестве задания. Ответ пользователя заносится в другой целочисленный массив, который также отсортирован по неубыванию.

Статистика хранится в текстовом файле с названием «stats.txt». В файле хранится информация о каждом сеансе тренировки в виде 16 строк для каждого сеанса (или 15 строк, если отсутствует позиция с отношением количества правильных ответов к неправильным, в случае отсутствия неправильных ответов): дата и время завершения тренировки, имя пользователя, количество выполненных упражнений, количество использованных подсказок, количество повторений, количество правильно выполненных упражнений, количество правильно выполненных упражнений с подсказкой, процент правильно выполненных упражнений, количество неправильно выполненных упражнений, количество неправильно выполненных упражнений с подсказкой, процент неправильно выполненных упражнений, серия правильных ответов максимальной длины, серия неправильных ответов максимальной длины, средний размер серии правильных ответов, отношение количества правильных ответов к количеству неправильных и строка, состоящая из 40 символов „-“, являющаяся разделителем между данными о сеансах. Пример файла «stats.txt» приведен в *Приложении 3*.

1.2.2 Диаграмма классов

В ходе анализа предметной области была разработана диаграмма классов. Она представлена в *Приложении 2* в нотации UML.

Trenirovka — класс, реализующий функциональную часть тренажера. **Trenirovka** включает в себя:

- массив с 8-мью первыми простыми числами;
- нижний предел генерации числа-задания;
- верхний предел генерации числа-задания;
- составное число-задание, которое необходимо разложить на простые множители;
- массив, состоящий из простых множителей числа-задания;
- массив, состоящий из простых множителей — ответ пользователя.

Методы:

- функция для получения простого числа;
- функция генерации составного числа-задания;
- функция, возвращающая задание (упражнение) для интерфейсной части тренажера в виде широкой строки;

- функция, возвращающая подсказку для интерфейсной части тренажера в виде широкой строки;
- функция проверки ответа пользователя;
- функция очистки массивов внутри класса.
- функция очистки массива – ответа пользователя.

Класс **Stats** реализует сбор и сохранение статистики. Этот класс включает в себя:

- имя файла;
- имя пользователя;
- булева переменная — была ли активирована подсказка;
- общее количество выполненных упражнений;
- количество использованных подсказок;
- количество повторенных заданий;
- количество правильно решенных упражнений;
- количество правильно решенных упражнений с подсказкой;
- процент правильно решенных упражнений;
- количество неправильно решенных упражнений;
- количество неправильно решенных упражнений с подсказкой;
- процент неправильно решенных упражнений;
- текущая серия правильных ответов;
- максимальная серия правильных ответов;
- текущая серия неправильных ответов;
- максимальная серия неправильных ответов;
- средняя длина серии правильных ответов;
- сумма всех серий правильных ответов;
- количество серий правильных ответов;
- отношение правильных ответов к неправильным.

Методы:

- функция для проверки на то, является ли текущая серия правильных или неправильных ответов максимальной;
- функция, увеличивающая количество серий правильных ответов при окончании очередной серии;
- функция, вычисляющая процент правильных ответов;
- функция, вычисляющая процент неправильных ответов;
- функция, вычисляющая среднюю длину правильных ответов;
- функция, вычисляющая отношение правильных ответов к неправильным;

- функция, увеличивающая счетчик общего количества выполненных упражнений;
- функция, увеличивающая счетчик количества использованных подсказок;
- функция, увеличивающая счетчик количества повторенных упражнений;
- функция, увеличивающая счетчик количества правильно выполненных упражнений;
- функция, увеличивающая счетчик количества неправильно выполненных упражнений;
- функция, сохраняющая статистику в файл;
- функция, возвращающая статистику для интерфейсной части в виде широкой строки из файла;
- функция, проверяющая имя данного сеанса тренировки (если имя по умолчанию, значит тренировка не была начата или уже была закончена);
- функция, меняющая имя пользователя.

1.2.3 Алгоритм генерации числа

Дано: *number_array* – массив простых множителей числа для задания, *predel* – предел для числа-задания, *end* – булева переменная, условие для цикла, если переменная истина, то прекратить генерацию числа, *c* – число для задания.

сгенерировать случайное число в диапазоне от 30 до 1000 и сохранить число в переменной *predel*

пока (*end* = ложь)

| переменная *prime* = получить случайное простое число от 2 до 19 из массива простых чисел

| переменная *new_c* = *c* * *prime*

| если *new_c* <= *predel* то

| | записать в массив *number_array* число *prime*

| | *c* = *new_c*

| иначе

| *end* = истина

| конец ветвления

конец цикла

вернуть *c*

1.2.4 Алгоритм генерации и проверки задания

Дано: *c* – число для задания, *number_array* – массив простых множителей числа для задания, *answer_array* – массив простых множителей – ответ пользователя.

сгенерировать число *c* для задания (алгоритм 1.2.3)
вывести число-задание для пользователя
запросить у пользователя ответ в виде строки, где через пробел записаны простые множители
занести простые множители из строки в массив *answer_array*
отсортировать массивы *number_array* и *answer_array*
вернуть результат сравнения *number_array* и *answer_array*

1.2.5 Основной алгоритм

вывести условия меню
пока пользователь не дал команду завершить программу
| если пользователь дал команду «начать тренировку» то
| | пока пользователь не дал команду «завершить тренировку»
| | | вызвать алгоритм генерации и проверки задания 1.2.4
| | | если задание выполнено верно, то сообщить пользователю о том, что задание
| | | решено правильно и увеличить счетчик количества правильно решенных
| | | заданий для статистики
| | | иначе сообщить о том, что задание решено неправильно и увеличить счетчик
| | | количества неправильно решенных заданий для статистики
| | | конец ветвления
| | если пользователь дал команду «показать подсказку», то вывести сообщение с
| | | подсказкой и увеличить счетчик количества использованных подсказок
| | если пользователь дал команду «повторить задание», то вывести задание
| | | повторно без генерации нового задания и увеличить счетчик количества
| | | повторенных заданий
| | конец цикла
| | сохранить статистику тренировки в файл
| конец ветвления
| если пользователь дал команду «перейти к просмотру статистики» то вывести
| | статистику из файла на экран
конец цикла

1.3 Входные и выходные данные

Входные данные:

- Выбор пунктов меню.

- Ответ пользователя на вопрос тренировки.

Выходные данные:

- Натуральное число, составляющее задание
- Статистика, сохраненная в результате работы тренажера

1.4 Системные требования

Рекомендуемая конфигурация:

- Intel-совместимый процессор с частотой не менее 1,6 ГГц;
- не менее 512 МБ ОЗУ;
- не менее 20 МБ свободного места на диске;
- дисковод CD-ROM/DVD-ROM.

Операционная система: Windows XP (x86) с пакетом обновления 3 (SP3) или более поздние. Среда разработки – интегрированная среда Qt Creator (версия Qt 5.4), язык C++ (стандарт C++ 11 и более поздние).

2 РАБОЧИЙ ПРОЕКТ

2.1 Общие сведения о работе системы

Программный продукт разработан в интегрированной среде Qt Creator (версия Qt 5.4) на языке C++ (стандарт C++ 11 и более поздние), с использованием компилятора MinGW GCC (версия 7.3.0). Программа работает под управлением операционной системы Windows XP (x86) Professional (SP3) и более поздними.

2.2 Функциональное назначение программного продукта

Разработанный программный продукт предназначен для повышения качества знаний пользователей по теме «Разложение натурального числа на простые множители», а также для снижения нагрузки на преподавателя. Программа имеет следующие функциональные возможности:

- предоставление пользователю задания;
- проверка правильности вычисления пользователем предложенного задания;
- прекращение тренировки по желанию пользователя;
- сбор и сохранение статистики после прекращения тренировки.

Программа имеет следующие функциональные ограничения:

- необходимо вводить только натуральные числа;
- числа-задания должны быть целыми числами из диапазона [30; 1000].

2.3 Инсталляция и выполнение программного продукта

Программу можно запустить с помощью проводника операционной системы, запустив скомпилированный исполняемый .exe файл.

Для выполнения программы необходимо:

1. Скопировать на жесткий диск компьютера папку Practice, содержащую исполняемый файл Practice.exe и папку source с исходными текстами программы.
2. Запустить исполняемый файл Practice.exe.

2.4 Описание программы

Программа включает 2 класса – класс Trenirovka, реализующий функциональную часть тренажера и класс Stats для сбора и сохранения статистики. Также программа включает главную процедуру, функцию для регистрации класса главного окна, функцию для инициализации главного окна, а так же функции для обработки сообщений для следующих окон: главное окно, дочернее окно с главным меню, дочернее окно с тренажером, дочернее

окно со статистикой, диалоговое модальное окно с инструкцией и диалоговое модальное окно со сведениями о проекте и авторе.

В таблице 2.1 приведено описание класса `Trenirovka` (заголовочный файл `trenirovka.h`). Для реализации строк были выбраны широкие `std::wstring` вместо обычных `std::string` для лучшей совместимости с юникодом и интерфейсной (WinAPI) частью программы.

Таблица 2.1 – Описание класса `Trenirovka`

Поле	Тип	Назначение
<code>prime_array</code>	<code>std::array<uint8_t,8></code>	Массив простых чисел от 2 до 19
<code>low_predel</code>	<code>uint16_t</code>	Нижний предел генерации числа
<code>up_predel</code>	<code>uint16_t</code>	Верхний предел генерации числа
<code>chislo</code>	<code>uint16_t</code>	Число для задания
<code>number_array</code>	<code>std::vector<uint16_t></code>	Массив, состоящий из простых множителей числа-задания
<code>answer_array</code>	<code>std::vector<uint16_t></code>	Массив-ответ пользователя
Метод		Описание
<code>Trenirovka()</code>		Конструктор по умолчанию для класса
<code>uint16_t getPrimeNumber() const</code>		Метод, возвращающий случайное простое число из массива <code>prime_array</code>
<code>uint16_t createNumber()</code>		Метод, формирующий массив <code>number_array</code> и возвращающий число-задание для метода <code>getZadanie</code>
<code>void parseAnswer(const std::wstring &str)</code>		Метод, разбирающий строку с ответом и заполняющий массив <code>answer_array</code>
<code>std::wstring getZadanie()</code>		Метод, возвращающий строку с заданием (с числом, которое надо разложить)
<code>std::wstring getHint()</code>		Метод, возвращающий строку с подсказкой (со случайным простым множителем из массива <code>number_array</code>)
<code>bool checkAnswer(const std::wstring &str)</code>		Метод, проверяющий ответ пользователя
<code>void clear_all()</code>		Метод, очищающий массивы <code>number_array</code> и <code>answer_array</code>
<code>void clear_answer_array()</code>		Метод, очищающий массив <code>answer_array</code> (ответ пользователя)

В таблице 2.2 приведено описание класса Stats (заголовочный файл stats.h).

Таблица 2.2 – Описание класса Stats

Поле	Тип	Назначение
filename	std::string	Название файла со статистикой
username	std::wstring	Имя пользователя
hintActivated	bool	Булева переменная – была ли активирована подсказка
count_complited	uint64_t	Количество выполненных упражнений
count_hint	uint64_t	Количество использованных подсказок
count_repeat	uint64_t	Количество повторенных заданий
count_right	uint64_t	Количество правильно выполненных упражнений
count_hinted_right	uint64_t	Количество правильно выполненных упражнений с подсказкой
percent_right	double	Процент правильно выполненных упражнений
count_wrong	uint64_t	Количество неправильно выполненных упражнений
count_hinted_wrong	uint64_t	Количество неправильно выполненных упражнений с подсказкой
percent_wrong	double	Процент неправильно выполненных упражнений
current_series_right	uint64_t	Текущая серия правильных ответов
series_right	uint64_t	Серия правильных ответов максимальной длины
current_series_wrong	uint64_t	Текущая серия неправильных ответов
series_wrong	uint64_t	Серия неправильных ответов максимальной длины
count_series_right	uint64_t	Количество серий правильных ответов
avg_series_right	double	Средний размер серии правильных ответов
summa_series_right	uint64_t	Сумма всех серий правильных ответов
ratio_right_wrong	double	Отношение количества правильных ответов к количеству неправильных
Метод		Описание
Stats()		Конструктор по умолчанию для класса
void calculate_percent_right()		Функция, вычисляющая процент правильных ответов

Продолжение таблицы 2.2

Метод	Описание
void check_for_max_series()	Функция для проверки на то, является ли текущая серия правильных или неправильных ответов максимальной
void add_series_right()	Функция, увеличивающая количество серий правильных ответов при окончании очередной серии
void calculate_percent_wrong()	Функция, вычисляющая процент неправильных ответов
void calculate_avg_series_right()	Функция, вычисляющая среднюю длину правильных ответов
void calculate_ratio()	Функция, вычисляющая отношение правильных ответов к неправильным
void complete()	Функция, увеличивающая счетчик общего количества выполненных упражнений
void hint()	Функция, увеличивающая счетчик количества использованных подсказок
void repeat()	Функция, увеличивающая счетчик количества повторенных упражнений
void right()	Функция, увеличивающая счетчик количества правильно выполненных упражнений
void wrong()	Функция, увеличивающая счетчик количества неправильно выполненных упражнений
void statsToFile()	Функция, сохраняющая статистику в файл
bool checkName()	Функция, проверяющая имя данного сеанса тренировки (если имя по умолчанию, значит тренировка не была начата или уже была закончена)
void changeName(const std::wstring &ws)	Функция, меняющая имя пользователя
std::wstring statsFromFile()	Функция, возвращающая статистику для интерфейсной части в виде широкой строки из файла

В таблице 2.3 приведены функции и процедуры, используемые в программе.

Таблица 2.3 – Функции и процедуры программы

Прототип	Назначение
int APIENTRY wWinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPTSTR lpCmdLine, int nCmdShow)	Главная функция
ATOM MyRegisterClass(HINSTANCE hInstance)	Функция, регистрирующая класс окна перед инициализацией
BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)	Функция, инициализирующая главное окно программы
LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений в главном окне
inline void GetNewZadanie()	Функция, генерирующая новое задание с помощью класса Trenirovka и выводящая задание на экран
inline void GetPrevZadanie()	Функция, выводящая предыдущее задание на экран
LRESULT CALLBACK MenuWndProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений для дочернего окна с главным меню
LRESULT CALLBACK TrenirovkaWndProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений для дочернего окна с тренажером
LRESULT CALLBACK StatsWndProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений для дочернего окна со статистикой
LRESULT CALLBACK About(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений для дочернего окна «О программе»
LRESULT CALLBACK getNameWndProc(HWND hDlg, UINT message, WPARAM wParam, LPARAM lParam)	Функция — обработчик сообщений для дочернего окна, которое получает имя пользователя перед началом тренировки

В таблице 2.4 приведены важнейшие переменные, используемые в программе.

Таблица 2.4 – Важнейшие переменные программы

Имя	Тип	Назначение
hInst	HINSTANCE	Текущий экземпляр приложения
szTitle[100]	WCHAR	Текст строки заголовка
szWindowClass[100]	WCHAR	Имя класса главного окна
mainWindow	HWND	Дескриптор главного окна
menuWindow	HWND	Дескриптор окна с главным меню
trenirovkaWindow	HWND	Дескриптор окна с тренажером
statsWindow	HWND	Дескриптор окна со статистикой
trenirovka	Trenirovka	Объект класса Trenirovka
stats	Stats	Объект класса Stats

2.5 Разработанные меню и интерфейсы

После запуска программы на выполнение появится окно, содержащее главное меню (рис. 2.1), который помимо самого меню содержит информацию о программе.

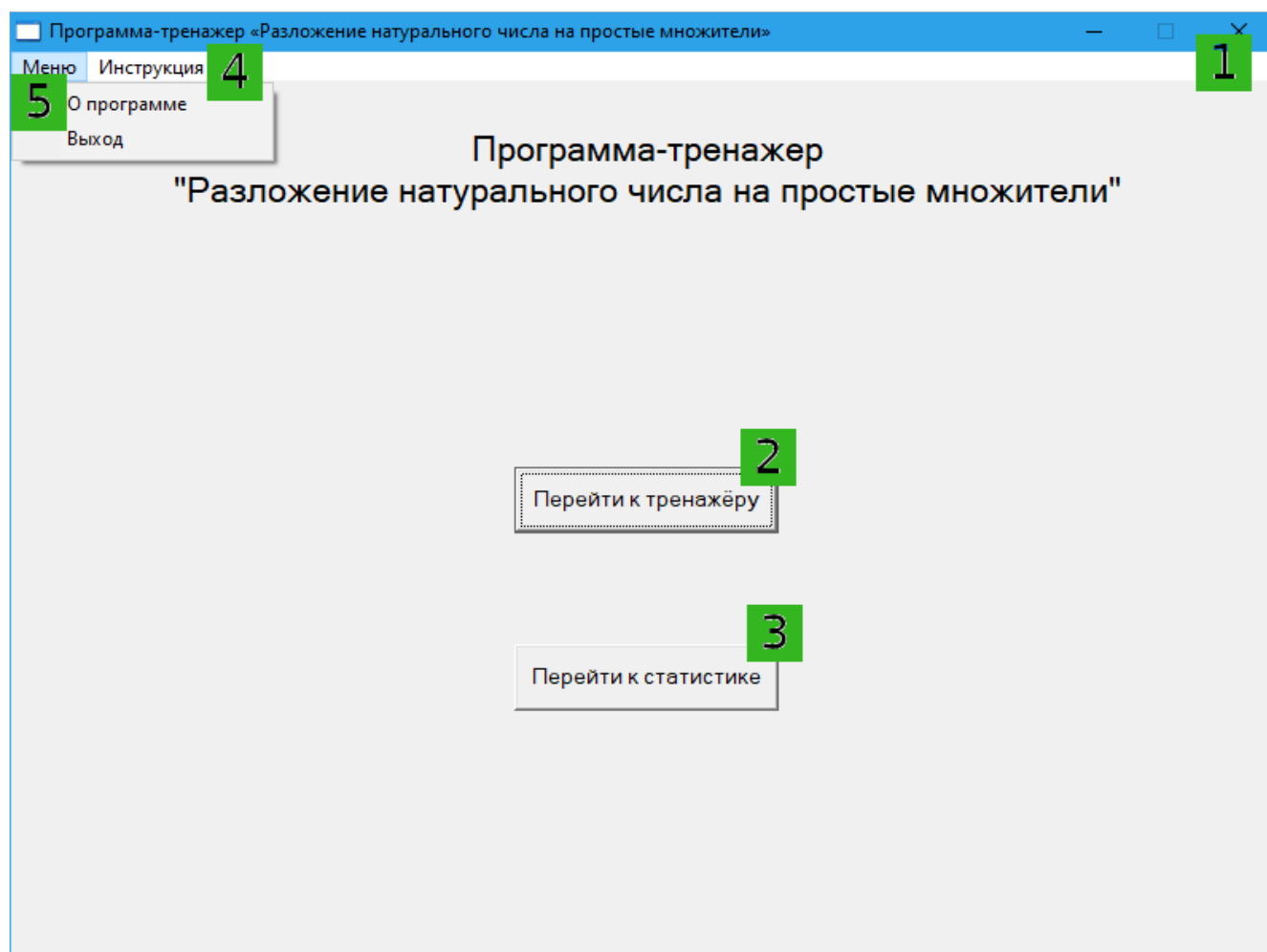


Рисунок 2.1 – Окно программы с главным меню

Кнопка «2» позволяет перейти к тренировке, кнопка «3» – к просмотру статистики, пункт меню «4» – к окну с инструкцией, пункт меню «5» контекстного меню – к окну со сведениями об авторе программы, а кнопка «1» - завершает работу программы.

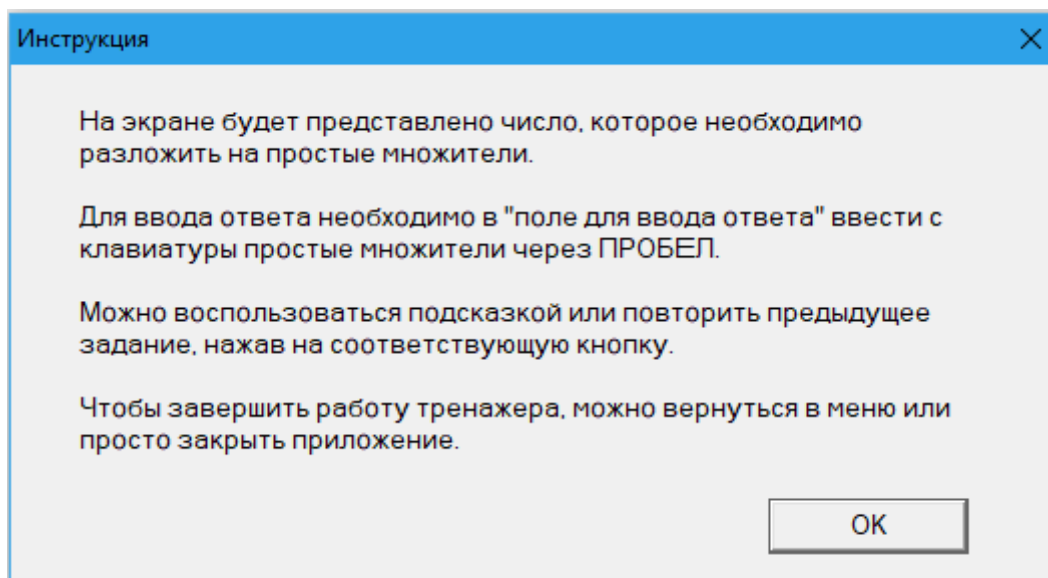


Рисунок 2.2 – Окно «Инструкция»

При нажатии на пункт меню «4» на экран показывается всплывающее модальное окно с инструкцией (рис. 2.2).

Нажав кнопку «ОК» можно закрыть это окно. При нажатии на пункт меню «5» показывается всплывающее модальное окно «О программе» (рис. 2.3).

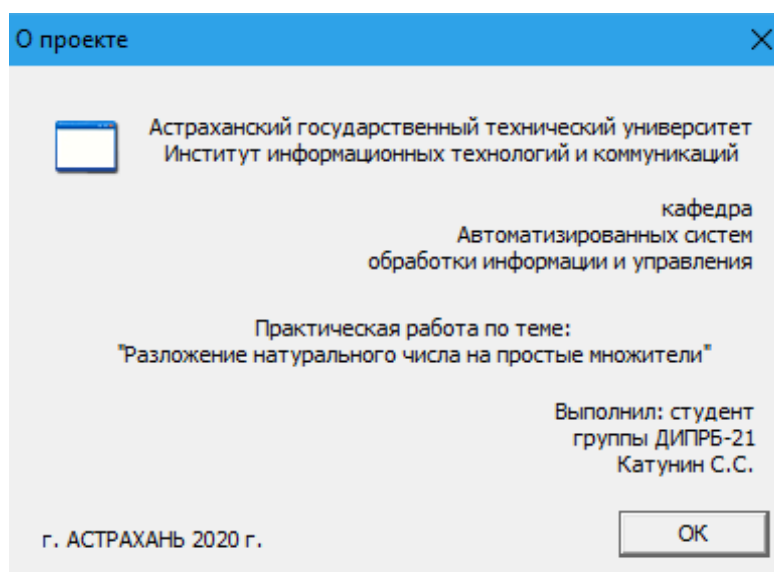


Рисунок 2.3 – Окно со сведениями о проекте

При нажатии на кнопку «2» пользователя встречает диалоговое окно, в котором необходимо ввести имя пользователя (рис. 2.4).

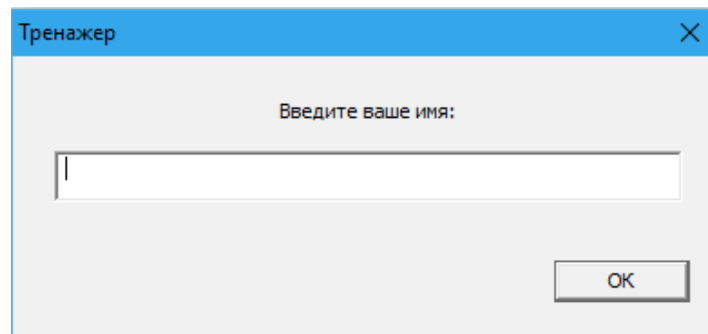


Рисунок 2.4 – Окно, в котором необходимо ввести имя

Текстовое поле используется для ввода имени пользователя.

После ввода (если оставить поле пустым, то программа задаст пользователю имя «Аноним») и нажатия на кнопку «ОК», на экране показывается окно с тренажером (рис. 2.5).

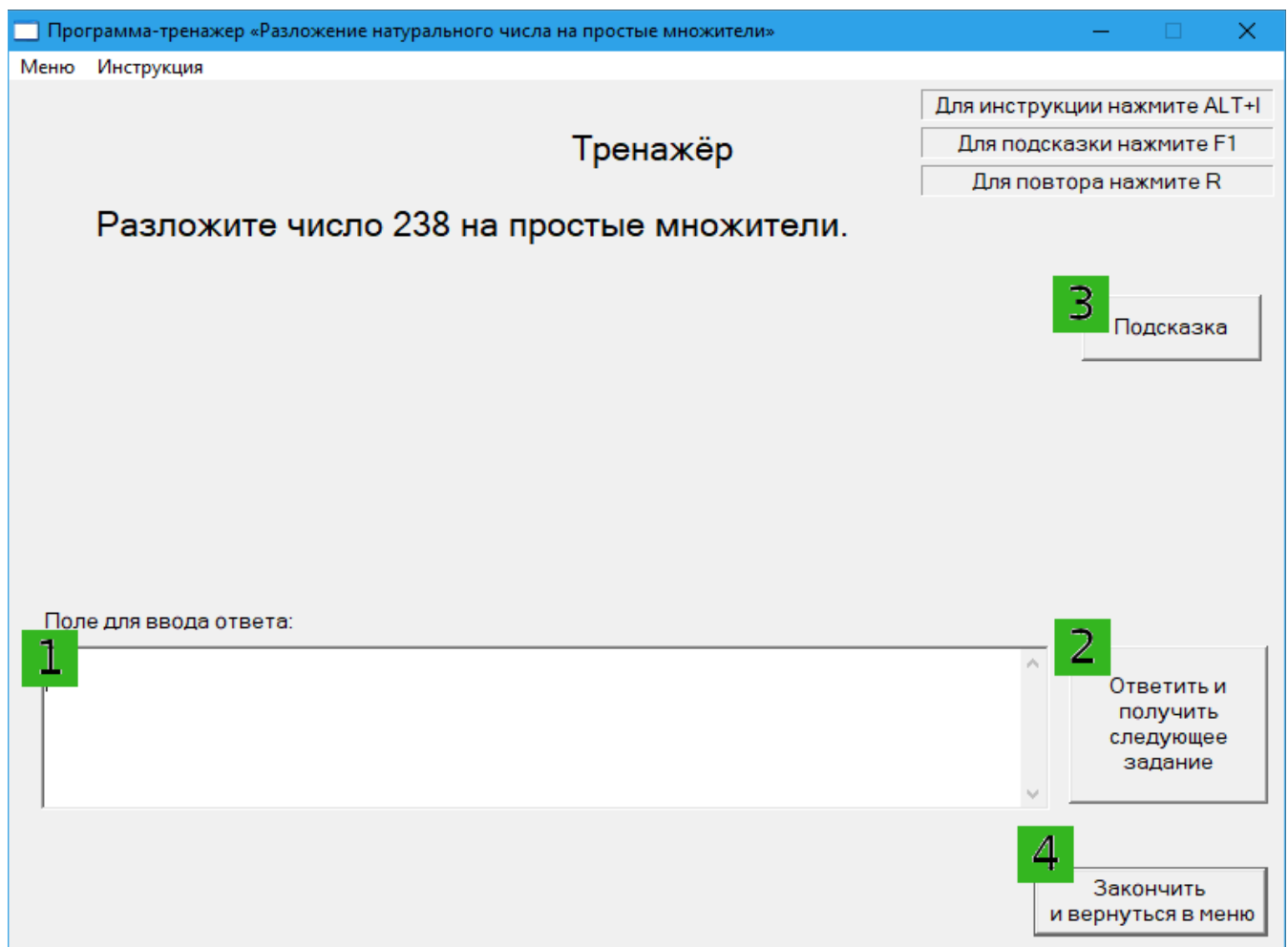


Рисунок 2.5 – Окно с тренажером

В текстовое поле «1» осуществляется ввод ответа пользователя на задание, с помощью кнопки «2» пользователь может ответить на задание и получить следующее, кнопка «3» используется для получения подсказки, а кнопка «4» - заканчивает тренировку и осуществляет возврат в окно с главным меню (рис. 2.1).

При нажатии на кнопку «3» на экран показывается всплывающее модальное окно (месседжбокс) с подсказкой (рис. 2.6).

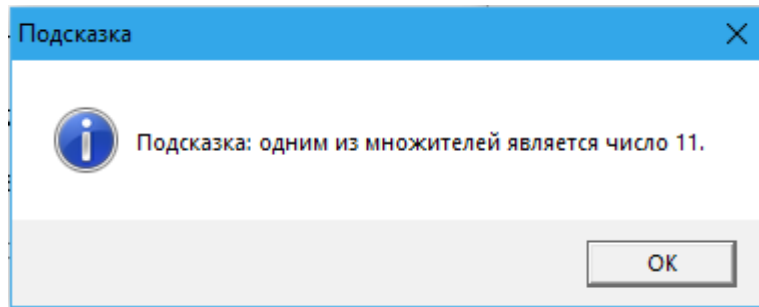


Рисунок 2.6 – Окно с подсказкой

В случае ввода неправильного ответа и нажатия на кнопку «2» содержимое окна с тренажером изменяется: поле для ввода ответа «1» становится недоступным пользователю и появляется кнопка для повтора неправильного решенного задания (рис. 2.7).

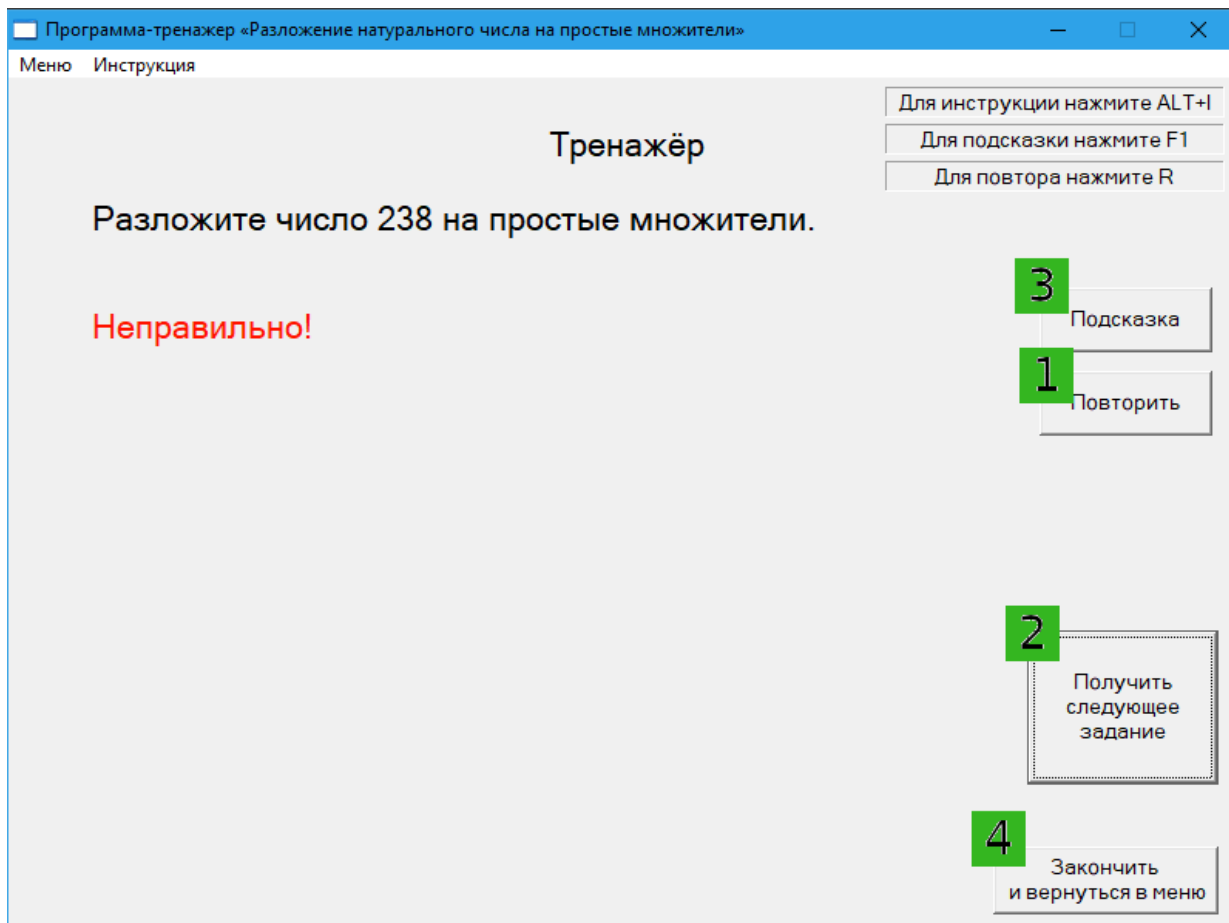


Рисунок 2.7 – Окно с тренажером в случае неправильного ответа

Если пользователь нажмет на пункт «1» или «2», то содержимое окно сменится (см. рис. 2.5), где снова будет доступно поле для ввода ответа. Отличие в том, что если была нажата кнопка «1», то задание остается неизменным, а если «2», то будет сгенерировано новое задание.

Если в главном меню программы нажать на кнопку «3», то окно с главным меню сменится на окно со статистикой (рис. 2.8). В нем отображены все данные о сеансах тренировок: дата, имя, общее количество решенных заданий, количество правильно и неправильно решенных заданий, количество повторов, использованных подсказок, проценты правильно и неправильно выполненных упражнений, а так же серии правильных и неправильных ответов.

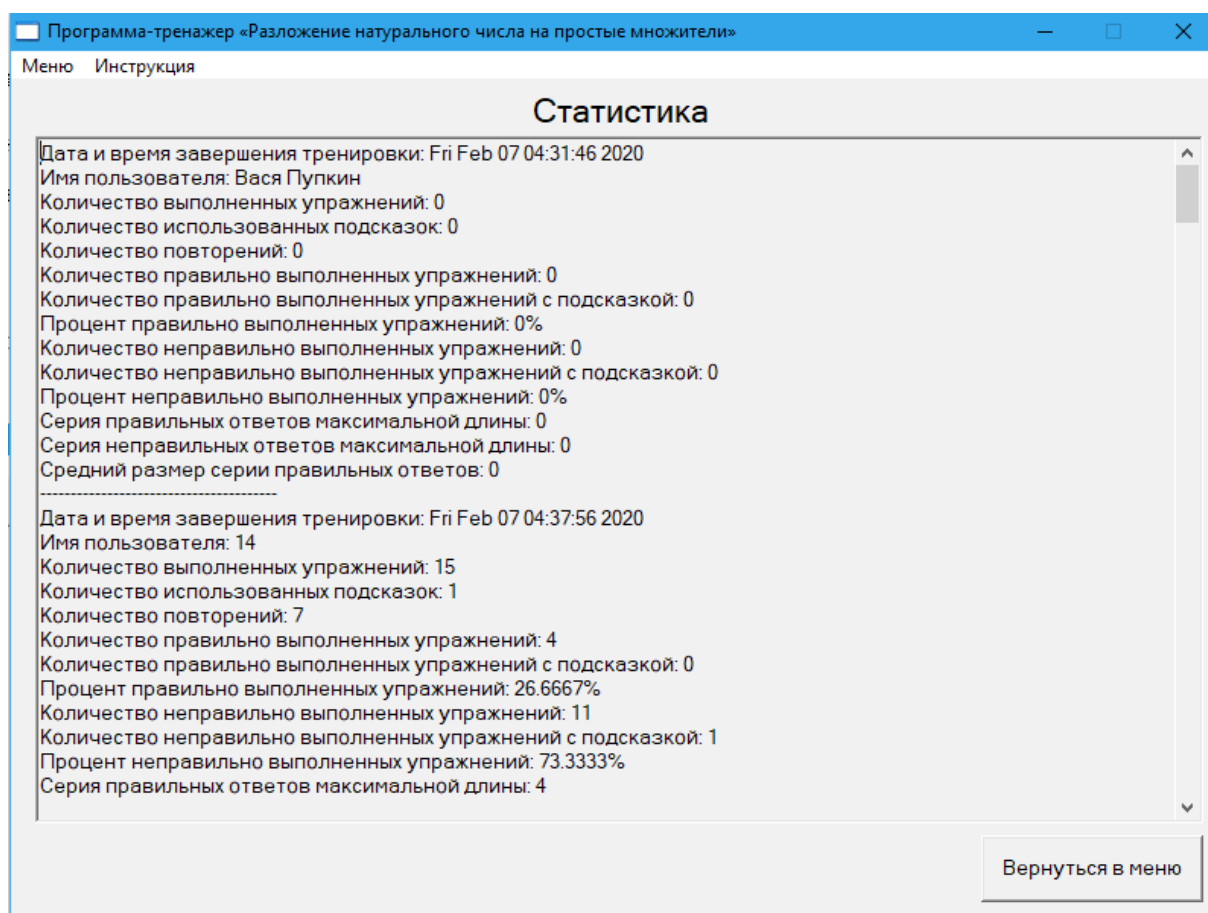


Рисунок 2.8 – Окно со статистикой

С помощью кнопки «Вернуться в меню» можно вернуться в главное меню программы (рис. 2.1).

2.6 Сообщения системы

Программа сообщает об ошибках с помощью месседжбоксов. На рисунке 2.9 представлено сообщение об ошибке, возникающее в случае, если программа не может открыть файл со статистикой при переходе к просмотру к статистики.

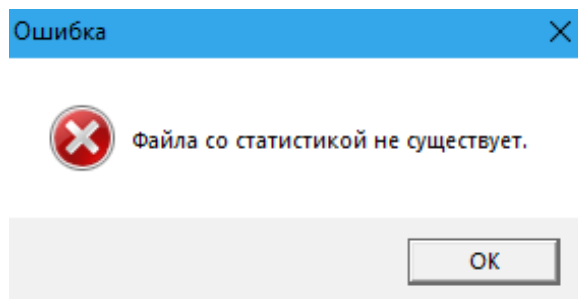


Рисунок 2.9 – Ошибка: файла со статистикой не существует.

В случае появления других сообщений следует обратиться к разработчику.

3 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

1. Запустить программу на выполнение. Появится меню (см. рис. 2.1).
2. Нажать на кнопку «Инструкция» в контекстном меню, убедиться что было отображено окно с инструкцией (см. рис. 2.2).
3. Нажать на кнопку «О программе» в контекстном меню, убедиться что было отображено окно со сведениями о проекте (см. рис. 2.3).
4. Нажать на кнопку «Перейти к тренажёру» в главное меню, убедиться что было запрошено имя пользователя перед тренировкой (см. рис. 2.4).
5. Ввести имя, нажать кнопку «ОК», убедиться, что было отображено окно с тренажёром (см. рис. 2.5).
6. Ввести правильный ответ в поле для ввода ответа и убедиться, что после нажатия на кнопку «Ответить и получить следующее задание» на экране сменилось задание и была отображена надпись «Правильно!».
7. Нажать на кнопку «Подсказка», убедиться, что подсказка является простым множителем числа-задания (см. рис. 2.6).
8. Ввести неправильный ответ в поле для ввода ответа и убедиться, что после нажатия на кнопку «Ответить и получить следующее задание» на экране сменилось задание и была отображена надпись «Неправильно!».
9. Нажать на кнопку «Повторить», убедиться, что было отображено предыдущее задание, которое было выполнено неправильно и выполнить его (см. пункт 6).
10. Повторять пункты 6, 7, 8, 9 в разной комбинации, например, чтобы получить правильный ответ с подсказкой, без подсказки, и так далее.
11. Нажать на кнопку «Закончить и вернуться в меню», убедиться, что был осуществлен возврат в главное меню (см. рис. 2.1).
12. Нажать на кнопку «Перейти к статистике» и убедиться, что был осуществлен переход к окну со статистикой (см. рис. 2.7).
13. Убедиться, что в статистике была добавлена запись о прошлом сеансе, и что статистика корректна.
14. Повторить пункты 4, 5, 10, 11, 12, 13.

ЗАКЛЮЧЕНИЕ

В результате выполнения индивидуального задания на практику исследована предметная область, спроектирован оконный интерфейс приложения, произведен реинжиниринг и рефакторинг программного продукта, разработанного в рамках курсового проекта по дисциплине «Программирование и информатика» (1 семестр). Тренажёр предлагает задания на умение раскладывать число на простые множители и проверяет правильность ответа. Тренировка продолжается до тех пор, пока пользователь не решит ее закончить. Также программа собирает статистику по всем выполненным упражнениям и сохраняет её в файл.

Программа отвечает поставленным требованиям и может быть использована для обучения школьников средних классов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Белов С.В., Лаптев В.В., Морозов А.В., Толасова В.В., Мамлеева А.Р. Требования к оформлению студенческих работ. – Астрахань, АГТУ, 2017. 104 с.
2. Ишмухаметов Ш. Т. Методы факторизации натуральных чисел: учебное пособие — Казань: Казанский университет, 2011. — 190 с.
3. Щупак Ю. А. Win32 API. Разработка приложений Для Windows. – СПб.: Питер, 2008. – 592 с.: ил.
4. Петзолд Ч. Программирование для Windows 95. – СПб.: Питер, 1997. – 242 с.: ил.
5. Окна на чистом WinAPI. Или просто о сложном – [Электронный ресурс] режим доступа: <https://habr.com/ru/post/352096> (29.01.2020)
6. MinGW-w64 - for 32 and 64 bit Windows / Wiki2 / Unicode apps – [Электронный ресурс] режим доступа: [https://sourceforge.net/p/mingw-w64/wiki2/Unicode apps/](https://sourceforge.net/p/mingw-w64/wiki2/Unicode%20apps/) (29.01.2020)

ПРИЛОЖЕНИЕ 1**Техническое задание на разработку программы-тренажера
«Разложение натурального числа на простые множители»**

по дисциплине «Программирование и информатика»

Направление 09.03.04 Программная инженерия

Исполнитель: студент гр. ДИПР621 Катунин С.С.

1 Назначение, цели и задачи разработки

Цель разработки – автоматизация обучения и контроля знаний по теме «Разложение натурального числа на простые множители».

Назначение разработки:

- повышение качества знаний пользователей;
- снижение нагрузки на преподавателя.

Основные задачи, решаемые разработчиком в процессе выполнения курсового проекта:

- анализ предметной области;
- разработка программного продукта в соответствии с требованиями;
- документирование проекта в соответствии с установленными требованиями.

2 Характер разработки: прикладная квалификационная работа.**3 Основания для разработки**

- Учебный план направления 09.03.04 «Программная инженерия» 2018 года набора.
- Рабочая программа дисциплины Программирование и информатика».
- Распоряжение по кафедре АСОИУ № ____ от «__» _____ 201__ г.

4 Плановые сроки выполнения – осенний семестр 2019/20 учебного года:

Начало « 06 » сентября 2018 г.

Окончание « 18 » декабря 2018 г.

5 Требования к проектируемой системе**5.1 Требования к функциональным характеристикам**

Проектируемая система представляет собой графическое приложение и должна обеспечивать выполнение следующих основных функций:

- предоставление пользователю задания;
- проверка правильности вычисления пользователем предложенного задания;
- прекращение тренировки по желанию пользователя;
- сбор статистики по всем выполненным упражнениям и подсчёт следующих данных:
 - ✓ количество выполненных упражнений;

- ✓ количество использованных подсказок;
 - ✓ количество повторенных заданий;
 - ✓ количество и процент правильно выполненных упражнений;
 - ✓ количество и процент неправильно выполненных упражнений;
 - ✓ серию правильных ответов максимальной длины;
 - ✓ серию неправильных ответов максимальной длины;
 - ✓ средний размер серии правильных ответов;
 - ✓ при наличии неправильных ответов – отношение количества правильных ответов к количеству неправильных.
- **Интерфейс программы:** текст русский, шрифты кириллический и латинский, заголовки, термины и другая важная информация выделены цветом.

Система имеет функциональные ограничения:

- необходимо вводить только натуральные числа;
- числа-задания должны быть целыми числами из диапазона [30; 1000].

5.2 Требования к эксплуатационным характеристикам

Программа не должна аварийно завершаться при любых действиях пользователя.

Время реакции программы на действия пользователя не должно превышать 10 секунд.

5.3 Требования к программному обеспечению:

Средства разработки: интегрированная среда Qt Creator (версия Qt 5.4), язык C++ (стандарт C++ 11 и более поздние).

Операционная система: Windows XP (x86) с пакетом обновления 3 (SP3) или более поздние.

5.4 Требования к аппаратному обеспечению:

Рекомендуемая конфигурация:

- Intel-совместимый процессор с частотой не менее 1,6 ГГц;
- не менее 512 МБ ОЗУ;
- не менее 20 МБ свободного места на диске;
- Дисковод CD-ROM/DVD-ROM.

6 Стадии и этапы разработки

6.1 Эскизный проект (ЭП)

- Анализ предметной области.
- Подготовка проектной документации.

6.2 Технический проект (ТП)

- Разработка структур и форм представления данных.

- Разработка структуры программного комплекса.
- Подготовка пояснительной записки.

6.3 Рабочий проект (РП)

- Программная реализация.
- Тестирование и отладка программы.
- Подготовка программной и эксплуатационной документации.

6.4 Эксплуатация (Э)

Описание и анализ результатов проведенного исследования.

7 Требования к документированию проекта

К защите курсового проекта должны быть представлены следующие документы:

- Пояснительная записка к курсовому проекту:
- Презентация доклада.
- Программа, презентация и пояснительная записка к курсовому проекту на оптическом носителе.

Требования к структуре документов определены соответствующими стандартами ЕСПД.

Требования к оформлению определены соответствующими методическими указаниями.

8 Порядок контроля и приемки

Контроль выполнения курсового проекта проводится руководителем поэтапно в соответствии с утвержденным графиком выполнения проекта.

На завершающем этапе руководитель осуществляет нормоконтроль представленной исполнителем документации и принимает решение о допуске (недопуске) проекта к защите.

Защита курсового проекта проводится комиссией в составе не менее двух человек, включая руководителя проекта.

В процессе защиты проекта исполнитель представляет документацию, делает краткое сообщение по теме разработки и демонстрирует ее программную реализацию.

При выставлении оценки учитывается:

- степень соответствия представленной разработки требованиям технического задания;
- качество программной реализации, документации и доклада по теме проекта;
- соблюдение исполнителем графика выполнения курсового проекта.

9 Литература

1. Ишмухаметов Ш. Т. Методы факторизации натуральных чисел: учебное пособие — Казань: Казанский университет, 2011. — 190 с.
2. Щупак Ю. А. Win32 API. Разработка приложений Для Windows. – СПб.: Питер, 2008. – 592 с.: ил.

ПРИЛОЖЕНИЕ 2

Диаграмма классов

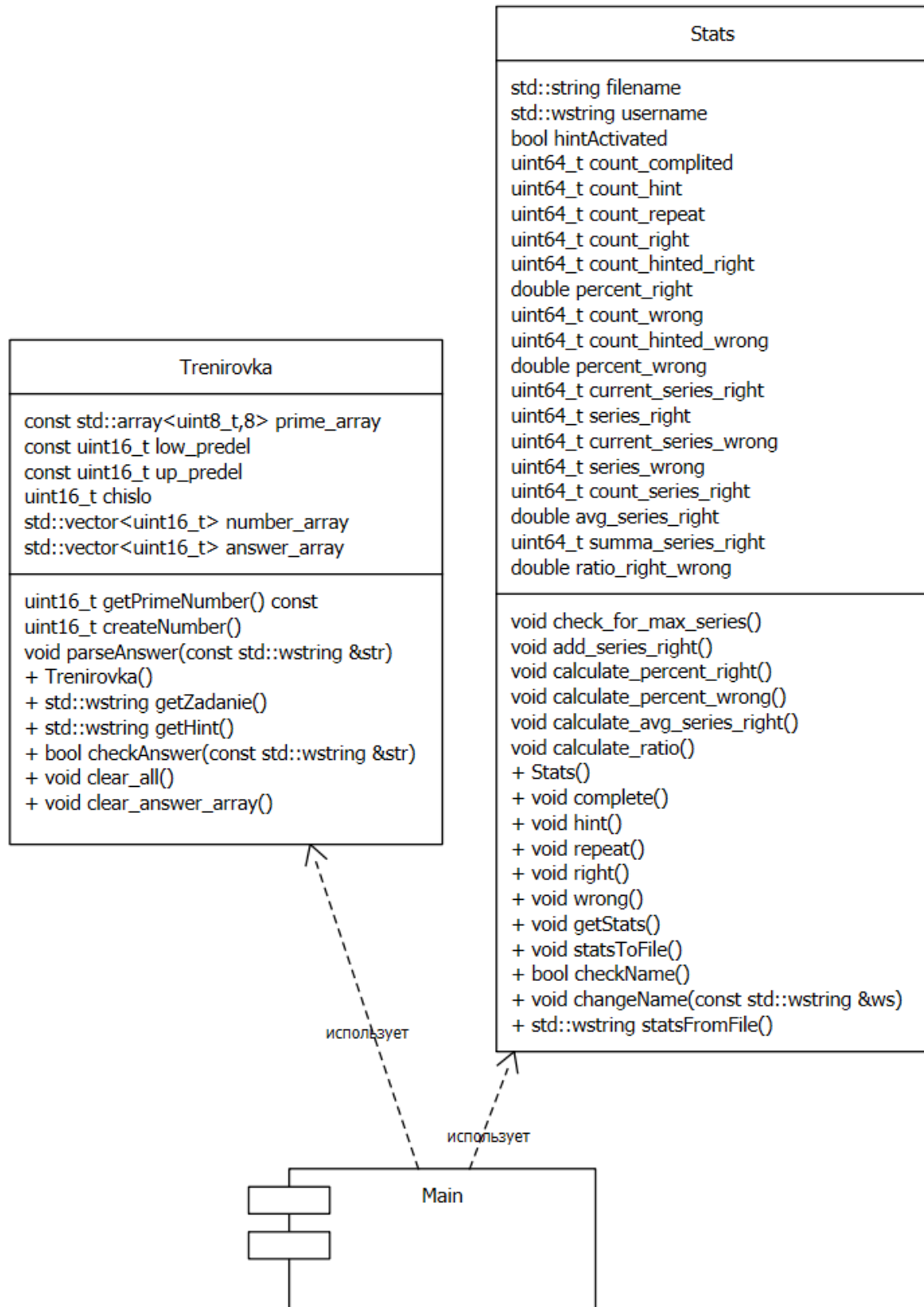


Рисунок П.2.1 – Диаграмма классов

ПРИЛОЖЕНИЕ 3

**Файл со статистикой
(примерное содержание файла stats.txt)**

Дата и время завершения тренировки: Fri Feb 06 16:31:46 2020
Имя пользователя: Вася Пупкин
Количество выполненных упражнений: 11
Количество использованных подсказок: 6
Количество повторений: 1
Количество правильно выполненных упражнений: 9
Количество правильно выполненных упражнений с подсказкой: 6
Процент правильно выполненных упражнений: 81.8182%
Количество неправильно выполненных упражнений: 2
Количество неправильно выполненных упражнений с подсказкой: 0
Процент неправильно выполненных упражнений: 18.1818%
Серия правильных ответов максимальной длины: 5
Серия неправильных ответов максимальной длины: 1
Средний размер серии правильных ответов: 3
Отношение количества правильных ответов к количеству неправильных:
4.5

Дата и время завершения тренировки: Fri Feb 06 17:19:12 2020
Имя пользователя: Кеша
Количество выполненных упражнений: 5
Количество использованных подсказок: 2
Количество повторений: 0
Количество правильно выполненных упражнений: 5
Количество правильно выполненных упражнений с подсказкой: 2
Процент правильно выполненных упражнений: 100%
Количество неправильно выполненных упражнений: 0
Количество неправильно выполненных упражнений с подсказкой: 0
Процент неправильно выполненных упражнений: 0%
Серия правильных ответов максимальной длины: 5
Серия неправильных ответов максимальной длины: 0
Средний размер серии правильных ответов: 5
