

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка ПО для определения внешнебаллистических параметров артиллерийских систем.»

Студент

Е.Н. Козлов
(И.О. Фамилия)

(личная подпись)

Руководитель

Доктор физ-мат. наук, Профессор А.И. Сафронов
(ученая степень, звание, И.О. Фамилия)

Тольятти 2022

Аннотация

Выпускная квалификационная работа на тему: «Разработка ПО для определения внешнебаллистических параметров артиллерийских систем».

Работа выполнена студентом Тольяттинского государственного университета, института математики, физики и информационных технологий, группы Моб-1802а, Козловым Егором Николаевичем.

Объём работы - 53 страницы, на которых представлены 13 рисунков и 1 приложение. При написании работы использовались 21 источник.

Объект исследования: вычислительные системы.

Объектом исследования данной работы являются вычислительные приложения.

Предмет исследования – вычислительное приложение на Python.

Цель работы – разработать вычислительное приложение на Python.

Практическая значимость работы заключается в том, что разработанное приложение можно использовать в военном учебном центре при университете.

Отчет состоит из введения, трех глав и заключения.

В первой главе представлен анализ предметной области.

Во второй главе описана разработка и реализация приложения.

В третьей главе проводится тестирование приложения.

Abstract

Final qualifying work on the topic: "Software development for determining the external ballistic parameters of artillery systems."

The work was done by a student of the Togliatti State University, Institute of Mathematics, Physics and Information Technologies, Mob-1802a group, Egor Nikolaevich Kozlov.

Scope of work - 56 pages, which contain 13 drawings and 1 application. When writing the work, 14 sources were used.

Object of research: computer systems.

The object of study of this work are computational applications.

The subject of the research is a computational application in Python.

The purpose of the work is to develop a computational application in Python.

The practical significance of the work lies in the fact that the developed application can be used in a military training center at the university.

The report consists of an introduction, three chapters and a conclusion.

The first chapter presents an analysis of the subject area.

The second chapter describes the development and implementation of the application.

The third chapter is testing the application.

Оглавление

Введение.....	6
Глава 1 Анализ предметной области.....	8
1.1 Характеристика предметной области	8
1.1.1 Характеристика учебного заведения.....	8
1.1.2 Краткая характеристика ВУЦ и его видов деятельности.....	8
1.1.3 Сущность задачи автоматизации.....	8
1.2 Постановка задачи	9
1.2.1 Цель и назначение автоматизированного варианта решения задачи	9
1.2.2 Этапы решения задачи.....	9
1.3 Анализ существующих разработок и обоснование выбора технологии проектирования	10
Глава 2 Разработка и реализация проектных решений	15
2.1 Выбор среды разработки.....	15
2.1.1 Java.....	15
2.1.2 C++.....	18
2.1.3 C#	20
2.1.4 Python.....	22
2.2. Физическое моделирование АИС	24
2.2.1 Выбор архитектуры АИС	24
2.2.2 Функциональность проекта.....	24
2.2.3 Структура проекта.....	24
2.2.4 Описание используемых библиотек.....	25

2.3 Технологическое обеспечение задачи	33
2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации	33
2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации	33
Глава 3 Тестирование приложения	34
Заключение	41
Список используемой литературы и используемых источников.....	42
Приложение А Листинг программы.....	44

Введение

На данный момент любая отрасль человеческой деятельности так или иначе связана с применением электронно-вычислительных устройств, персональных компьютеров, мобильных устройств. Преимущество использования компьютера для обработки информации заключается в его безупречном выполнении инструкций, т.е. программного кода. Вследствие чего компьютер не ошибается (при условии, что в него введены правильные данные). Кроме того, компьютерная обработка экономит массу времени для человека.

В мире существует множество разнообразных аппаратных и программных платформ. Программное обеспечение сильно привязано к операционным системам (ОС). Это приводит к их несовместимости. Например, программа, разработанная под Windows, не запускается под ОС Linux или под ОС iOS. Программное обеспечение, как правило, разрабатывается под определенную операционную систему. И это негативно сказывается на совместимости программ, т.к. приложение, разработанное, например, под Windows отказывается запускаться под операционной системой IOS без дополнительных манипуляций (установка эмулятора Windows). Это вынуждает создавать мультиплатформенные приложения с использованием платформ типа .NET или Java.

Объектом исследования данной работы являются вычислительные приложения.

Предмет исследования – Военный учебный центр при ТГУ.

Цель работы – разработать вычислительное приложение на Python.

Для достижения поставленной цели необходимо решить ряд задач:

- проанализировать литературу по заданной теме;
- изучить предметную область;
- обосновать необходимость разработки информационной системы;

- описать техническое, программное и технологическое обеспечение;
- провести тестирование разработанного приложения.

В работе представлено три главы:

- аналитическая, в которой содержится анализ предметной области;
- проектная, в которой описаны процессы, представлена разработка ПО и приведен контрольный пример;
- заключительная, которая является тестированием разработанного ПО.

Глава 1 Анализ предметной области

1.1 Характеристика предметной области

1.1.1 Характеристика учебного заведения

ВУЦ (Военный учебный центр) – это учебное подразделение при вузе, где студенты проходят программу военной подготовки офицеров, сержантов запаса. В ТГУ подготовка будущих офицеров осуществляется по военно-учетным специальностям артиллерии.

1.1.2 Краткая характеристика ВУЦ и его видов деятельности

«Основными задачами военного учебного центра являются:

- обучение граждан по программе военной подготовки для прохождения военной службы по контракту на воинских должностях, подлежащих замещению офицерами (программа подготовки офицеров кадра);
- обучение граждан по программе военной подготовки офицеров запаса и по программе военной подготовки сержантов запаса (программы подготовки запаса);
- проведение воспитательной (военно-политической) работы;
- проведение работы по военно-профессиональной ориентации молодежи.»[15]

1.1.3 Сущность задачи автоматизации

Вычисления – сложный процесс, который имеет рутинный характер. Именно из-за рутинности становятся наиболее вероятны человеческие ошибки вследствие снижения внимания.[1]

Применение приложения для компьютера позволяет свести риск человеческих ошибок к минимуму. Это обуславливается тем, что программное обеспечение контролирует вводимые данные на корректность: например, нельзя ввести текстовые символы в поле для ввода, которое предназначено для ввода

только чисел. Контролируется пустой ввод – т.е. если входные данные должны быть введены, но не были введены, то приложение выдаст соответствующее предупреждение и прервет выполнение программы до нужной корректировки данных со стороны пользователя программы.

Выводы по параграфу (актуальность задачи)

Актуальность данной работы заключается в отсутствии программного обеспечения для заданных вычислительных процессов.

1.2 Постановка задачи

1.2.1 Цель и назначение автоматизированного варианта решения задачи

Главные цели разработанного приложения – это автоматизация процессов вычислений и повышения общего уровня автоматизации ВУЦ.

1.2.2 Этапы решения задачи

Процесс разработки вычислительного приложения включает в себя следующие этапы:

- изучение предметной области – вычислительный процесс на Python.
- анализ аналогов программных продуктов, позволяющих решить задачу;
- анализ существующих средств разработки для реализации проекта;
- описание входных и выходных данных;
- разработка ПО;
- тестирование ПО;

Выводы по параграфу (слабые места, предложения по их устранению)

В данном параграфе приводится постановка задачи, которая должна быть выполнена в рамках работы над ВКР.

1.3 Анализ существующих разработок и обоснование выбора технологии проектирования

На данный момент для автоматизации вычислений часто используют следующие программные продукты:

- NumLockCalculator,
- Microsoft Excel,
- MathCAD.

NumLockCalculator – простое приложение, которое представляет собой удобный инженерный калькулятор (рисунок 1).

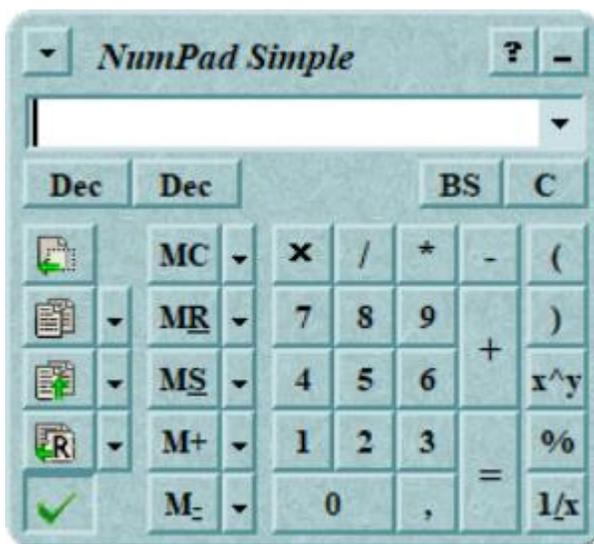


Рисунок 1 – Внешний вид окна программы NumLockCalculator

Данное приложение имеет некоторые функции, которые недоступны обычным калькуляторам. В частности, поддерживаются разные системы счисления (двоичная, восьмеричная, шестнадцатеричная), имеется поддержка скобок приоритета, встроена работа с буфером обмена и сохранением истории операций вычисления, имеется богатый функционал в части дополнительных настроек. Однако полноценным вычислительным средством это приложение

назвать нельзя, т.к. это всего лишь достаточно примитивный инструмент, хоть и обладает удобными функциями.[3]

Microsoft Excel – табличный редактор (на рисунке 2 приведен скриншот окна данного приложения), который позволяет производить вычисления с помощью формул.

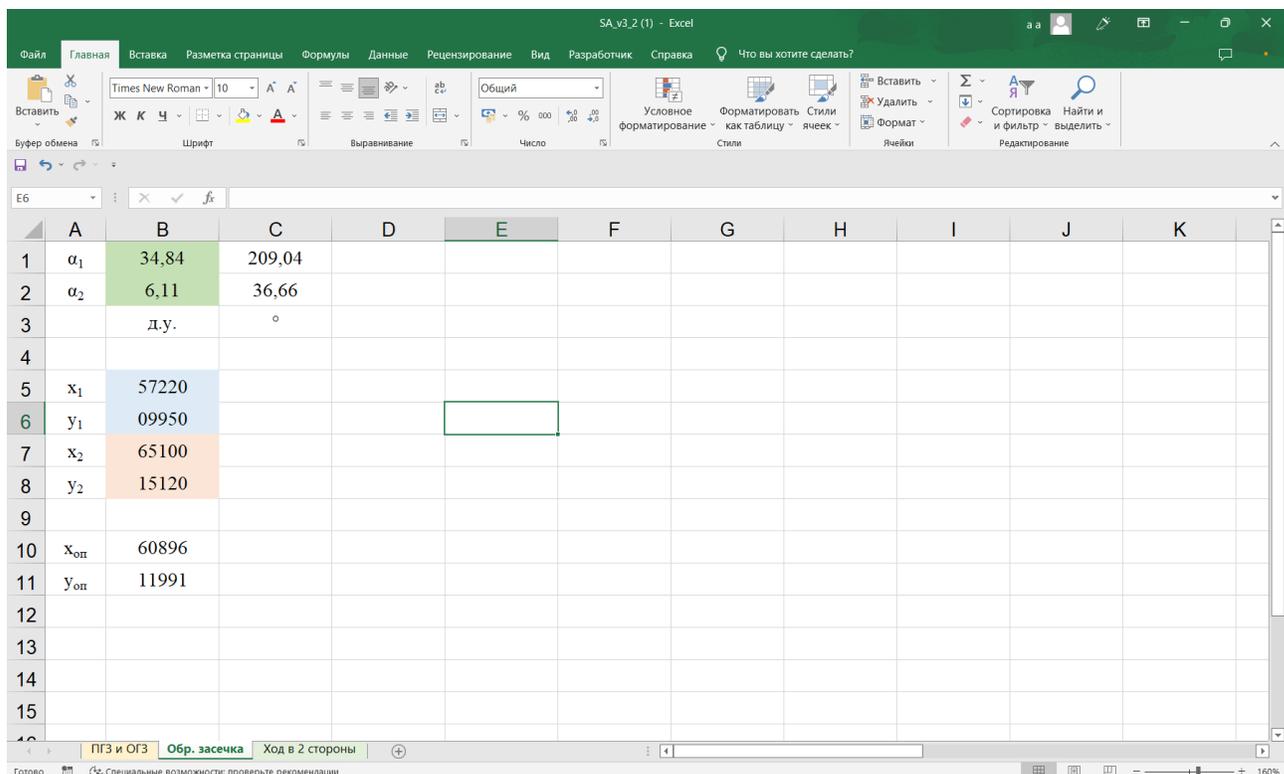


Рисунок 2 – Внешний вид окна программы Microsoft Excel

Его характерной особенностью является простота и наглядность вычислений. Пользователь легко контролирует процесс вычислений, может обрабатывать большие массивы однотипных данных, поддерживается копирование (или размножение) формул. Данный инструмент достаточен для большинства инженерных и бухгалтерских задач, позволяет создавать небольшие приложения, если использовать макросы и встроенный язык программирования –

VBA (visual basic application). Минусом приложения является платность и привязка к Microsoft. Но это не единственные минусы Excel.

Основные недостатки Excel:

- изучить это программное обеспечение сложнее, чем MS Word или любое другое программное обеспечение. В Excell используются формулы, а изучение формул сложнее для новичка;

- это программное обеспечение имеет высокую цену. Есть много альтернатив, например, OpenOffice — это бесплатный инструмент, который вы можете использовать для создания файлов электронных таблиц;

- трудно обнаружить ошибки в этом программном обеспечении. Поскольку вы можете хранить миллионы записей в этом программном обеспечении, поиск ошибок в нем становится сложным;

- если вы сохраните много данных в этом Excell, использование программного обеспечения станет медленным. Программное обеспечение будет реагировать на действия пользователя медленнее, и вам потребуется больше времени для ввода или просмотра данных в нем.

Пользователи, которым необходимо выполнять сложные научные расчеты или статистический анализ, должны любой ценой избегать Excel. В отчете, опубликованном Евой Голдуотер из Массачусетского университета под названием «Using Excel for Statistical Data Analysis - Caveats», делается вывод о том, что Excel не является хорошей программой для анализа данных. В нем упоминаются многочисленные проблемы с удобством использования программного обеспечения, в том числе проблемы с его внутренними формулами.

В результате данное приложение не годится для того, чтобы производить в нём сложные расчёты.

Далее рассмотрим MathCAD.

MathCAD – полноценный продукт для математических вычислений (на рисунке 3 приведен скриншот окна данного приложения).

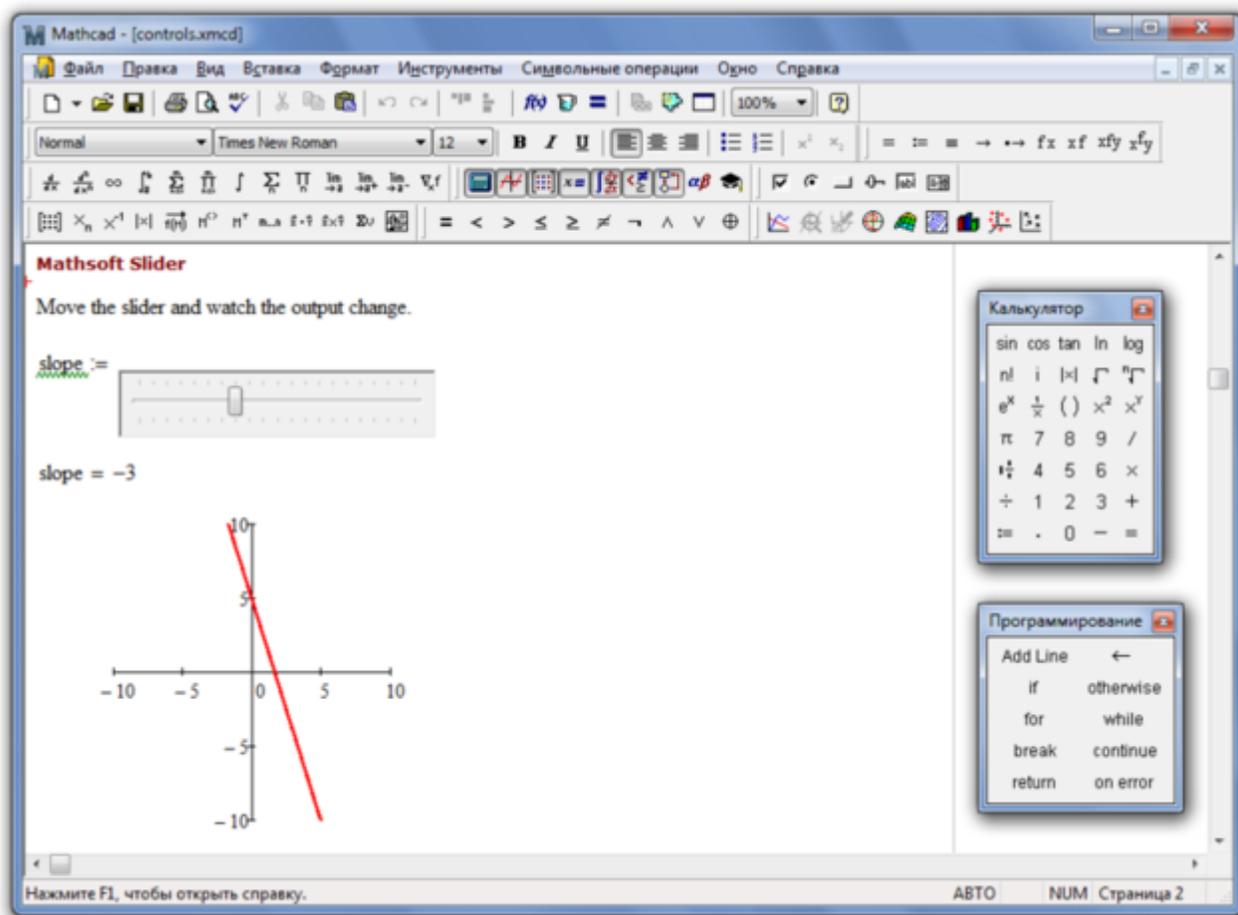


Рисунок 3 – Внешний вид окна программы MathCAD

В нем можно производить почти любые вычисления (например, действия над алгебраическими выражениями, действия над обычными дробями), многие из которых невозможны в других приложениях. Программа позволяет решать уравнения и системы уравнений. Имеется возможность построения графиков функций двух и трех переменных (графики на плоскости и в объеме). Минусом приложения MathCAD можно также назвать его высокую стоимость лицензии, а также работу только в операционной системе (ОС) Windows.

Далее сравним достоинства и недостатки данной программы.

Некоторые основные достоинства MathCAD:

- естественный математический язык;

- наглядность;
- хорошая диагностика ошибок;
- высокая точность вычислений;
- обеспечение работы с комплексными числами;
- реализация многих стандартных функций вычислительной математики;
- возможности символьных математических преобразований.

К основным недостаткам системы относятся следующие:

- ограниченные возможности существующих операторов;
- трудности реализации сложных алгоритмов.

Можно сделать вывод, что система MathCAD не предназначена для разработки сложных программ, но она очень удобна для тестирования их отдельных частей.

Поэтому необходимо разработать такое программное обеспечение, которое будет не привязано к определенной операционной системе (или будет иметь возможность перекомпилировать исходные тексты в приложение под другую операционную систему). Данное приложение не должно быть сложным, должно иметь интуитивно-понятный интерфейс.

Выводы по параграфу (решение).

В данном параграфе рассмотрены три аналога программного обеспечения, дана краткая характеристика этим приложениям.

Выводы по главе 1.

В данной главе работы приведен анализ предметной области. Описана характеристика организации.

Задача автоматизации, поставленная в рамках ВКР, заключается в разработке вычислительного приложения в python.

Глава 2 Разработка и реализация проектных решений

2.1 Выбор среды разработки

2.1.1 Java

Java – это язык программирования общего назначения, который следует парадигме объектно-ориентированного программирования и подходу «Написать один раз и использовать везде». Java используется для настольных, сетевых, мобильных и корпоративных приложений.

«Плюсы языка

Объектно-ориентированное программирование.

Java – это объектно-ориентированный язык, который помогает нам повысить гибкость и возможность повторного использования кода. Используя концепцию ООП, мы можем легко повторно использовать объект в других программах.

Это также помогает нам повысить безопасность, связывая данные и функции в единое целое и не позволяя внешнему миру получить к ним доступ. Это также помогает организовать большие модули в более мелкие, чтобы их было легко понять.

Java — язык высокого уровня с простым синтаксисом и плавной кривой обучения.

Любой язык можно считать простым, если его легко выучить и понять. Синтаксис Java прост, его легко писать, изучать, поддерживать и понимать, код легко поддается отладке.

Кроме того, Java менее сложен, чем такие языки, как C и C++, потому что многие сложные функции этих языков удаляются из Java.

Стандарт для корпоративных вычислительных систем.

Корпоративные приложения — главное преимущество Java с 90-х годов, когда организации начали искать надежные инструменты программирования не на

С. Java поддерживает множество библиотек — строительных блоков любой корпоративной системы. Библиотеки помогают разработчикам создавать любые функции, которые могут понадобиться компании. Java широко распространен — это язык, который преподают в рамках введения в программирование в большинстве школ и университетов. Возможности интеграции Java впечатляют: большинство хостинг-провайдеров поддерживают Java. Более того, Java — язык, дешевый в обслуживании: работать с Java можно с любого компьютера, вне зависимости от конкретной аппаратной инфраструктуры.

Стабильность и сообщество.

Уже много лет развитию Java способствуют сообщество, поддержка Oracle и изобилие приложений и языков на JVM. Кроме того, постоянно выпускаются новые версии Java с новыми интересными функциями. Программы на Java более стабильны по сравнению с программами на других языках. Кроме того, в кратчайшие сроки выпускается новая версия Java с более продвинутыми функциями, что делает ее более стабильной.

Многопоточность.

Java — это многопоточный язык, в Java одновременно может выполняться более одного потока. Поток — это наименьшая единица обработки в программировании. Многопоточность помогает нам получить максимальную загрузку процессора.

Несколько потоков совместно используют общую область памяти и повышают эффективность и производительность приложения. Эти потоки работают независимо друг от друга, не влияя друг на друга.

Автоматическое управление памятью.

Разработчикам Java не нужно вручную писать код для управления памятью благодаря автоматическому управлению памятью (АММ). АММ также используется в языке программирования Swift и при очистке памяти

приложениями, которые автоматически обрабатывают распределение и освобождение памяти. Что именно это означает?

Минусы языка.

Java работает медленно и имеет низкую производительность.

Java потребляет много памяти и значительно медленнее, чем родные языки, такие как C или C++. Он также медленный по сравнению с ними, потому что каждый код должен быть интерпретирован в код машинного уровня.

Эта низкая производительность связана с дополнительным уровнем компиляции и абстракции JVM.

Java обеспечивает не столь привлекательный внешний вид графического интерфейса.

Хоть в Java и существует множество конструкторов для создания графического интерфейса, они не подходят для создания сложного пользовательского интерфейса. При их использовании возникает много несоответствий. Существует множество популярных фреймворков, таких как Swing, SWT, JavaFX, JSF, для создания графического интерфейса. Но они недостаточно зрелы, чтобы разрабатывать сложный пользовательский интерфейс. Выбор одного из них, который может подойти именно вам, может потребовать дополнительных исследований.

Java требует значительного объема памяти.

Java требует значительного объема памяти по сравнению с другими языками, такими как C, C++ или Python. Во время выполнения сборки мусора может отрицательно сказаться эффективность использования памяти и производительность системы.

Подробные и сложные коды.

Коды Java многословны, а это означает, что в них много слов и много длинных и сложных предложений, которые трудно читать и понимать. Это может снизить читабельность кода.

Java фокусируется на том, чтобы быть более управляемой, но в то же время она должна скомпрометировать ее чрезмерно сложными кодами и длинными объяснениями для каждой вещи.» [13]

2.1.2 C++

C++ – компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщенное программирование.

C++, несомненно, является одним из старейших и наиболее эффективных языков программирования, который до сих пор продолжает доминировать в сфере программирования. Каждому новичку интересно узнать о преимуществах и недостатках C++, на основе которых он может начать свой путь программирования.

Плюсы языка.

C++ предлагает функцию переносимости или независимости от платформы, которая позволяет пользователю легко запускать одну и ту же программу в разных операционных системах или интерфейсах. Предположим, вы пишете программу в ОС LINUX и по какой-то очевидной причине переключаетесь на ОС Windows, вы сможете запустить ту же программу и в Windows без каких-либо ошибок. Эта функция является очень удобной для программиста.

Одним из самых больших преимуществ C++ является возможность объектно-ориентированного программирования, которая включает в себя такие понятия, как классы, наследование, полиморфизм, абстракция данных и инкапсуляция, которые обеспечивают повторное использование кода и делают программу еще более надежной.

Поскольку C++ тесно связан с C, который представляет собой процедурный язык, тесно связанный с машинным языком, C++ допускает низкоуровневую

манипуляцию данными на определенном уровне. Встроенные системы и компилятор созданы с помощью C++.

C++ предоставляет программисту полный контроль над управлением памятью. Это можно рассматривать и как преимущество, и как недостаток, поскольку это увеличивает ответственность пользователя за управление памятью. Эта концепция реализована с помощью DMA (динамического выделения памяти) с использованием указателей.

C++ имеет большое сообщество, которое поддерживает его, предоставляя онлайн-курсы и лекции, как платные, так и бесплатные. По статистике, C++ является 6-м наиболее часто используемым тегом в StackOverflow и GitHub.

C++ в значительной степени совместим с C. Практически каждая безошибочная программа на C является корректной программой на C++. В зависимости от используемого компилятора каждая программа C++ может работать с файлом с расширением .cpp.

Минусы языка.

«Главными минусами языка являются его сложность и высокий порог входа. Решения об изменениях в C++ принимают на заседаниях комитета, участие в которых формально может принять любой разработчик. Но отправляться на такие собрания в другую страну за свой счет трудно, поэтому большинство участников — представители крупных IT-корпораций, таких как Google, Яндекс. А вот представителей интересов студентов или стартапов немного. Поэтому эксперты могут принять концепции, которые им кажутся очевидными, а для остальной аудитории они будут сложными.»[14]

Хотя объектно-ориентированное программирование обеспечивает большую безопасность обрабатываемых данных по сравнению с другими языками программирования, которые не являются объектно-ориентированными, такими как C, определенные проблемы с безопасностью все еще существуют из-за наличия дружественных функций, глобальных переменных и указателей.

Как обсуждалось ранее, C++ дает пользователю полный контроль над управлением памятью компьютера с помощью прямого доступа к памяти. В C++ отсутствует функция сборщика мусора для автоматической фильтрации ненужных данных.

C++ не поддерживает никаких встроенных потоков. Потоки — это относительно новая концепция C++, которой изначально не было. Теперь C++ поддерживает лямбда-функции (анонимные функции).

C++ сложно выучить?

«Сложно, но реально. C++ — достаточно серьезный язык, к нему требуется осознанный подход. Желательно иметь общие представления о том, что такое программирование, как устроено «железо» и как работает операционная система.

Программируя на C++, нужно держать в голове сразу много нюансов: отслеживать типы данных, контролировать выделение и освобождение памяти. Сейчас появляется все больше средств, которые перекладывают эти обязанности на сам язык. Уже на этапе написания кода современные IDE (среды разработки) помогают следить за соблюдением стандартов, отлавливают большое количество ошибок, не всегда очевидных человеку без глубокого знания языка, автоматизируют процесс написания большого объема кода с помощью типовых конструкций.»[14]

2.1.3 C#

C# - один из наиболее быстро растущих, востребованных и при этом «удобных» языков программирования. Это модификация фундаментального языка C от компании Microsoft, призванная создать наиболее универсальное средство для разработки программного обеспечения для большого количества устройств и операционных систем.

Основные преимущества языка.

С самого начала C# был основан на принципах объектно-ориентированного программирования (ООП). Эта концепция кодирования предполагает, что вы можете

определить тип и структуру данных, чтобы применить к ним набор стандартных функций. ООП собирает данные в объекты, что упрощает разбиение приложения на более мелкие части, которые быстрее создаются, управляются и комбинируются.

Язык C# считается языком высокого уровня, поскольку его синтаксис напоминает человеческий язык. Другими словами, он имеет высокий уровень абстракции от машинного кода, поэтому нам нужно скомпилировать код, написанный на C#, чтобы аппаратное обеспечение могло понять его команды.

C# относится к C-семейству языков программирования. Хотя само по себе это не выгодно, изучение C# облегчит изучение C, C++ или Java, поскольку эти языки используют схожий синтаксис и принципы программирования.

Управление распределением памяти — одна из важнейших задач для поддержания производительности приложения. Для этого в C# есть встроенный сборщик мусора.

Недостатки языка C#.

C# полностью основан на платформе Microsoft .Net, поэтому это не гибкий язык.

Как мы знаем, если мы что-то изменим в написанном на C# коде, мы должны сначала его скомпилировать. Еще одна вещь: когда мы запускаем программное обеспечение WPF, у нас возникает проблема с медленной загрузкой, и это причина, по которой C# работает медленно.

Что разрабатывают с помощью C#.

Язык C# практически универсален. Можно использовать его для создания любого ПО: продвинутых бизнес-приложений, видеоигр, функциональных веб-приложений, приложений для Windows, macOS, мобильных программ для iOS и Android.

2.1.4 Python

Python – это язык программирования общего назначения, который широко применяется в различных областях. Язык скриптовый, он универсален и является одним из самых популярных языков программирования в мире.[4]

Что пишут на Python? Что вздумается. Любой программный продукт, независимо от сложности и специфики, можно создать на Python и одним из сотен фреймворков, разработанных для него. [19]

«Возможности языка.

Как я уже писал ранее, Python применяют в различных сферах. Разработчики со всего мира полюбили этот язык программирования за множество возможностей.

Объектно-ориентированное программирование (ООП). Реализация ООП в Python хоть и специфична по сравнению с иными объектно-ориентированными языками, но одновременно является неплохо продуманной.

Обобщенное и функциональное программирование.

Модули и пакеты. Программное обеспечение (ПО) на Python оформляется в виде модулей, которые могут быть собраны в пакеты.

Интроспекция. Это возможность запросить тип и структуру объекта во время выполнения программы. В Python для любого объекта можно узнать всю информацию о его внутренней структуре.[20]

Обработка исключений, итераторы и генераторы:

- управление контекстом выполнения;
- декораторы, регулярные выражения.

Преимущества Python.

Язык характеризуется логичным синтаксисом, вследствие чего исходный код программ, написанных «на питоне», легко читается и воспринимается.[21]

Еще одно из преимуществ данного языка программирования — его условная легкость. Он считается наиболее подходящим для начинающих специалистов: разрабатывать несложные программы можно научиться уже спустя пару-тройку дней изучения.[18]

Большое интернет-сообщество. Если разработчик сталкивается с вопросами и трудностями, он всегда может спросить совета у коллег, что значительно ускоряет решение проблем.

Гибкость и масштабируемость. Python позволяет разработчикам адаптировать высокоуровневую логику приложения, что позволяет легко расширять сложные приложения по мере необходимости.[17]

Разработка на Python идет быстрее, чем на большинстве других языках.

Python является интерпретируемым языком программирования. Это значит, что до запуска он представляет собой обычный текстовый файл. Соответственно, программировать можно почти на всех платформах.»[15]

Также стоит отметить, что Python является самым популярным языком в области Data Processing (рисунок 4).

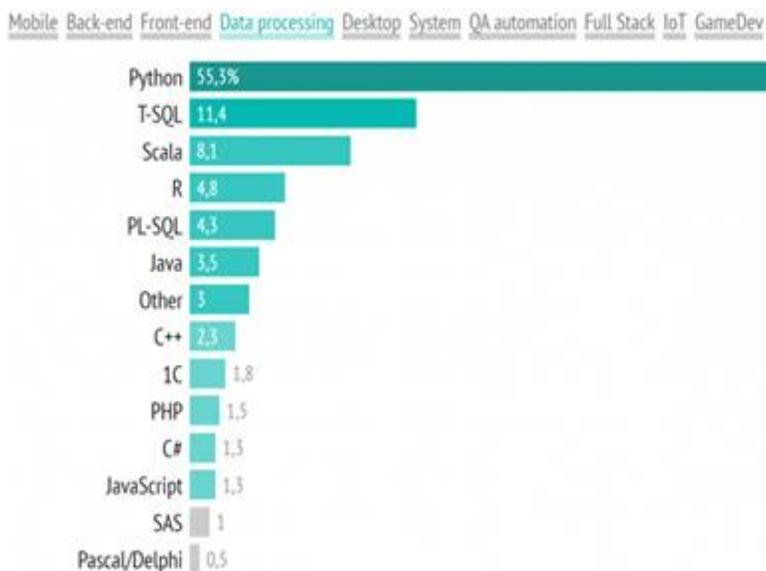


Рисунок 4 - Популярность языков в Data Processing

Подведём итог.

Ранее были приведены преимущества и недостатки нескольких языков программирования. Опираясь на факты и личный опыт, для разработки приложения я выбрал язык программирования Python.[5]

Выводы по параграфу.

В рамках данного параграфа описывается выбор языка программирования. Выделены основные преимущества и недостатки каждого языка.

2.2 Физическое моделирование АИС

2.2.1 Выбор архитектуры АИС

На этапе физического моделирования создается модель, которая представляет собой «совокупность структуры, методов и средств редуцированного воплощения системы, необходимая для проверки в реальных условиях работоспособности будущей системы и адекватности ее моделей».[2]

Разрабатываемая система строится на базе десктопного приложения. Т.е. это обычное приложение, скомпилированное в виде исполняемого файла для Microsoft Windows.

2.2.2 Функциональность проекта

При запуске приложения (файл main.exe) происходит отображение главной формы приложения, которая включает в себя три вкладки. Каждая вкладка – отдельная ветвь вычислений.

2.2.3 Структура проекта

Проект состоит из одного файла, написанного на языке python – main.py.

Python – современный язык программирования, который позволяет создавать приложения, которые работают на различных платформах: как

мобильных, так и десктопных, как на ОС семейства Windows, так и на ОС семейства UNIX. В этом его основное преимущество.[6]

По Python имеется множество литературы. В сети интернет у языка python активное многоязычное сообщество пользователей. Разработано множество библиотек и модулей, расширяющих стандартный функционал языка. Благодаря этому Python активно используют в анализе данных, парсинге, написании веб-приложений, приложений для мобильных устройств. Даже программное обеспечение некоторых космических аппаратов написано именно на данном языке программирования. [7]

2.2.4 Описание используемых библиотек

Библиотеки, используемые при разработке продукта:

«tkinter – кросс-платформенная событийно-ориентированная графическая библиотека на основе средств Tk, написанная Стином Лумхольтом и Гвидо ван Россумом. Входит в стандартную библиотеку Python. Tkinter — это свободное программное обеспечение, распространяемое под Python-лицензией.

Tk является базовым классом любого Tkinter приложения. При создании объекта этого класса запускается интерпретатор tcl/tk и создаётся базовое окно приложения.

Tkinter является событийно-ориентированной библиотекой. В приложениях такого типа имеется главный цикл обработки событий. В Tkinter такой цикл запускается методом `mainloop`. Для явного выхода из интерпретатора и завершения цикла обработки событий используется метод `quit`.

Таким образом минимальное приложение на Tkinter будет таким:

```
from tkinter import *
root = Tk()
root.mainloop()
```

В приложении можно использовать несколько интерпретаторов tcl/tk. Так как после вызова метода `mainloop` дальнейшие команды `python` исполняться не будут до выхода из цикла обработки событий, необходимо метод `mainloop` всех интерпретаторов кроме последнего осуществлять в фоновом режиме. Пример запуска двух интерпретаторов:

```
from tkinter import *
root1 = Tk()
root2 = Tk()
root1.after(500, root1.mainloop) # первый цикл запускаем в фоне
root2.mainloop()
```

При использовании двух и более интерпретаторов необходимо следить, чтобы объекты, созданные в одном интерпретаторе, использовались только в нём. Например, изображение, созданное в первом интерпретаторе, может быть использовано много раз в этом же интерпретаторе, но не может быть использовано в других интерпретаторах. Необходимость в запуске нескольких интерпретаторов в одном приложении возникает крайне редко. Для создания дополнительного окна приложения в большинстве случаев достаточно виджета `Toplevel`.»[16]

`scipy` – это библиотека с открытым исходным кодом, используемая для решения математических, научных, инженерных и технических задач. Базовая структура данных, используемая `SciPy`, представляет собой многомерный массив, предоставляемый модулем `NumPy`. `NumPy` предоставляет некоторые функции для линейной алгебры, преобразований Фурье и генерации случайных чисел, но не с общностью эквивалентных функций в `SciPy`.

«Библиотека `SciPy` позволяет выполнять расширенные математические операции — математический анализ и линейную алгебру, различные преобразования, инженерные расчеты. Этот пакет имеет функции обработки

изображений и построения графиков. Отличительной чертой SciPy является использование эффективных структур данных, созданных с помощью NumPy.

В SciPy есть функции из линейной алгебры, вычисления интегралов и решения дифференциальных уравнений, реализация различных сложных алгоритмов. Операций и методов больше, чем в NumPy, они более трудоемки. Зато их написание в виде кода очень простое.» [11]

«Основные модули SciPy для научных вычислений:

- `scipy.special` — специальные функции, возможности и понятия из математической физики;
- `scipy.integrate` — функции для численного интегрирования и решения обыкновенных дифференциальных уравнений;
- `scipy.optimize` — алгоритмы оптимизации (в т.ч. минимизации математических функций);
- `scipy.interpolate` — методы для интерполяции, т.е. для приближенного нахождения какой-либо величины по уже известным отдельным ее значениям;
- `scipy.fft` — преобразования Фурье (операции, сопоставляющие одной функции вещественной переменной другую функцию вещественной переменной);
- `scipy.signal` — методы для обработки и преобразования сигналов;
- `scipy.linalg` — операции линейной алгебры. Модули с этим названием есть и в NumPy, и в SciPy, но их возможности различаются. В SciPy больше продвинутых функций и методов;
- `scipy.sparse.csgraph` — методы для работы с разреженными графами, особыми структурами данных;
- `scipy.spatial` — функции для работы с пространственными структурами данных и алгоритмами;
- `scipy.stats` — операции для статистических расчетов;
- `scipy.ndimage` — обработка многомерных изображений;
- `scipy.io` — ввод и вывод, загрузка и сохранение файлов.

Пакеты необходимо импортировать отдельно.» [11]

«Установить библиотеку можно несколькими способами:

- скачать научный дистрибутив. Этот способ рекомендуют разработчики;

- скачать и установить IDE, затем загрузить через нее SciPy. Среды для программирования на Python позволяют легко подключать библиотеки. Примеры IDE — Pyzo или PyCharm;

- воспользоваться менеджером `pip`. Это программа для загрузки, обновления и удаления официальных пакетов Python. Она встроена в язык программирования и автоматически устанавливается вместе с ним. Команда `python -m pip install name` загрузит и установит пакет с нужным именем. В нашем случае это `scipy`. При установке SciPy через Anaconda — дистрибутив языков программирования Python и R с набором свободных библиотек для Data Science и машинного обучения — команды будут другими;

- для пользователей Linux или MacOS — использовать встроенный файловый менеджер. В Ubuntu и Debian это `apt-get`, для Fedora — `dnf`, в MacOS — `brew`.» [11]

`numpy` — библиотека с открытым исходным кодом для языка программирования Python. Возможности: поддержка многомерных массивов; поддержка высокоуровневых математических функций, предназначенных для работы с многомерными массивами. Эта библиотека имеет несколько важных функций, которые делают ее популярным инструментом. Во-первых, её исходный код находится в свободном доступе на GitHub, поэтому NumPy называют open-source модулем для Python.

«Во-вторых, NumPy написана на C и Fortran. Это компилируемые языки (языки программирования, преобразующие текст в машинный код — набор инструкций для конкретного типа процессора. Преобразование производится с помощью специальных программ-компиляторов, благодаря чему вычисления на

компилируемых языках происходят быстрее), на которых вычисления производятся гораздо быстрее и эффективнее, чем на интерпретируемых языках (языки программирования, которые не заточены под конкретный тип процессора и могут быть запущены на разных типах устройств). Сам Python является одним из этих языков.»[11]

«Где используется NumPy(рисунок 5):

– научные вычисления. NumPy используется учеными для решения многомерных задач в области математики и физики, биоинформатики, вычислительной химии и даже когнитивной психологии;

– создание новых массивных библиотек. На основе NumPy появились новые типы массивов с возможностями, выходящими за рамки того, что предлагала библиотека. Например, библиотеки Dask, CuPy или XND;

– data Science. NumPy лежит в основе экосистемы интеллектуального анализа данных. Библиотека используется для всех этапов обработки данных: извлечения и преобразования, анализа, моделирования и оценки, репрезентация;

– machine Learning. Библиотеки машинного обучения scikit-learn и SciPy также извлекают выгоду из вычислительной мощности NumPy;

– визуализация данных. По сравнению непосредственно с Python возможности NumPy позволяют исследователям визуализировать большие наборы данных, которые гораздо больше по размеру. Например, эта библиотека лежит в основе системы PyViz, в которую входят десятки программ для визуализации.» [11]

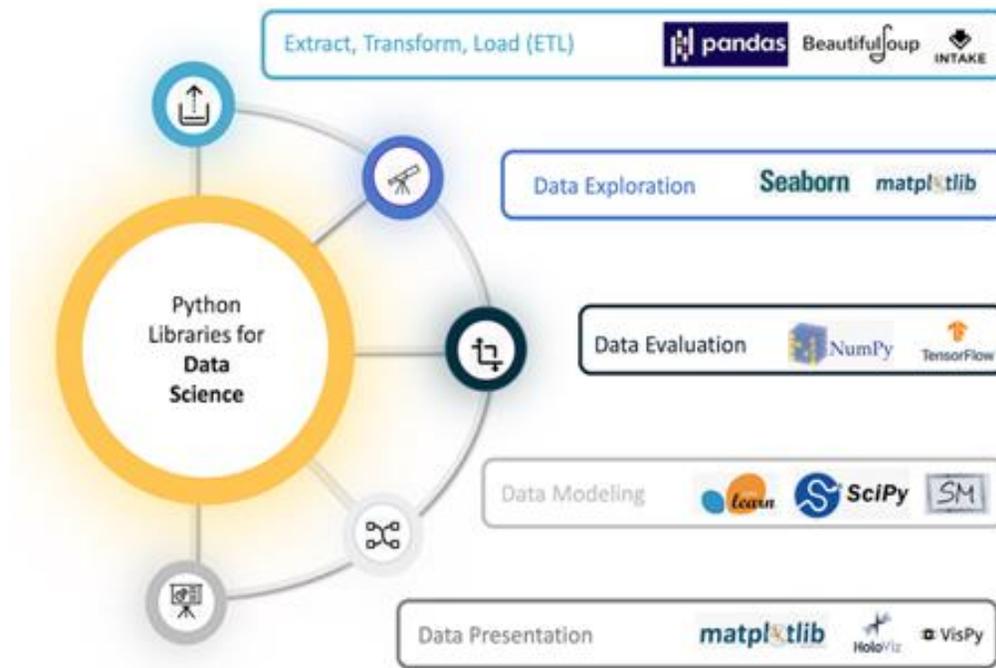


Рисунок 5 – Экосистема для анализа больших данных на основе NumPy

«Как работает NumPy.

Давайте посмотрим, как устроены массивы (рисунок 6), которые обрабатывает NumPy. Рассмотрим однородный двумерный массив. Он выглядит как простая таблица — две оси значений и ячейки внутри (элементы массива). Если появляется третья ось, то массив становится трехмерным. Важное условие является то, что все элементы должны иметь единый тип данных, например, только целые числа.» [11]

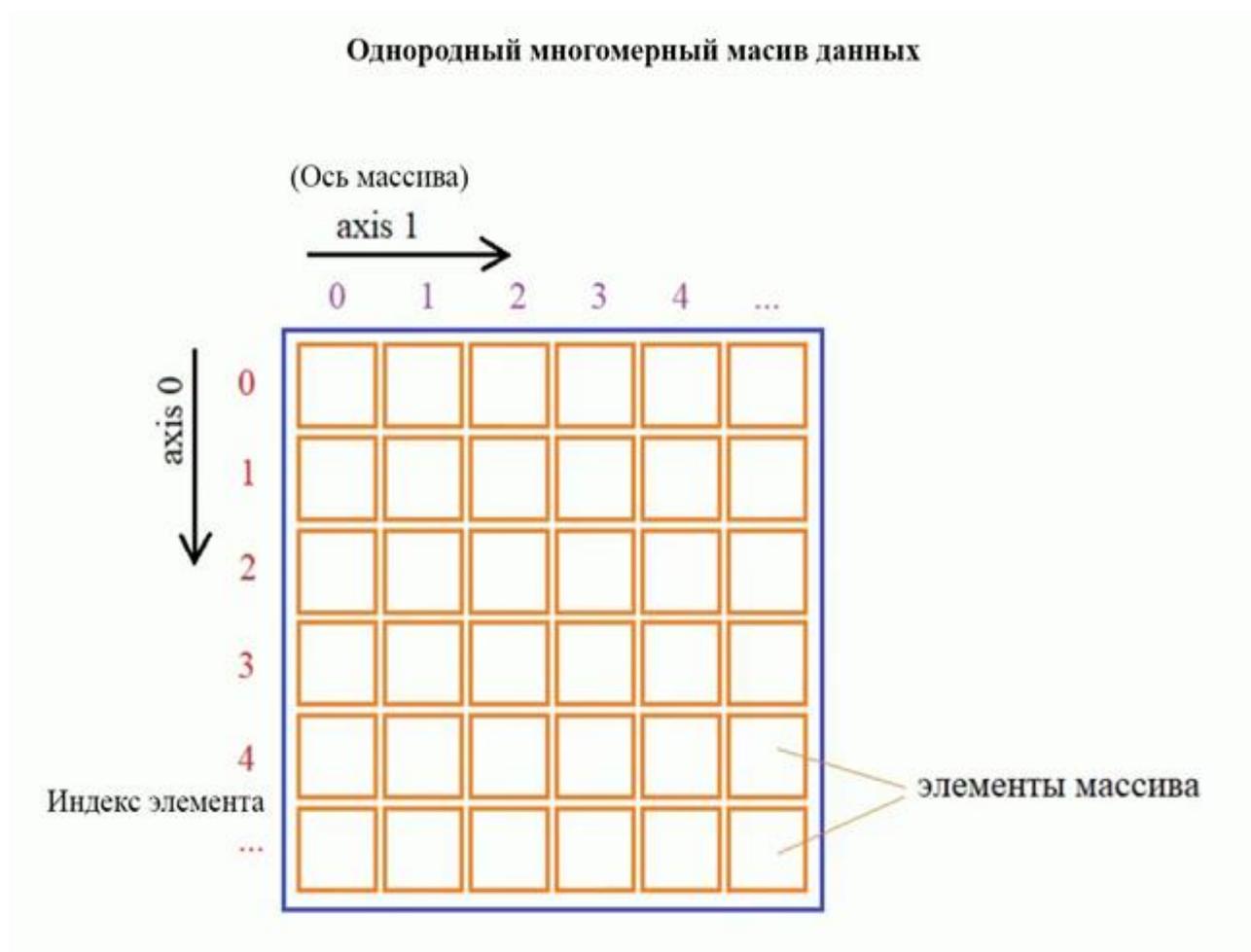


Рисунок 6 – Пример визуализации двумерного массива

«Конечно, помимо двумерных массивов, библиотека NumPy может обрабатывать и другие массивы с различным количеством осей. Эта изменчивость представляют числом N , как и любую переменную в математической задаче. Поэтому обычно говорят, что NumPy работает с N -мерными массивами данных.

С этими данными NumPy выполняет вычисления с использованием математических функций, генераторов случайных чисел, линейных уравнений или преобразований Фурье. Например, вы можете использовать метод `linalg.solve` для решения системы:

```
import numpy as np*  
left = np.array([[1,3],[2,-4]])  
right = np.array([9,8])  
np.linalg.solve(left,right)
```

Ответ: array([6.,1.])»[11]

«Как и сам Python, библиотеку NumPy легко освоить и использовать. Для начала работы достаточно понять концепцию массивов. Например, в базовых арифметических вычислениях есть способ обработки массивов, который называют трансляцией или broadcasting.» [11]

«Если в массиве значения указаны в милях, а результат нужно получить в километрах, вы можете умножить его на простое число 1,6 (скалярное значение). NumPy принимает самостоятельное решение умножить на заданное число каждый элемент в массиве, и пользователю не приходится прописывать для этого отдельную команду.» [11]

«Подробной документации NumPy на русском языке до сих пор нет, а в русском интернете можно найти только короткие выдержки, в которых упущены многие моменты. Поэтому, чтобы стать экспертом высокого уровня в Data Science или Machine Learning, придется подтянуть английский. Начать можно с информации на официальном сайте.» [11]

Как загрузить библиотеку.

«Для загрузки NumPy просто введите в командной строке: `import numpy as np`. Не удивляйтесь, если библиотека окажется установленной, ведь модуль NumPy входит в состав множества пакетов и других библиотек.» [11]

«Конечно, можно написать просто `import numpy`, ошибки в этом не будет. Но `np` — это общепринятое название, которое стало правилом и упростило процесс написания кода, поэтому, один раз прописав `import numpy as np`, в последующих строках можно использовать `np` вместо `numpy`.» [11]

Выводы по параграфу.

В данном параграфе приведена и описана физическая модель приложения.

2.3 Технологическое обеспечение задачи

2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

Работа с программой начинается с загрузки на экран главной формы приложения для входа в систему.

Для взаимодействия с системой применяются событийный и диалоговый режимы.

Событиями называются процессы двух видов: активизируемые пользователем, а также программные события, на основании которых выполняются процедуры контроля допустимости данных.[9]

2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

Приложение позволяет вводить данные на форме и выводить результаты (вычисленные данные) на экран.

Выводы по параграфу.

В данном параграфе приводится описание технологического обеспечения задачи.

Выводы по главе 2.

В рамках данной главы приводится описание процессов разработки и реализации информационной системы.

Глава 3 Тестирование приложения

В процессе тестирования приложения доказывается корректность его работы. Для этого выполняется проверка работы всех возможных сценариев.

Первая форма приложения, которую видит пользователь – главное окно с открытой первой вкладкой (рисунок 7).

Приложение имеет три вкладки:

- ПГЗ и ОГЗ;
- обр. засечка;
- ход в 2 стороны.

The image shows a screenshot of a Windows-style application window titled "Калькулятор". The window has a standard title bar with minimize, maximize, and close buttons. Below the title bar, there are three tabs: "ПГЗ и ОГЗ", "Обр. засечка", and "Ход в 2 стороны". The "ПГЗ и ОГЗ" tab is currently selected. The main content area of the window contains a list of input fields for variables: x1, y1, d, alpha, x1(2), y1(2), x2, and y2. Each variable label is followed by a horizontal input field. In the bottom right corner of the main area, there is a button labeled "Подсчитать".

Рисунок 7 – Форма

Используя эту форму, пользователь может в любой момент перейти на необходимую ему в данный момент вкладку и произвести корректировку входных данных и/или произвести вычисления.

Первая вкладка с введенными исходными данными показана на рисунке 8.

При нажатии на кнопку Подсчитать, произойдут вычисления. Результаты вычислений по первой вкладке приведены на рисунке 9.

x1	0
y1	0
d	0
α	2
x1(2)	41300
y1(2)	05250
x2	42540
y2	06050

Подсчитать

Рисунок 8 – Входные данные на первой вкладке

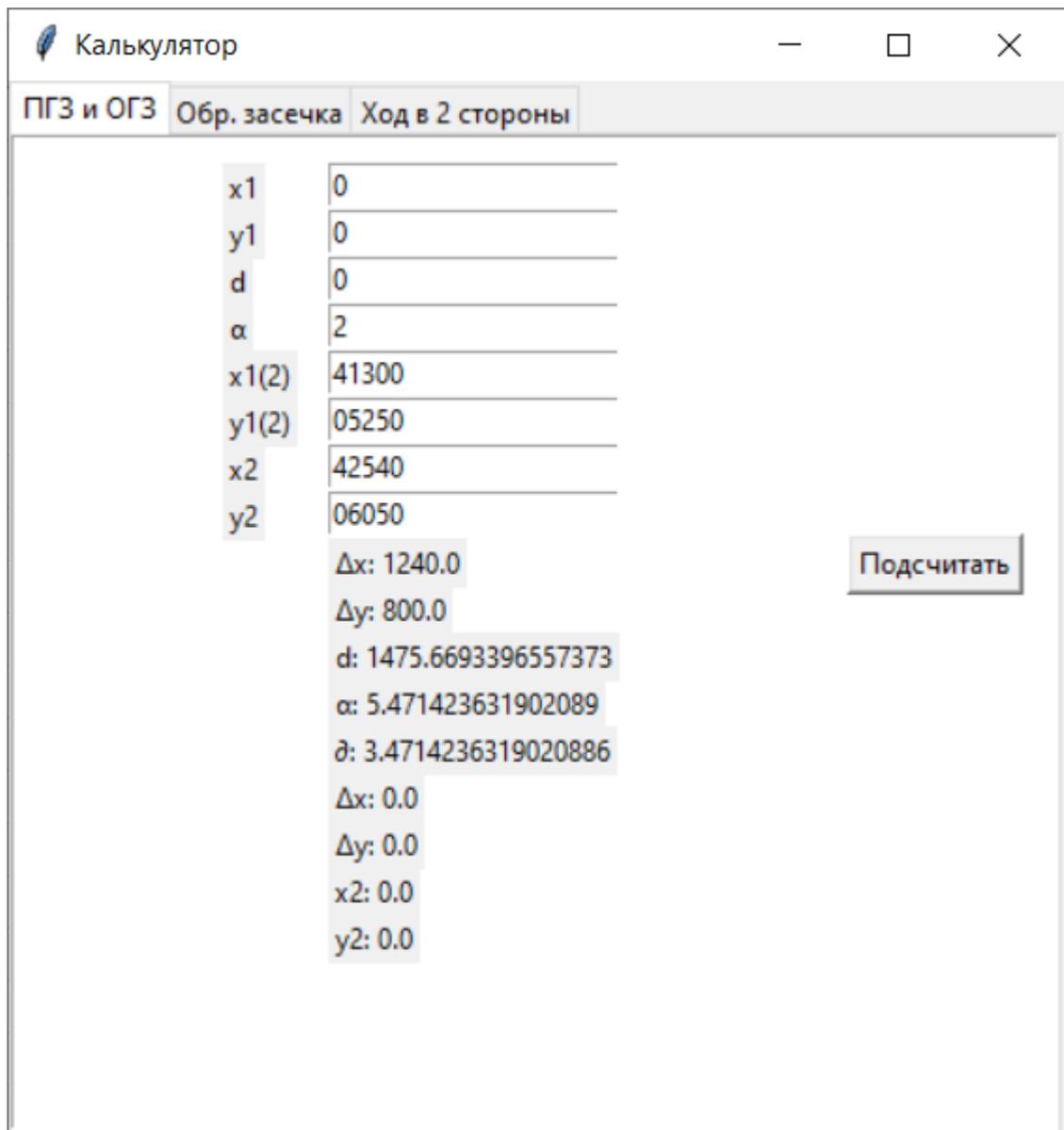


Рисунок 9 – Результаты расчетов на первой вкладке

Вторая вкладка с введенными исходными данными показана на рисунке 10.

При нажатии на кнопку Подсчитать, произойдут вычисления. Результаты вычислений по второй вкладке приведены на рисунке 11.

The image shows a window titled "Калькулятор" (Calculator) with three tabs: "ПГЗ и ОГЗ", "Обр. засечка", and "Ход в 2 стороны". The "Ход в 2 стороны" tab is selected. The window contains six input fields for variables and a "Подсчитать" (Calculate) button. The values entered in the fields are: alpha1 = 34,84; alpha2 = 6,11; x1 = 57220; y1 = 09950; x2 = 65100; and y2 = 60896.

Variable	Value
α_1	34,84
α_2	6,11
x_1	57220
y_1	09950
x_2	65100
y_2	60896

Рисунок 10 – Входные данные на второй вкладке

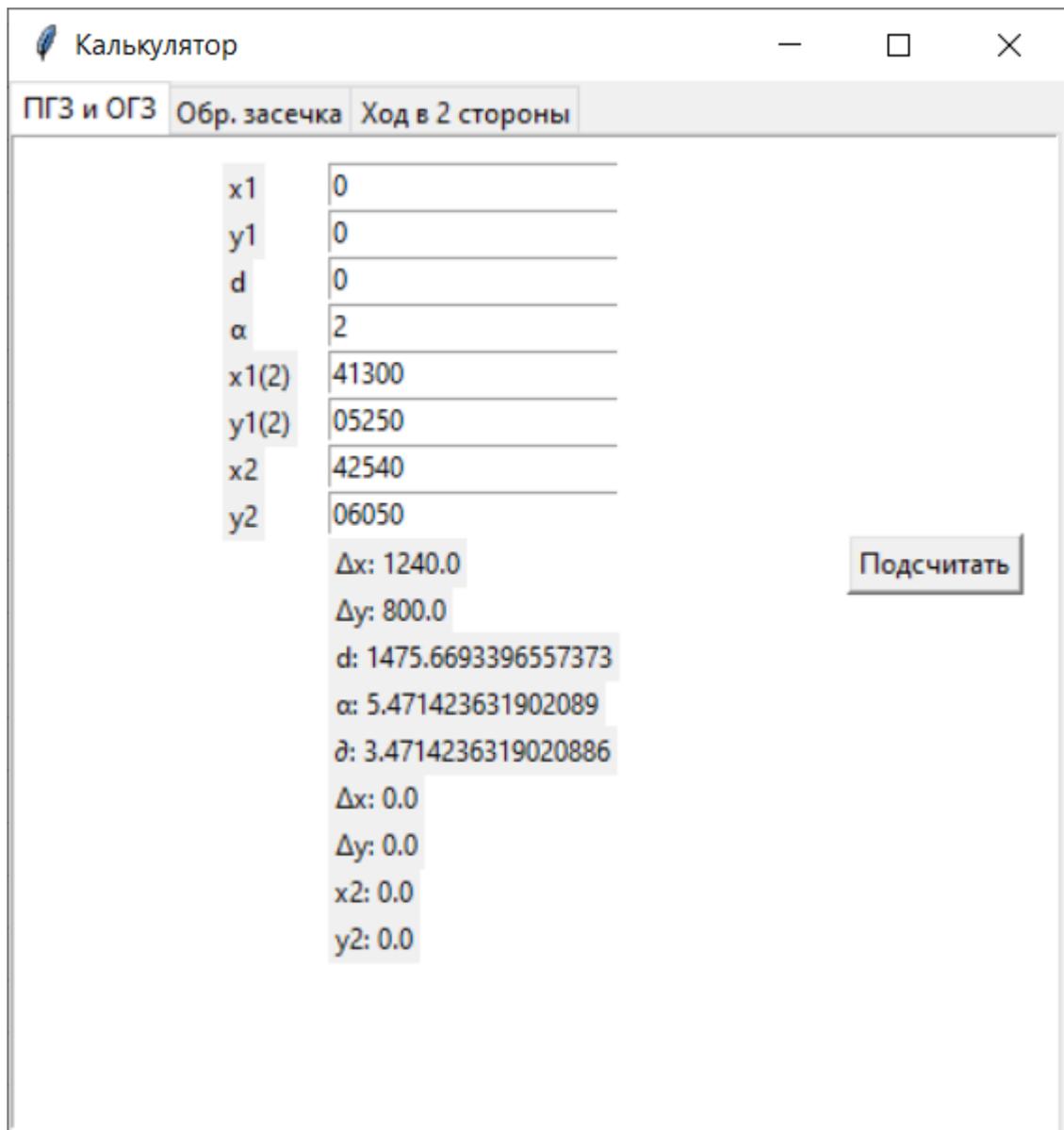


Рисунок 11 – Результаты расчетов на второй вкладке

Третья вкладка с введенными исходными данными показана на рисунке 12.

При нажатии на кнопку Подсчитать, произойдут вычисления. Результаты вычислений по третьей вкладке приведены на рисунке 13.

x	23826
y	41977
d1	718
α 1	15.91
d2	789
α 2	12.65

Подсчитать

Рисунок 12 – Входные данные на третьей вкладке

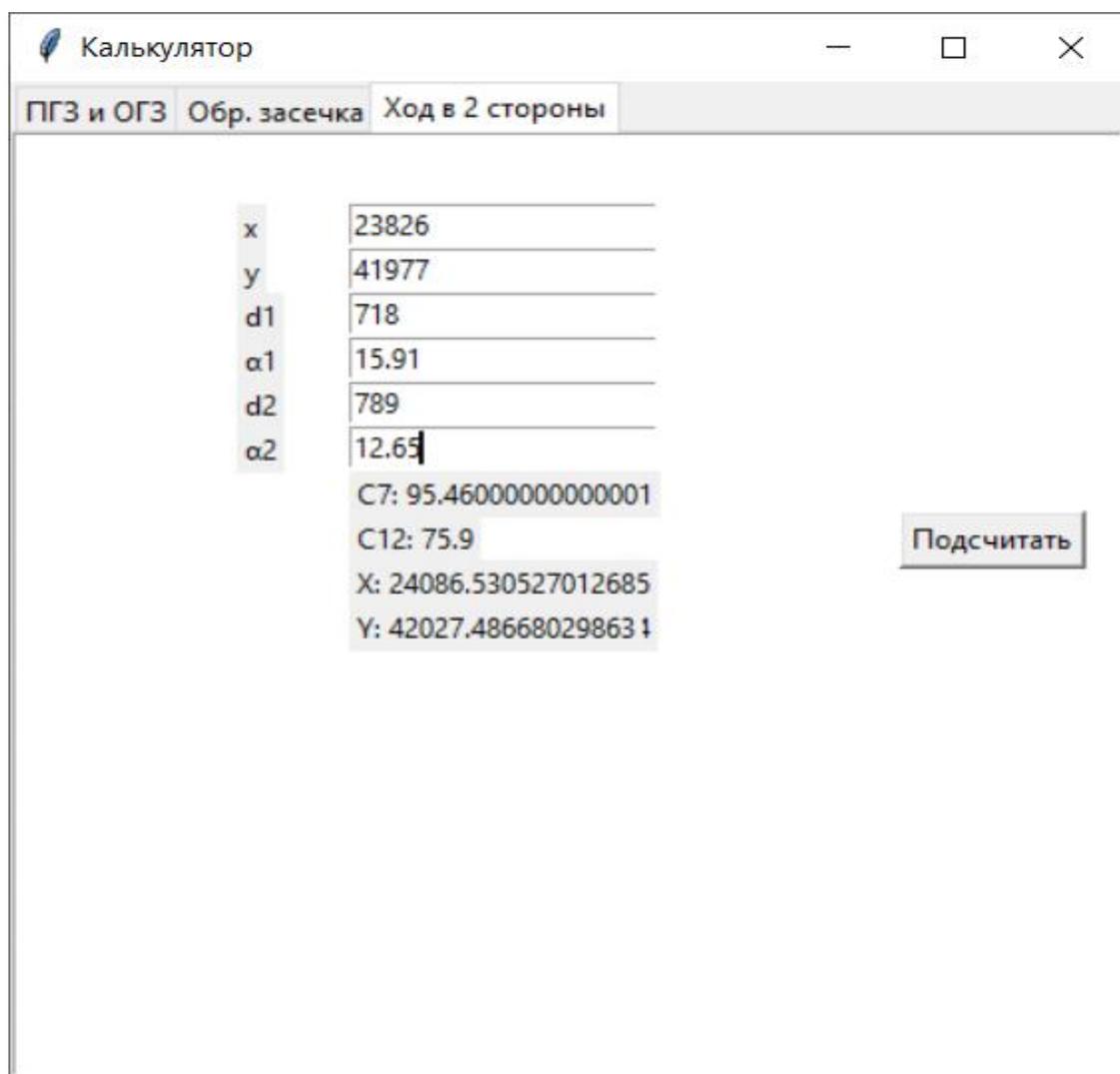


Рисунок 13 – Результаты расчетов на третьей вкладке

Выводы по параграфу.

В параграфе приведено описание работы приложения и продемонстрирован контрольный пример работы.

Выводы по главе 3.

В рамках данной главы приводится итоговое тестирование разработанного приложения. Полный программный код можно увидеть в приложении (см. Приложение А).

Заключение

В рамках выполнения данной выпускной квалификационной работы было создано программное обеспечение для вычислений с использованием Python.

Затронутая тема является очень актуальной, т.к. на сегодняшний день автоматизация расчетов определяет экономию времени на выполнение рутинных и сложных вычислений, что приводит к высокой производительности труда.

Одним из достоинств данной программы является простота (все приложение состоит из одного исполняемого файла). Среди минусов можно отметить, что нет достаточной наглядности. Данные недостатки можно исправить в последующих версиях программы. Кроме того, в перспективах развития программы можно добавить в программу справочный раздел, в котором будут объяснения расчетов всех уравнений.

Также можно сделать программу более визуально ориентированную на пользователя, т.е. более подробно подписать поля, а поле с выводом результата упростить, сделав его более понятным для человека, который недостаточно хорошо разбирается в математике.

Данное приложение успешно показало себя на тестах. Приложение было передано для тестирования Военному учебному центру при ТГУ и получило хорошие отзывы как от преподавателей, так и от студентов

Подводя итог, можно сделать вывод, что разработанное программное обеспечение успешно функционирует и позволяет автоматизировать расчеты, что будет довольно полезно для Военного учебного центра ТГУ. Данное приложение также может быть полезным для других военных учебных организаций.

Список используемой литературы и используемых источников

1. Адигеев М.Г. Жизненный цикл программного обеспечения. – Ростов-на-Дону: Изд-во ЮФУ, 2013. – 41 с.
2. Бабанов А.М. Технология разработки программного обеспечения: структурный подход. – Томск: ТГУ, 2016. – 157 с.
3. Рудаков А.В. Технология разработки программных продуктов. – М.: Академия, 2014. – 190 с.
4. Бизли, Дэвид М. Python. Подробный справочник / Бизли Дэвид М.. - М.: Символ-плюс, 2014. - 751 с.
5. Гифт, Ноа Python в системном администрировании UNIX и Linux / Ноа Гифт. - М.: Символ-плюс, 2018. - 315 с.
6. Марк, Лутц Изучаем Python / Лутц Марк. - М.: Символ-плюс, 2019. - 225 с.
7. Прохоренок, Н. А. Python. Самое необходимое (+ DVD-ROM) / Н.А. Прохоренок. - М.: БХВ-Петербург, 2019. - 416 с.
8. Россум, Г. Язык программирования Python / Г. Россум, Ф.Л.Дж. Дрейк, Д.С. Откидач и др.. - М.: [не указано], 2019. - 263 с.
9. Саммерфилд, Марк Программирование на Python 3. Подробное руководство / Марк Саммерфилд. - М.: Символ-плюс, 2016. - 274 с.
10. Соловьев, И. А. Вычислительная математика на смартфонах, коммуникаторах и ноутбуках с использованием программных сред Python / И.А. Соловьев, А.В. Червяков, А.Ю. Репин. - М.: Лань, 2017. - 272 с.
11. SciPy and NumPy Libraries. [Электронный ресурс] / Режим доступа: URL: <https://blog.skillfactory.ru/>
12. Форсье, Джефф Django. Разработка веб-приложений на Python / Джефф Форсье. - М.: Символ-плюс, 2020. - 654 с.

13. Скрытые возможности в Java [Электронный ресурс] / Режим доступа:
URL: <https://webstudio-uwk.ru/>
14. Язык программирования C++ [Электронный ресурс] / Режим доступа:
URL: <https://blog.skillfactory.ru/>
15. Военный учебный центр ТГУ [Электронный ресурс] / Режим доступа:
URL: <https://tltsu.ru/>
16. Библиотека tKinter [Электронный ресурс] / Режим доступа: URL: it-black.ru/
17. Martin Gioldmeyr. Python Fastlane / M. Gioldmeyr. – Independently publisher, 2020. - 129 с.
18. Michael Learn. Learn Python programming / M. Learn. – Independently publisher, 2019. - 202 с.
19. Nat Dunn, Webucator. Python 3.8 / N. Dunn. – Webucator, 2020. - 554 с.
20. Christian Mayer. Python One-Liners / C. Mayer. – No Starch Press, 2020. - 216 с.
21. Al Sweigart. Automate the Boring Stuff with Python / A. Sweigart. – No Starch Press, 2019. – 592 с.

Приложение А

Листинг программы

```
from tkinter import *
import scipy
import numpy
import tkinter.ttk as ttk
root = Tk()
root.title("Калькулятор")
root.geometry("450x450")
#####

nb = ttk.Notebook(root) # создание вкладок
nb.pack(fill='both', expand='yes') # отображение компонента nb

# вкладки
f1 = Text(root)
f2 = Text(root)
f3 = Text(root)

# названия вкладок
nb.add(f1, text='ПГЗ и ОГЗ')
nb.add(f2, text='Обр. засечка')
nb.add(f3, text='Ход в 2 стороны')

#####
```

Продолжение Приложения А

ВВОД ДАННЫХ 1

B1_var = StringVar()

B1_entry = Entry(f1, textvariable=B1_var, justify=LEFT)

B1_entry.place(relx=.3, y=10)

B1_label = Label(f1, text='x1', justify=RIGHT)

B1_label.place(relx=.2, y=10)

B2_var = StringVar()

B2_entry = Entry(f1, textvariable=B2_var, justify=LEFT)

B2_entry.place(relx=.3, y=30)

B2_label = Label(f1, text='y1', justify=RIGHT)

B2_label.place(relx=.2, y=30)

B3_var = StringVar()

B3_entry = Entry(f1, textvariable=B3_var, justify=LEFT)

B3_entry.place(relx=.3, y=50)

B3_label = Label(f1, text='d', justify=RIGHT)

B3_label.place(relx=.2, y=50)

B4_var = StringVar()

B4_entry = Entry(f1, textvariable=B4_var, justify=LEFT)

B4_entry.place(relx=.3, y=70)

B4_label = Label(f1, text='α', justify=RIGHT)

B4_label.place(relx=.2, y=70)

E1_var = StringVar()

Продолжение Приложения А

```
E1_entry = Entry(f1, textvariable=E1_var, justify=LEFT)
E1_entry.place(relx=.3, y=90)
E1_label = Label(f1, text='x1(2)', justify=RIGHT)
E1_label.place(relx=.2, y=90)
```

```
E2_var = StringVar()
E2_entry = Entry(f1, textvariable=E2_var, justify=LEFT)
E2_entry.place(relx=.3, y=110)
E2_label = Label(f1, text='y1(2)', justify=RIGHT)
E2_label.place(relx=.2, y=110)
```

```
E3_var = StringVar()
E3_entry = Entry(f1, textvariable=E3_var, justify=LEFT)
E3_entry.place(relx=.3, y=130)
E3_label = Label(f1, text='x2', justify=RIGHT)
E3_label.place(relx=.2, y=130)
```

```
E4_var = StringVar()
E4_entry = Entry(f1, textvariable=E4_var, justify=LEFT)
E4_entry.place(relx=.3, y=150)
E4_label = Label(f1, text='y2', justify=RIGHT)
E4_label.place(relx=.2, y=150)
```

```
#####
```

```
# вычисления на 1 вкладке
```

Продолжение Приложения А

```
def count0():  
    B1 = float(B1_var.get())  
    B2 = float(B2_var.get())  
    B3 = float(B3_var.get())  
    B4 = float(B4_var.get())  
    E1 = float(E1_var.get())  
    E2 = float(E2_var.get())  
    E3 = float(E3_var.get())  
    E4 = float(E4_var.get())  
    C4 = B4 * 6  
    E6 = E3 - E1  
    E7 = E4 - E2  
    E6_label = Label(f1, text='Δx: '+str(E6), justify=CENTER)  
    E6_label.place(relx=.3, y=170)  
    E7_label = Label(f1, text='Δy: '+str(E7), justify=CENTER)  
    E7_label.place(relx=.3, y=190)  
    E8 = numpy.sqrt(E6*E6+E7*E7)  
    E8_label = Label(f1, text='d: '+str(E8), justify=CENTER)  
    E8_label.place(relx=.3, y=210)  
    if numpy.arctan2(E6, E7) < 0:  
        F9 = numpy.degrees(numpy.arctan2(E7, E6)) + 360  
    else:  
        F9 = numpy.degrees(numpy.arctan2(E7, E6)) + 0  
    E9 = F9 / 6  
    E9_label = Label(f1, text='α: '+str(E9), justify=CENTER)  
    E9_label.place(relx=.3, y=230)
```

Продолжение Приложения А

```
if E9-B4>30:
```

```
    E11 = E9-B4-60
```

```
else:
```

```
    E11 = E9-B4
```

```
E11_label = Label(f1, text='∂: '+str(E11), justify=CENTER)
```

```
E11_label.place(relx=.3, y=250)
```

```
B7 = B3 * numpy.cos(C4 * scipy.pi / 180)
```

```
B7_label = Label(f1, text='Δx: '+str(B7), justify=CENTER)
```

```
B7_label.place(relx=.3, y=270)
```

```
B8 = B3 * numpy.sin(C4 * scipy.pi / 180)
```

```
B8_label = Label(f1, text='Δy: '+str(B7), justify=CENTER)
```

```
B8_label.place(relx=.3, y=290)
```

```
B9 = B1+B7
```

```
B10 = B2+B8
```

```
B9_label = Label(f1, text='x2: '+str(B9), justify=CENTER)
```

```
B9_label.place(relx=.3, y=310)
```

```
B10_label = Label(f1, text='y2: '+str(B10), justify=CENTER)
```

```
B10_label.place(relx=.3, y=330)
```

```
#####
```

```
btn0 = Button(f1, text="Подсчитать", command=count0)
```

```
btn0.place(relx=.8, rely=.4)
```

```
#####
```

Продолжение Приложения А

ВВОД ДАННЫХ 2

B1b_var = StringVar()

B1b_entry = Entry(f2, textvariable=B1b_var, justify=LEFT)

B1b_entry.place(relx=.3, rely=.1)

B1b_label = Label(f2, text='α1', justify=RIGHT)

B1b_label.place(relx=.2, rely=.1)

B2b_var = StringVar()

B2b_entry = Entry(f2, textvariable=B2b_var, justify=LEFT)

B2b_entry.place(relx=.3, rely=.2)

B2b_label = Label(f2, text='α2', justify=RIGHT)

B2b_label.place(relx=.2, rely=.2)

B5b_var = StringVar()

B5b_entry = Entry(f2, textvariable=B5b_var, justify=LEFT)

B5b_entry.place(relx=.3, rely=.3)

B5b_label = Label(f2, text='x1', justify=RIGHT)

B5b_label.place(relx=.2, rely=.3)

B6b_var = StringVar()

B6b_entry = Entry(f2, textvariable=B6b_var, justify=LEFT)

B6b_entry.place(relx=.3, rely=.4)

B6b_label = Label(f2, text='y1', justify=RIGHT)

B6b_label.place(relx=.2, rely=.4)

B7b_var = StringVar()

Продолжение Приложения А

```
B7b_entry = Entry(f2, textvariable=B7b_var, justify=LEFT)
```

```
B7b_entry.place(relx=.3, rely=.5)
```

```
B7b_label = Label(f2, text='x2', justify=RIGHT)
```

```
B7b_label.place(relx=.2, rely=.5)
```

```
B8b_var = StringVar()
```

```
B8b_entry = Entry(f2, textvariable=B8b_var, justify=LEFT)
```

```
B8b_entry.place(relx=.3, rely=.6)
```

```
B8b_label = Label(f2, text='y2', justify=RIGHT)
```

```
B8b_label.place(relx=.2, rely=.6)
```

```
#####
```

```
def count1():
```

```
    B1b = float(B1b_var.get())
```

```
    B2b = float(B2b_var.get())
```

```
    B5b = float(B5b_var.get())
```

```
    B7b = float(B7b_var.get())
```

```
    B6b = float(B6b_var.get())
```

```
    B8b = float(B8b_var.get())
```

```
    B10b = (B5b*numpy.tan(B1b * 6 * scipy.pi/180)-B7b*numpy.tan(B2b * 6 *  
scipy.pi/180)-B6b+B8b)/(numpy.tan(B1b * 6 * scipy.pi/180)-numpy.tan(B2b *  
6*scipy.pi/180))
```

```
    B10b_label = Label(f2, text='X(оп): ' + str(B10b), justify=CENTER)
```

```
    B10b_label.place(relx=.4, rely=.7)
```

```
    B11b = B6b+(B10b-B5b)*numpy.tan((B1b*6)*scipy.pi/180)
```

Продолжение Приложения А

```
B11b_label = Label(f2, text='Y(оп): ' + str(B11b), justify=CENTER)
B11b_label.place(relx=.4, rely=.8)
```

```
#####
```

```
btn1 = Button(f2, text="Подсчитать", command=count1)
btn1.place(relx=.8, rely=.4)
```

```
#####
```

```
# ввод данных 3
```

```
B2a_var = StringVar()
B2a_entry = Entry(f3, textvariable=B2a_var, justify=LEFT)
B2a_entry.place(relx=.3, y=30)
B2a_label = Label(f3, text='x', justify=RIGHT)
B2a_label.place(relx=.2, y=30)
```

```
B3a_var = StringVar()
B3a_entry = Entry(f3, textvariable=B3a_var, justify=LEFT)
B3a_entry.place(relx=.3, y=50)
B3a_label = Label(f3, text='y', justify=RIGHT)
B3a_label.place(relx=.2, y=50)
```

```
B6a_var = StringVar()
B6a_entry = Entry(f3, textvariable=B6a_var, justify=LEFT)
B6a_entry.place(relx=.3, y=70)
```

Продолжение Приложения А

```
B6a_label = Label(f3, text='d1', justify=RIGHT)
```

```
B6a_label.place(relx=.2, y=70)
```

```
B7a_var = StringVar()
```

```
B7a_entry = Entry(f3, textvariable=B7a_var, justify=LEFT)
```

```
B7a_entry.place(relx=.3, y=90)
```

```
B7a_label = Label(f3, text='α1', justify=RIGHT)
```

```
B7a_label.place(relx=.2, y=90)
```

```
B11a_var = StringVar()
```

```
B11a_entry = Entry(f3, textvariable=B11a_var, justify=LEFT)
```

```
B11a_entry.place(relx=.3, y=110)
```

```
B11a_label = Label(f3, text='d2', justify=RIGHT)
```

```
B11a_label.place(relx=.2, y=110)
```

```
B12a_var = StringVar()
```

```
B12a_entry = Entry(f3, textvariable=B12a_var, justify=LEFT)
```

```
B12a_entry.place(relx=.3, y=130)
```

```
B12a_label = Label(f3, text='α2', justify=RIGHT)
```

```
B12a_label.place(relx=.2, y=130)
```

```
#####
```

```
def count2():
```

```
    B2 = float(B2a_var.get())
```

```
    B3 = float(B3a_var.get())
```

Продолжение Приложения А

```
B6 = float(B6a_var.get())
B7 = float(B7a_var.get())
B11 = float(B11a_var.get())
B12 = float(B12a_var.get())
C7 = B7 * 6
C7_label = Label(f3, text='C7: '+str(C7), justify=CENTER)
C7_label.place(relx=.3, y=150)
C12 = B12*6
C12_label = Label(f3, text='C12: '+str(C12), justify=CENTER)
C12_label.place(relx=.3, y=170)
B16 = B2 + B6 * numpy.cos((B7 + 30) * 6 * scipy.pi / 180)+ B11 *
numpy.cos(C12 * scipy.pi / 180)
B17 = B3 + B6 * scipy.sin((B7 + 30) * 6 * scipy.pi / 180) + B11 *
scipy.sin(C12 * scipy.pi / 180)
B16_label = Label(f3, text='X: '+str(B16), justify=CENTER)
B16_label.place(relx=.3, y=190)
B17_label = Label(f3, text='Y: '+str(B17), justify=CENTER)
B17_label.place(relx=.3, y=210)

#####

btn2 = Button(f3, text="Подсчитать", command=count2)
btn2.place(relx=.8, rely=.4)

#####

root.mainloop() # запуск программы (главный цикл)
```