Содержание

Введение 7

1 Характеристика места практики 9

2 Сведения об имеющейся вычислительной технике и программном обеспечении 11

3 Сведения о прохождении практики в качестве дублера инженерно-технического работника 14

4 Индивидуальное задание 15

4.1 Постановка задачи 15

4.2 Проектирование структуры программы 16

4.2.1 Desktop версия приложения 16

4.2.2 Мобильная версия приложения 16

4.3 Функциональное взаимодействие модулей программы 16

4.3.1 Desktop версия приложения 16

4.3.2 Мобильная версия приложения 18

4.4 Разработка unit-тестов 21

4.5 Работа с программой 22

4.5.1 Desktop версия приложения 22

4.5.2 Мобильная версия приложения 25

Заключение 29

Список использованных источников 31

Приложения

А – Основные процедуры desktop приложения 32

Б – Код класса MyDraw.java

В – Код unit-тестов

ВВЕДЕНИЕ

Целью производственной практики по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем является закрепление и совершенствование приобретенных в процессе обучения профессиональных умений обучающихся по специальности, развитие общих и профессиональных компетенций, адаптация к конкретным условиям деятельности предприятий и организаций, освоение современных производственных процессов, обучение трудовым приемам, операциям и способам выполнения трудовых процессов.

В процессе прохождения производственной практики необходимо будет решить следующие задачи:

* приобрести практический опыт разработки алгоритма решения поставленной задачи и реализации его средствами автоматизированного проектирования, использования инструментальных средств на этапе отладки программного продукта и проведения тестирования программного модуля по определенному сценарию;
* приобрести практический опыт анализа алгоритмов, в том числе с применением инструментальных средств, разработки кода программного продукта на основе готовой спецификации на уровне модуля, осуществления рефакторинга и оптимизации программного кода;
* приобрести практический опыт разработки мобильных приложений;
* сформировать профессиональные умения и навыки;
* приобрести практические навыки в будущей профессиональной деятельности при выполнении функций дублера инженерно-технических работников;
* выполнить индивидуальное задание.

В результате прохождения производственной практики необходимо будет подтвердить уровень овладения видом деятельности «Разработка модулей программного обеспечения для компьютерных систем» и следующими профессиональными компетенциями:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4 Выполнять тестирование программных модулей.

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.

ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.

Производственная практика проводится на базе АО “МЕГАФОН РИТЕЙЛ”

1 ХАРАКТЕРИСТИКА МЕСТА ПРАКТИКИ

Майкопский государственный гуманитарно-технический колледж федерального государственного бюджетного образовательного учреждения высшего образования «Адыгейский государственный университет» является обособленным структурным подразделением федерального государственного бюджетного образовательного учреждения высшего образования «Адыгейский государственный университет».

Юридический адрес МГГТК ФГБОУ ВО «АГУ»: 385007, Россия, Республика Адыгея, г. Майкоп, ул. 2-я Ветеранов, 1.

Местонахождение образовательной организации (колледжа): 385007,   
г. Майкоп, I корпус: ул. 2-я Ветеранов, 1; II, III корпус: ул. Солнечная, 60.

Юридический адрес АГУ: 385000, Россия, Республика Адыгея, г. Майкоп, ул. Университетская, 208.

Основными задачами МГГТК ФГБОУ ВО «АГУ» являются:

* удовлетворение потребностей личности в интеллектуальном, культурном и нравственном развитии посредством получения среднего и начального профессионального образования;
* удовлетворение потребностей общества в специалистах со средним и начальным профессиональным образованием; формирование у лиц, обучающихся в колледже, гражданской позиции и трудолюбия;
* развитие ответственности, сохранение и приумножение нравственных и культурных ценностей общества.

В колледже обучается более 3000 студентов по 18-и специальностям. Обучение осуществляется на бюджетной и договорной основе. Кроме того, на базе учреждения проводятся переподготовка и повышение квалификации кадров по различным направлениям. Деятельность коллектива ориентирована на внедрение в учебный процесс инновационных педагогических и информационных технологий, направленных на развитие самостоятельной учебно-исследовательской работы студентов. В образовательном процессе широко используется электронный образовательный ресурс и дистанционные технологии.

Актуальная информация о колледже представлена на сайте http://mggtk.ru, а также в социальных сетях.

Учебные занятия проводятся в соответствии с графиком учебного процесса. В колледже установлена шестидневная учебная неделя. Учебные занятия проводятся в 2 смены:

1 смена: 8 ч.30 мин. – 13 ч. 00 мин.

2 смена: 13 ч. 40 мин. – 17 ч. 50 мин.

Директор колледжа – Тепсаева З.М. кандидат педагогических наук, Почетный работник среднего профессионального образования Российской Федерации.

Руководитель отдела, в котором проходила практика – преподаватель Ефремова Е.Ю.

2 СВЕДЕНИЯ ОБ ИМЕЮЩЕЙСЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКЕ И ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

Практика проходила в лаборатории автоматизированных информационных систем и технологий разработки баз данных, в кабинете 32.

Кабинет 32 имеет следующие характеристики:

* площадь (не менее 6 кв.м. на одно место) – 86,3 кв.м;
* электрощит – 1 шт.;
* электророзетки – 42 шт.;
* рабочие столы:
  + ученические – 13 шт.;
  + компьютерные – 13 шт.;
  + учительский стол-кафедра – 1 шт.;
  + расстановка рабочих мест с ПЭВМ – периметральная.

В кабинете установлены 13 компьютеров со следующими характеристиками:

* материнская плата: ASUS;
* процессор: 3.21 ГГЦ;
* видео карта: AMD Athlon X2 260;
* оперативная память (RAM): 4 ГБ;
* жесткий диск (HDD): 80 ГБ;
* тип привода: DVD+/-RW.

Компьютеры объединены в локальную сеть и имеют выход в Internet.

Программное обеспечение рабочего места:

* Microsoft Visio Professional 2019;
* Microsoft Office 2019;
* Git 2.32;
* .NET Framework Developer Pack 4.8;
* Microsoft Visual Studio 2019 Community, включая следующие компоненты:
  + .NET desktop development Workload;
  + Universal Windows Platform development Workload;
  + Data storage and processing Workload;
  + Entity Framework (EF); IIS, ASP.NET and web development, Mobile development with .NET;
* Android Studio 4.2.1, включая следующие компоненты:
  + Android SDK Tools;
  + Android SDK Platform-Tools;
  + Android SDK Build-Tools;
  + Android SDK Platform;
  + USB Driver;
* Android SDK 30.29;
* SQL Server Management Studio 18.9.1;
* Visual Studio Code 1.57;
* Notepad++ 8.1;
* Postman 8.6.2.

Программное обеспечение сервера:

* Microsoft Visio Professional 2019;
* Microsoft Office 2019;
* Git 2.32;
* .NET Framework Developer Pack 4.8;
* Microsoft Visual Studio 2019 Community, включая следующие компоненты:
  + .NET desktop development Workload;
  + Universal Windows Platform development Workload;
  + Data storage and processing Workload;
  + Entity Framework (EF); IIS, ASP.NET and web development, Mobile development with .NET;
* Android Studio 4.2.1, включая следующие компоненты:
  + Android SDK Tools;
  + Android SDK Platform-Tools;
  + Android SDK Build-Tools;
  + Android SDK Platform;
  + USB Driver;
* Android SDK 30.29;
* SQL Server Management Studio 18.9.1;
* Visual Studio Code 1.57;
* Notepad++ 8.1;
* Postman 8.6.2;
* Сервер Gogs;
* IIS Express 10.0, Nginx 1.21, Apache 2.4.48;
* Microsoft SQL Server 2017 Express;

В 32 кабинете проводятся практические занятия по дисциплинам «Основы алгоритмизации и программирования», «Технология разработки и защиты баз данных», «Разработка программных модулей», «Поддержка и тестирование программных модулей», «Разработка мобильных приложений», «Системное программирование».

3 СВЕДЕНИЯ О ПРОХОЖДЕНИИ ПРАКТИКИ В КАЧЕСТВЕ ДУБЛЕРА ИНЖЕНЕРНО-ТЕХНИЧЕСКОГО РАБОТНИКА

Во время прохождения практики в качестве дублера техника-программиста я выполнял следующие действия:

* ознакомление с системным и прикладным программным обеспечением на предприятии;
* работа с документацией;
* сбор и структурирование материала по теме индивидуального задания «Реализовать приложения для грибника»;
* анализ алгоритмов, в том числе с применением инструментальных средств;
* разработка кода программного продукта на основе готовой спецификации на уровне модуля;
* написание программного кода, реализующего функциональность разрабатываемого приложения;
* разработка интерфейса приложения;
* отладка программного продукта с использованием спе­циализированных программных средств;
* разработка тестовых наборов и тестовых сценариев;
* осуществление рефакторинга и оптимизации программного кода;
* тестирование программного продукта;
* разработка технической документации к программному продукту;
* обслуживание оргтехники;
* установка и настройка необходимого программного обеспечения;
* разработка кода по заданию руководителя практики.

Выполненная работа соответствует профессиональным компетенциям, предусмотренным программой производственной практики ПМ.01 специальности 09.02.07 Информационные системы и программирование.

4 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

4.1 Постановка задачи

В период практики необходимо написать desktop-приложение реализующее функцию построителя графика y=cosec(A\*x) с оцифровкой по осям, возможностью масштабирования и задания диапазона аргумента и параметра функции, а также версию данного приложения для мобильных платформ. Программа должна содержать текстовую и графическую информацию, достаточную для исследования особенностей данной функции, иметь удобный и понятный пользователю графический интерфейс, быть проста в использовании.

Программа должна предлагать следующие возможности:

* ввода и вывода диапазона построения функции;
* ввода значения параметра функции;
* строить график функции;
* масштабировать график функции;
* выдавать справочную информацию;
* сохранять результаты исследования в файлах.

Входными данными будут:

* результаты исследования функции y = cosec(А\*x);
* интервал, на котором необходимо построить график функции;
* значение параметра;
* выбранный масштаб.

Выходными данными будут:

* визуальное отображение графика функции;
* информация, полученная при исследовании функции;
* пользовательская справочная система.

Также необходимо разработать пакет unit-тестов для функции, вычисляющей значения исследуемой функции.

4.2 Проектирование структуры программы

4.2.1 Desktop версия приложения

Разработанное приложение имеет следующие окна:

* заставка;
* главное окно, которое содержит области:
* ввода значений параметра и диапазона построения графика функции;
* вывода информации о функции;
* вывода графика функции;
* справка;
* сведения о разработчике.

Обращение к окнам реализовано с помощью меню (за исключением окна заставки).

4.2.2 Мобильная версия приложения

Разработанное приложение имеет следующие окна:

* окно заставки;
* главное окно:
* окно информации о графике;
* окно инструкции по использованию приложения;
* окно «О программе».
* окно ввода данных.

Обращение к окнам реализовано с помощью меню (за исключением окна заставки).

4.3 Функциональное взаимодействие модулей программы

4.3.1 Desktop версия приложения

На основе разработанного технического проекта был написан программный продукт «Справочник грибника». Для создания приложения была использована среда программирования Visual Studio 2019 и язык программирования С#. В программе используются следующие языковые модули:

* System.Collections.ObjectModel;− Содержит классы, которые могут использоваться в качестве коллекций в объектной модели библиотеки многократного использования. Эти классы предназначены для использования со свойствами или методами, возвращающими коллекции.
* System− содержит фундаментальные и базовые классы, определяющие часто используемые типы значений и ссылочных данных, события и обработчики событий, интерфейсы, атрибуты и исключения обработки;
* System.Windows− содержит типы, используемые в приложениях WindowsPresentationFoundation (WPF), включая клиенты анимации, элементы управления пользовательского интерфейса, привязку данных и преобразование типов;
* System.IO −cодержит типы, поддерживающие ввод и вывод, включая возможности чтения и записи данных в потоках как синхронно, так и асинхронно, реализации пользовательского ведения журнала и обработки входящих и исходящих потоков данных в последовательных портах;
* System.Windows.Media.Imaging− Предоставляет типы, используемые для кодирования и декодирования растровых изображений.
* System.Windows.Controls Предоставляет классы для создания элементов, называемых элементами управления, которые позволяют пользователю взаимодействовать с приложением. Классы элементов управления являются основой пользовательского взаимодействия с приложением, так как они позволяют пользователю просматривать, выбирать или вводить данные или другую информацию.
* System.Collections.Generic – работа с обьектами коллекци (List<string>) [5].

В программе были созданы следующие пользовательские модули:

* GlavMenu.xaml.cs (главное окно программы) − обеспечивает функциональность самого приложения и переходов между окнами
* FileWorker.cs – система помощи, которая находит файлы в системе и считывает информацию
* o\_prog.xaml.cs (окно о программе) – этот модуль содержит сведения о разработчике программы.
* PageViewModel.cs – этот модуль обеспечивает нахождения и вывода информации в меню окон
* User.cs – это модуль, который задаёт параметры списка

В модуле GlavMenu.xaml.cs описаны и используются следующие процедуры:

* private void Vkus\_Click(object sender, RoutedEventArgs e) − открывает окно с информации
* private void inedible\_Click(object sender, RoutedEventArgs e)− открывает окно с информации
* private void conditionally\_Click(object sender, RoutedEventArgs e)− открывает окно с информации
* private void ogorod\_Click(object sender, RoutedEventArgs e)− открывает окно с информации
* private void edible\_Click(object sender, RoutedEventArgs e)-открывает окно с информации
* private void conditionally\_Click(object sender, RoutedEventArgs e)− открывает окно с информации
* private void ydov\_Click(object sender, RoutedEventArgs e)-открывает окно с информации
* private void vs\_Click(object sender, RoutedEventArgs e)− открывает окно с информации
* private void oprog\_Click(object sender, RoutedEventArgs e) − открывает справку;

Код данных процедур представлен в приложении А.

4.3.2 Мобильная версия приложения

На основе разработанного технического проекта была написана мобильная версия программного продукта «Исследование функции y=A/x». Для создания приложения была использована среда программирования Android Studio и язык программирования Java. В программе используются следующие языковые модули:

* android.content.Context - это базовый абстрактный класс, реализация которого обеспечивается системой Android. Этот класс имеет методы для доступа к специфичным для конкретного приложения ресурсам и классам и служит для выполнения операций на уровне приложения, таких, как запуск активностей, отправка широковещательных сообщений, получение намерений и прочее;
* android.graphics.Canvas - предоставляет методы для рисования, которые отображают графические примитивы на исходном растровом изображении;
* android.graphics.Color - содержит несколько констант и методов для конвертации и извлечения цветового компонента;
* android.graphics.Paint - содержит стили, цвета и другую графическую информацию для рисования графических объектов;
* android.view.View - компонент(виджет), который рисуется на экране;
* java.util.ArrayList - класс, позволяющий создавать списки;
* android.app.AlertDialog - это уведомление, которое появляется, когда определенное приложение использует систему Android для размещения важной информации для пользователя;
* android.content.Intent - класс, объекты которого используются для взаимодейтвия между отдельными частями Android приложения, либо между различными приложениями системы;
* android.os.Bundle - необходим для временного хранения данных в процессе выполнения;
* android.view.Menu – класс, представляющий меню в приложениях;
* android.view.MenuItem – класс, отвечающий за пункты меню;
* android.widget.Toast - это небольшое, всплывающее сообщение, появляющееся внизу и экрана на непродолжительное время;
* androidx.appcompat.app.AppCompatActivity – это базовый класс для действий, которые используют функциональные возможности панели поддержки библиотеки;
* android.widget.EditText - это стандартный виджет ввода текста в приложениях Android;
* java.io.FileOutputStream - поток вывода, который содержит методы, записывающие данные в файл;
* android.os.AsyncTask - предлагает простой и удобный механизм для перемещения трудоёмких операций в фоновый поток.

В программе были созданы следующие пользовательские модули:

* Activity\_Preview.java – окно заставки;
* Activity\_Chart.java – основное окно, в котором находится меню приложения;
* Activity\_ChartSettings.java – окно для ввода данных;
* MyDraw.java – отрисовка графика и холста;
* ChartValues.java – класс для хранения данных графика.

В модуле Activity\_Preview.java описаны и используются следующие процедуры:

* class Timer extends AsyncTask<Void, Integer, Void> - поток для таймера;
* public void Next\_Activity() – метод для перехода на другую активность.

В модуле Activity\_Chart.java описаны и используются следующие процедуры:

* public void saveText() – сохраняет данные в файле;
* public String openText(String Name) – считывает данные из файла;
* public boolean onCreateOptionsMenu(Menu menu) – создаёт меню;
* public boolean onOptionsItemSelected(MenuItem item) – обрабатывает нажатия на пункты меню;
* protected void onActivityResult(int requestCode, int resultCode, Intent data) – обрабатывает данные, возвращаемые из другой активности.

В модуле Activity\_ChartSettings.java описаны и используются следующие процедуры:

* public void onClickExit(View v) – обработка нажатия кнопки выхода без сохранения;
* public void onClickAcceptExit(View v) - обработка нажатия кнопки выхода с сохранением;
* public void saveText() – метод сохранения данных.

В модуле MyDraw.java описаны и используются следующие процедуры:

* public MyDraw(Context context) – конструктор класса MyDraw;
* protected void onDraw(Canvas canvas) – метод, который вызывает методы для отрисовки холста и графика;
* private void drawMetric(Canvas canvas) – метод для отрисовки системы координат и холста;
* private void drawChart(Canvas canvas) – метод для отрисовки графика.

Код класса MyDraw.java представлен в приложении Б.

4.4 Разработка unit-тестов

Для осуществления корректной работы разработанного приложения, в части правильности вычисления значений функции, был разработан тестовый проект, включающий в себя набор unit-тестов. Данные тесты охватывают все возможные тестовые случаи, которые необходимо проверить:

* вычисление значения функции в граничных точках;
* вычисление значения функции в точках разрыва;
* вычисление значения функции в точках минимумов и максимумов;
* …

Код данных тестов представлен в приложении В.

4.5Работа с приложением

4.5.1 Desktop версия приложения

При запуске программы открывается заставка (рисунок 1).

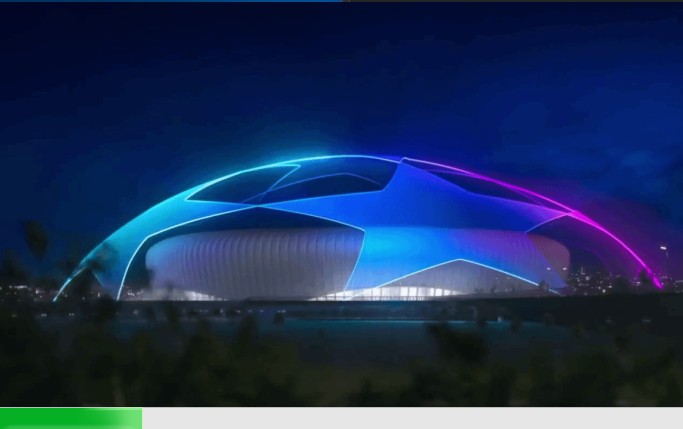


Рисунок 1 – Заставка

После завершения загрузки, откроется Главное окно программы.

Интерфейс программы имеет следующие элементы (рисунок 2):

* область вывода графика функции (1);
* кнопка вызова справки (2);
* кнопка, закрывающая приложение (3);
* поле для ввода начального значения диапазона построения (4);
* поле для ввода конечного значения диапазона построения (5);
* кнопка, по нажатию на которую строится график (6);
* поле текущего масштаба функции (7);
* кнопки, позволяющие изменить значение масштаба функции (8);
* информация о графике (9).

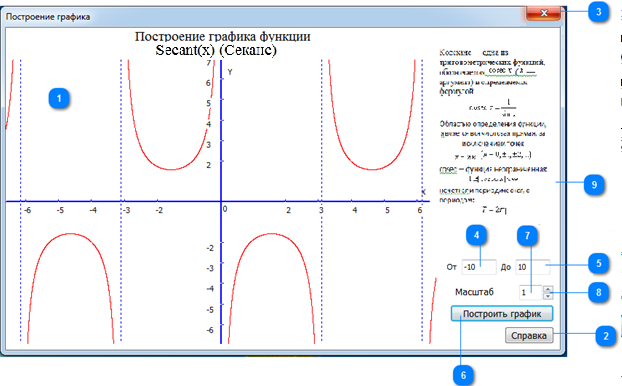


Рисунок 2 – Основное окно программы

Для исследования функции нужно указать левую и правую границы построения графика и масштаб. Далее надо нажать кнопку «Построить график». После этого на экране появятся уравнение функции и график. При необходимости можно изменить параметры.

Если у пользователя возникнуть вопросы по работе с программой, он может вызвать справочное окно, нажав на кнопку «Справка» или выбрав пункт меню «Справка» в Главном окне (рисунок 3).

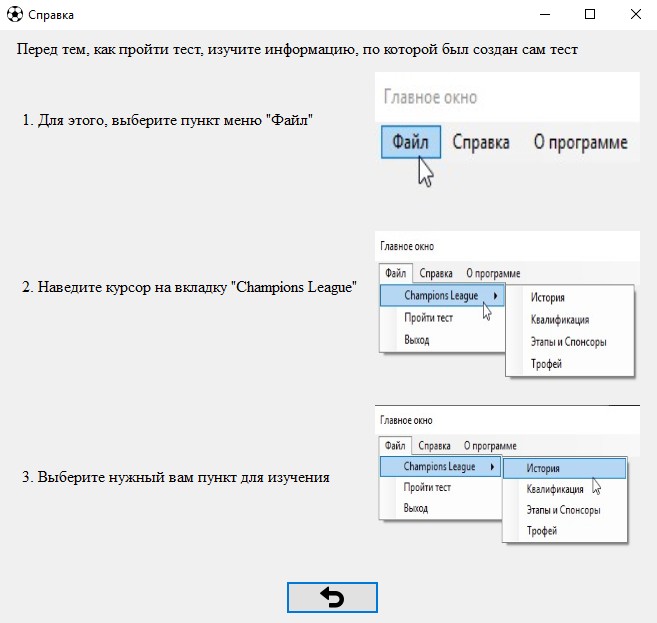


Рисунок 3 – Окно справки

При нажатии пункта меню «О программе» в главном окне, откроется окно просмотра информации о создателе программы (рисунок 4).

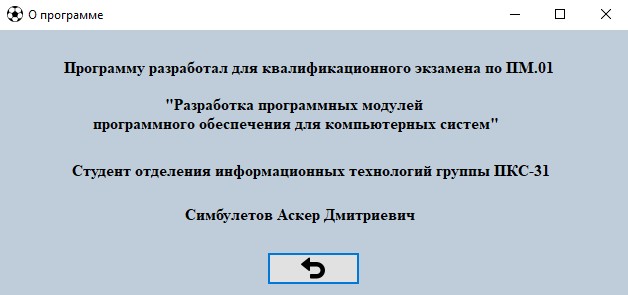


Рисунок 4 – Окно о программе

4.5.2 Мобильная версия приложения

При запуске программы открывается заставка (рисунок 5).

После завершения загрузки, откроется главное окно программы.

Меню приложения появится после нажатия на кнопку 1 (рисунок 6):

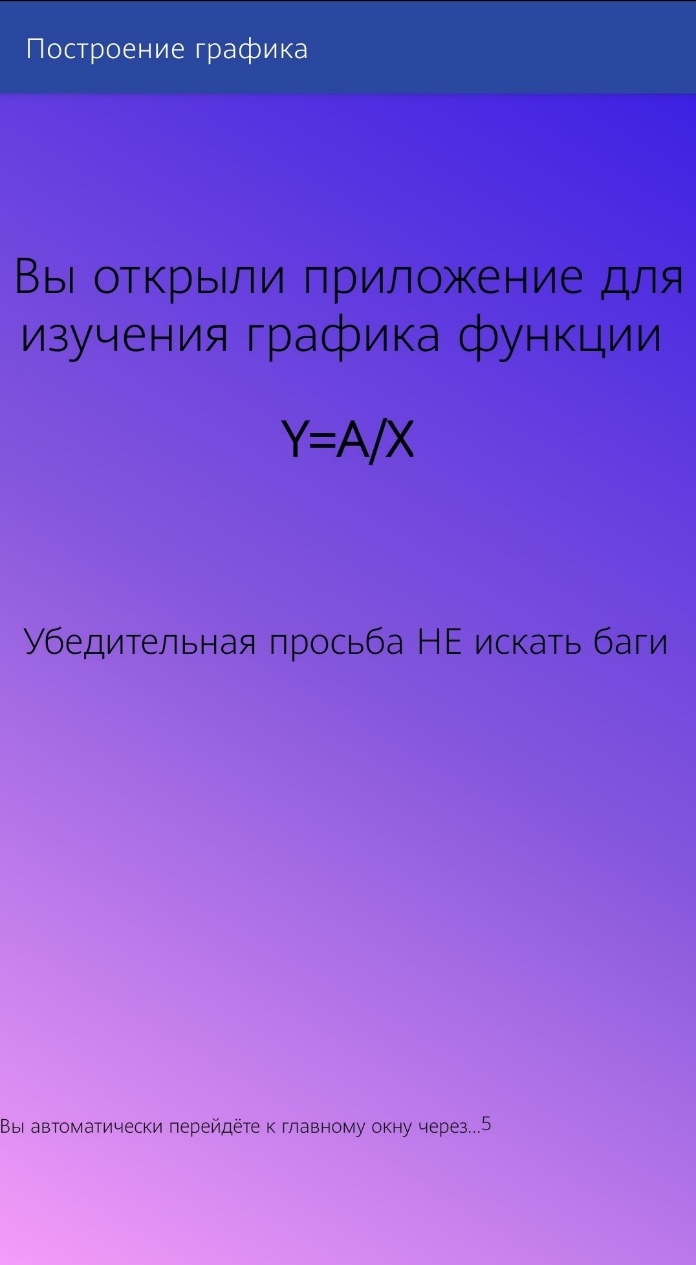


Рисунок 5 – Заставка

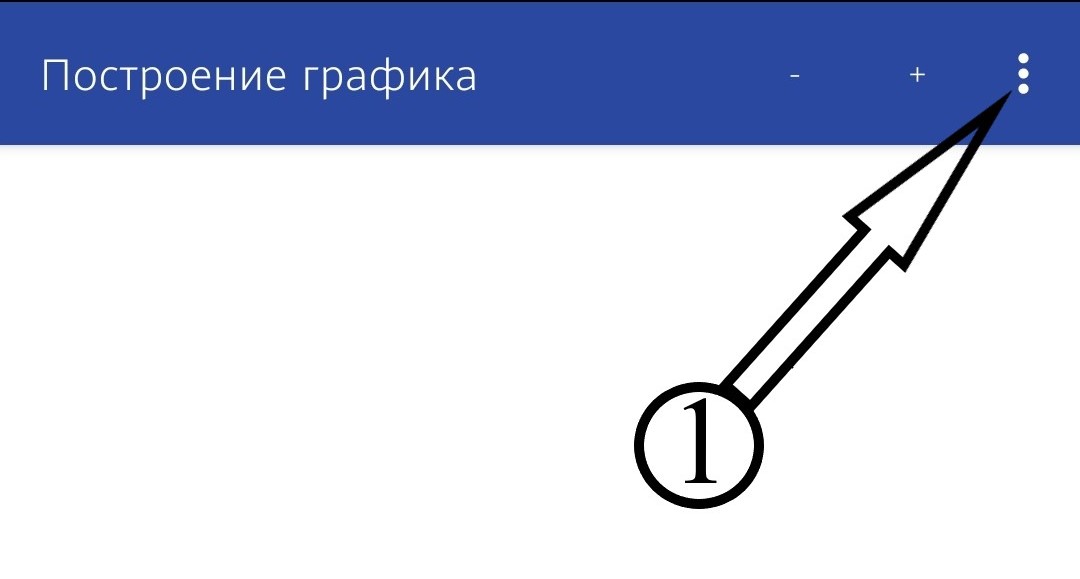


Рисунок 6 – Основное окно программы

Далее появится список с функциями приложения (рисунок 7).

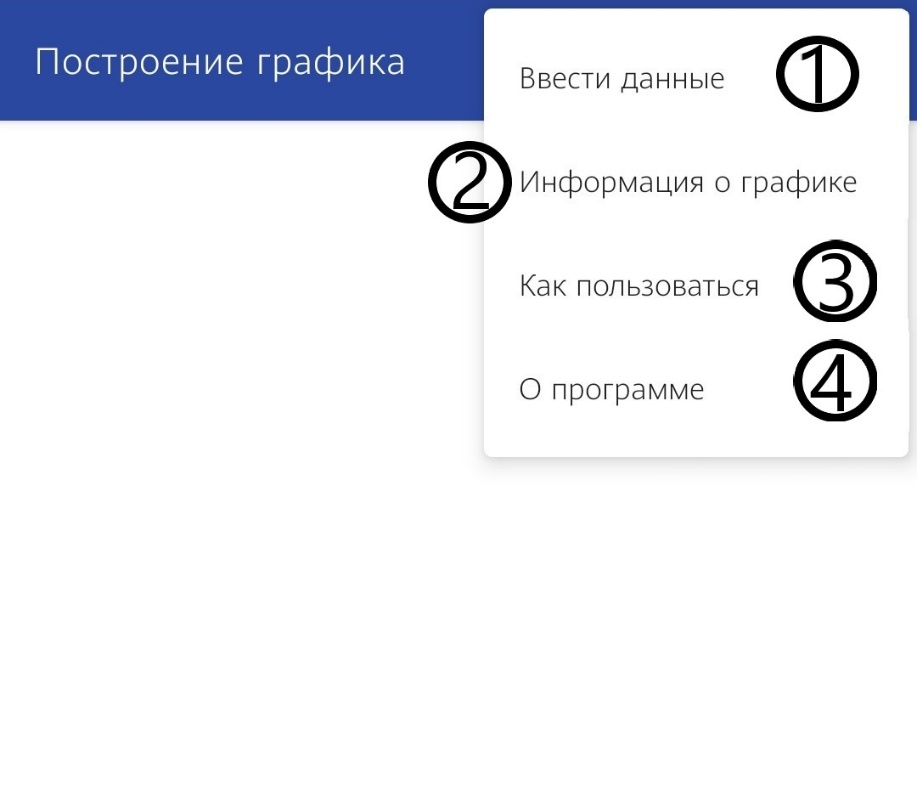


Рисунок 7 – Меню приложения

Для исследования функции необходимо нажать «Ввести данные» (кнопка 1, рисунок 7), указать левую и правую границы построения графика и параметр функции. Далее надо нажать кнопку «Принять». Если вы решили не принимать значения, то нажмите кнопку «Назад». После этого на экране появится график. Для отображения информации о графике нажмите кнопку «Информация о графике» (кнопка 2, рисунок 7), затем появится всплывающее окно (рисунок 8).

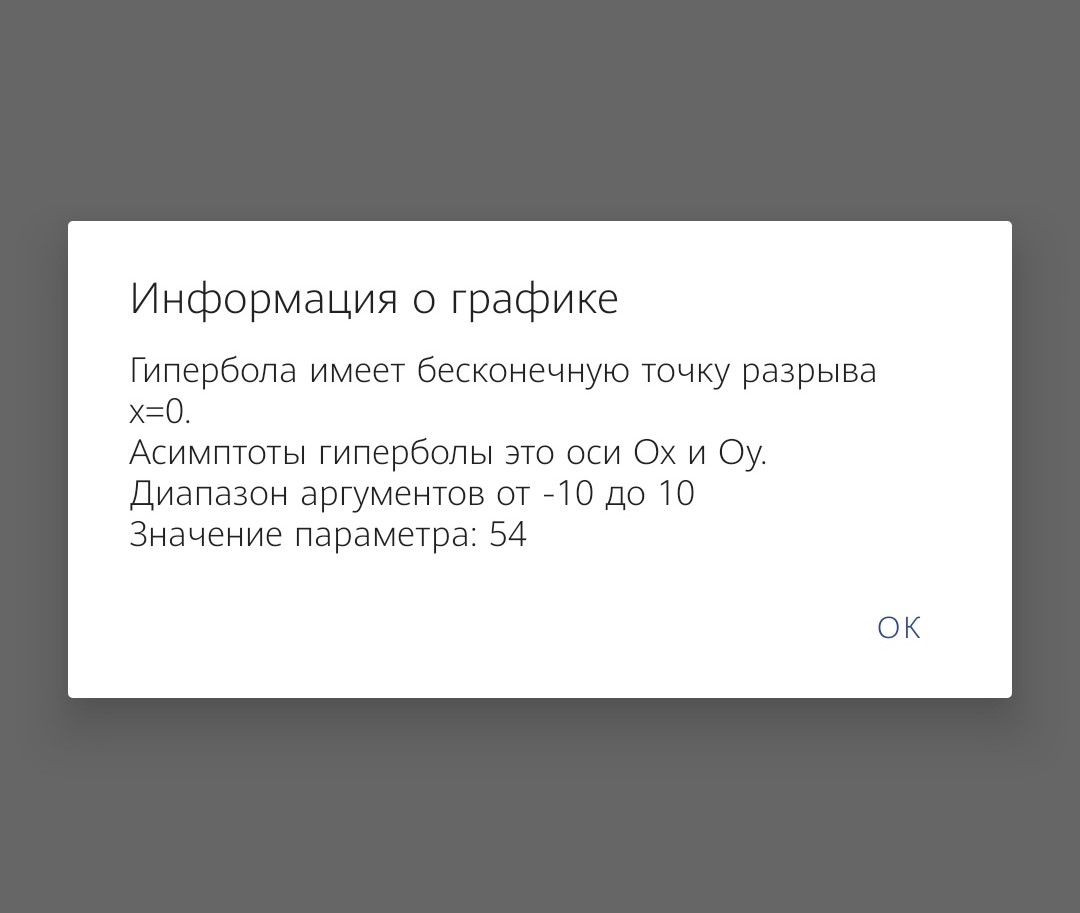


Рисунок 8 – Окно информации о графике

Если у пользователя возникнуть вопросы по работе с программой, он может вызвать справочное окно, выбрав пункт меню «Как пользоваться» (кнопка 3, рисунок 7) в главном окне (рисунок 9).

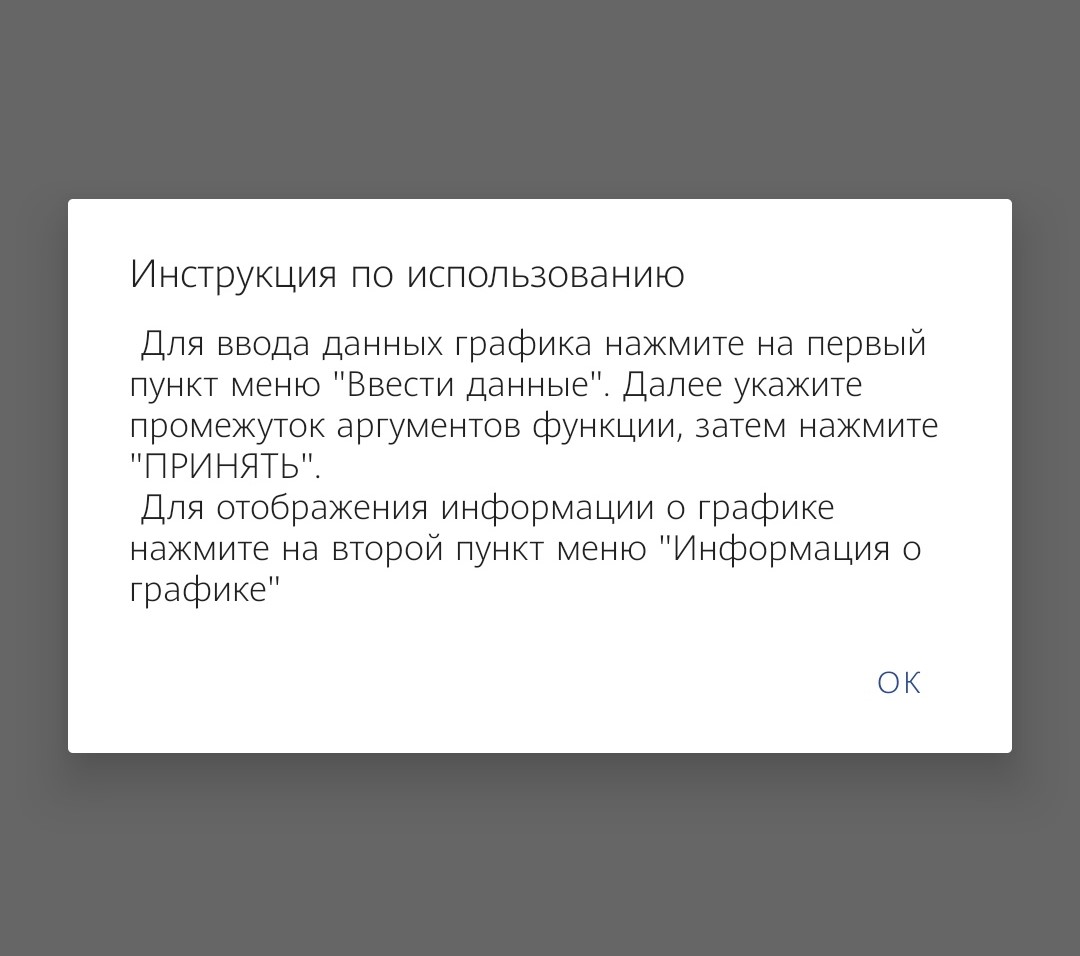


Рисунок 9 – Окно инструкции по использованию

При нажатии пункта меню «О программе» (кнопка 4, рисунок 7) в главном окне, откроется окно просмотра информации о создателе программы (рисунок 10).

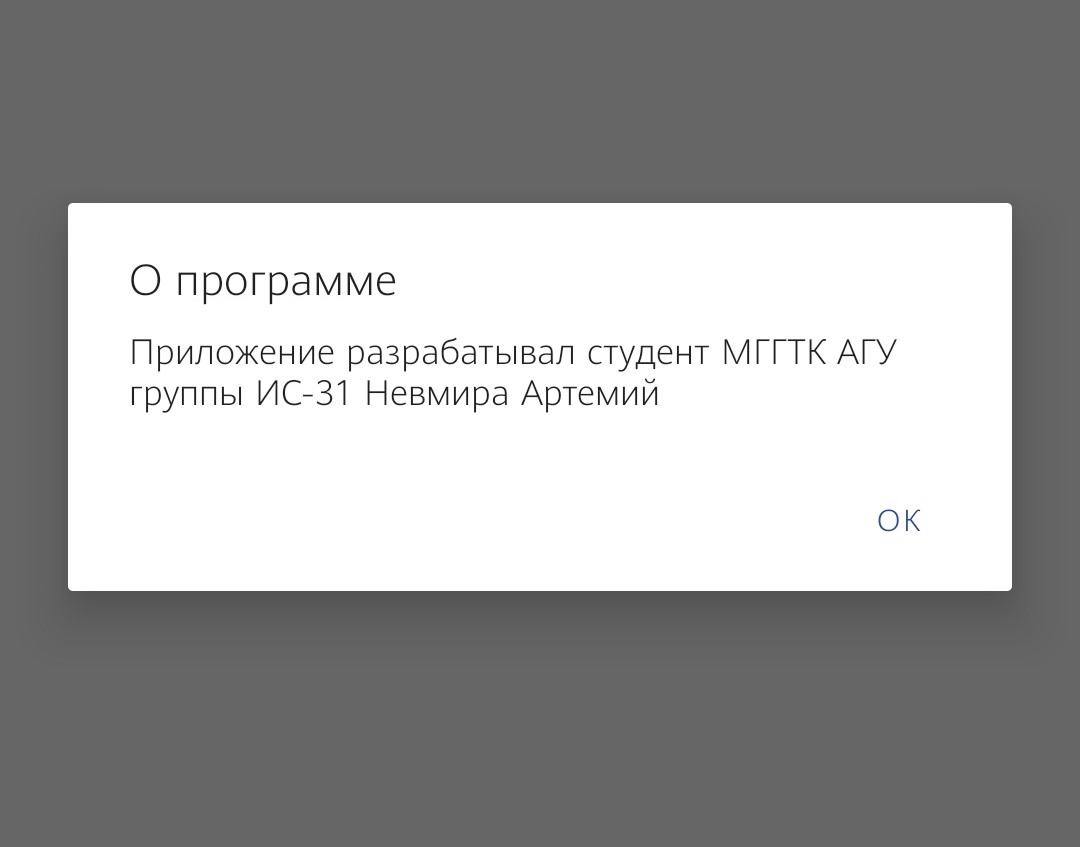


Рисунок 10 – Окно о программе

ЗАКЛЮЧЕНИЕ

В результате прохождения производственной практики мною были закреплены теоретические знания, полученные при изучении ПМ.01; отработаны профессиональные компетенции по получаемой специальности; приобретены практические навыки в будущей профессиональной деятельности; выполнено индивидуальное задание.

Мною было разработано приложение для исследования функции   
y = cosec(A\*x).

Программа имеет удобный и понятный пользователю графический интерфейс, проста в использовании и обеспечивает следующие возможности:

* ввод и вывод диапазона построения функции;
* ввод значения параметра функции;
* строит график функции;
* масштабирует график функции;
* выдает справочную информацию;
* сохранять результаты исследования в файлах.

В ходе выполнения разработки данного приложения были:

* проанализирована методика исследования функции;
* спроектирована структура приложения и программного кода;
* разработаны справочные материалы;
* выбраны средства реализации приложения;
* разработан программный код приложения;
* разработан тестовый проект, реализующий оптимальный набор unit-тестов.

Основными достоинствами программы являются:

* программа проста в установке, настройке и эксплуатации;
* данный программный продукт предназначен для широкого круга пользователей;
* наличие справочной системы.

К недостаткам данной программы можно отнести то, что она предназначена для исследования только одной функции.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Албахари Б., Албахари Дж. С# 7.0. Справочник. Полное описание языка. М.: Вильямс, 2018. 1024 с.
2. Виссер Д. Разработка обслуживаемых программ на языке С#. М.: ДМК Пресс, 2017. 194с.
3. Дарвин Я. Ф. Android. Сборник рецептов: задачи и решения для разработчиков приложений. 2-е изд. Пер. с англ. СПб.: ООО «Альфа-книга», 2018. 768 с.
4. Джепикс Ф., Троелсен Э. Язык программирования C# 7 и платформы .NET и .NET Core. М.: Вильямс, 2018. 1328 с.
5. Мюллер Д. П., Семпф Б., Сфер Ч. С# Для чайников. М.: Диалектика, 2016. 608 с.
6. Скит Д. С# Для профессионалов. Тонкости программирования. М.: Вильямс, 2019. 608 с.
7. Документация Microsoft. URL: https://docs.microsoft.com/ru-ru/documentation (дата обращения: 01.12.2021).
8. Библиотека классов .NET Framework. URL: https:// msdn.microsoft.com/ru-ru/library/mt472912(v=vs.110).aspx (дата обращения: 01.12.2021).
9. Как пользоваться timer в c#. URL: https://docs.microsoft.com/ru-ru/dotnet/api/system.timers.timer?view=netcore-2.1 (дата обращения: 05.12.2021).
10. Космос Планет. Сайт всё о космосе. URL: https://cosmosplanet.ru/ solnechnayasistema/stroenie-solnechnoj-sistemy.html (дата обращения: 02.12.2021).
11. Общие сведения о платформе Android. URL: https://developer.android.com/guide (дата обращения: 30.11.2021).

ПРИЛОЖЕНИЕ А

Основные процедуры desktop приложения