**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ ГОРОДА МОСКВЫ**

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ ИМЕНИ П.А. ОВЧИННИКОВА**

**Курсовая работа**

**на тему: «Создание приложение для контроля физической активности»**

**По дисциплине: «МДК 01.01 Разработка программных модулей»**

Выполнил студент: Магомедов Шамиль Расимович 2ИС

Преподаватель: Олейник Владимир Иванович

Подпись\_\_\_\_\_\_\_\_

Оценка\_\_\_\_\_\_\_ Дата\_\_\_\_\_\_\_\_\_\_\_

Москва

2023г

Цели курсовой работы:

Оглавление

[Глава 1 4](#_Toc151935605)

[Введение 4](#_Toc151935606)

[1.1 Преимущества физической активности 5](#_Toc151935607)

[1.2 Нужда физической активности в современности 6](#_Toc151935608)

[1.3 Немного о ИМТ тела 6](#_Toc151935609)

[1.4 Важное о УБМ 9](#_Toc151935610)

[1.5 Обзор литературы 10](#_Toc151935611)

[2. Постановка задачи 11](#_Toc151935612)

[3. Глава 1. Стадии и этапы разработки 12](#_Toc151935613)

[3.1 Анализ и разработка требований к приложению 12](#_Toc151935614)

[3.2 Планирование проекта 12](#_Toc151935615)

[3.3 Реализация продукта 13](#_Toc151935616)

[3.4 Тестирование программного продукта 13](#_Toc151935617)

[3.5 Написание Руководства по эксплуатации 14](#_Toc151935618)

[4.Идея создания приложения 14](#_Toc151935619)

[4.1 Язык программирования 15](#_Toc151935620)

[4.2 Особенности данного языка 17](#_Toc151935621)

[4.3 История появления языка 20](#_Toc151935622)

[4.4 Возможности Python 22](#_Toc151935623)

[Глава 2. Разработка приложения физической активности 26](#_Toc151935624)

[1.1 Алгоритм работы основного функционала приложения 26](#_Toc151935625)

[1.2 Использование программных библиотек 27](#_Toc151935626)

[1.3 Архитекутра приложения 27](#_Toc151935627)

[Заключение 28](#_Toc151935628)

[Список источников: 28](#_Toc151935629)

# Глава 1

# Введение

Физическая активность играет важную роль в нашей жизни. От раннего возраста наши тела нуждаются в регулярных физических упражнениях для поддержания здоровья и хорошей физической формы.

Одним из главных преимуществ физической активности является улучшение общего самочувствия человека. Регулярные тренировки помогают поддерживать хорошую физическую форму, укреплять мышцы и суставы, улучшают кровообращение, что в свою очередь способствует общему укреплению организма.

Необходимо отметить, что физическая активность должна быть разнообразной и регулярной. Рекомендуется заниматься спортом или другими видами активности минимум 150 минут в неделю. Это может быть ходьба, бег, плавание, йога или любая другая физическая деятельность, которая приносит удовольствие. ее роль в поддержании здоровья являются актуальными исследовательскими темами в современном обществе. В условиях сидячего образа жизни и увеличения количества хронических заболеваний, важно понимать, как физическая активность может быть использована для поддержания здоровья и профилактики различных заболеваний.

## Преимущества физической активности

Физическая активность также имеет положительное влияние на наше психическое здоровье. Во время тренировок вырабатывается эндорфин - гормон счастья, который помогает справляться с стрессом, улучшает настроение и повышает уровень энергии. Поэтому физическая активность часто рекомендуется людям с депрессией или тревожными расстройствами. Кроме того, физическая активность может быть связана с долголетием. Исследования показывают, что люди, ведущие активный образ жизни, имеют большие шансы дожить до старости и иметь лучшую жизненную позицию.

* Снижает риск заболевания сердечно-сосудистых заболеваний, инсульта, сахарного диабета 2 типа
* Помогает сохранять здоровый вес
* Улучшает самооценку
* Снижает симптомы депрессии и беспокойства

## Нужда физической активности в современности

Физические упражнения повышают устойчивость организма по отношению к действию целого ряда различных неблагоприятных факторов. Физические упражнения повышают иммунобиологические свойства крови и кожи, а также устойчивость к некоторым инфекционным заболеваниям.

При мышечной деятельности возникает влияние работающих мышц на внутренние органы. Это обеспечивает оптимальный уровень метаболизма (обмен веществ в организме), повышается активность ферментных систем, происходит более экономичное использование кислорода. Упражнения помогают улучшить контроль массы тела, прочность костной ткани (профилактика остеопороза), качество жизни в целом.

## 1.3 Немного о ИМТ тела

Кто и зачем придумал считать индекс массы тела (ИМТ)?

30 сентября 2020

Редкий человек, который хотя бы однажды не пытался рассчитать индекс массы тела. Все о нем знают, но не все понимают, зачем его вообще надо считать.

Индекс массы тела (ИМТ) - это величина, позволяющая определить, насколько вес человека соответствует его росту, имеется ли излишек веса, или наоборот, масса тела недостаточная.

Как посчитать ИМТ?

Нужно массу тела, то есть килограммы, поделить на рост.

ИМТ = вес (кг) / рост² (м)

Зачем знать ИМТ?

Его используют, чтобы выяснить риски по заболеваниям. Всемирная организация здравоохранения (ВОЗ) утверждает, что лишний вес увеличивает риск диабета второго типа, коронарной болезни сердца, рака груди, матки, кишечника, простаты, почек и желчного пузыря.

Как рассчитать ИМТ

Как рассчитать ИМТ

Кто придумал считать ИМТ?

Саму формулу вывел бельгийский астроном, математик и статист Адольф Кетле в 1832 году. Но известна она стала только через 140 лет, после исследования физиолога Анселя Кейса. Он проанализировал параметры 7 400 человек из пяти стран и сравнил разные формулы для определения лишнего веса. Метод расчета ИМТ как способ определения лишнего веса и ожирения показался ему самым простым и удобным.

Насколько точные прогнозы дает ИМТ?

Этот метод не всегда подходит отдельным людям. Потому что ИМТ не показывает реального процента жира и мышц. Например, у ИМТ мускулистого атлета может совпадать с индексом тучного нетренированного человека. Весить они будут одинаково, но при этом процент жира, внешний вид и риски для здоровья сильно отличаются. Также ИМТ не учитывает пол и возраст. Рамки индекса сделали универсальными, чтобы было удобнее проводить масштабные исследования. В то же время количество жира у женщин и мужчин различается в среднем на 10%.

Кроме того, пропорции мышечной и жировой ткани в организме изменяются. С возрастом замедляется метаболизм, начинается распад мышечной ткани и отложение жировой. Поэтому для правильных выводов необходимо брать во внимание и пол и возраст человека.

ИМТ не учитывает трёхмерность человека. Изменения роста и веса происходят нелинейно. Невысоким людям показывает, что они стройнее, чем есть, а высоких заставляет верить, что они толще.

Значение расчетов индекса массы тела (ИМТ)

Разные ученые предполагали свои версии формулы расчета ИМТ, но любая из них будет несовершенной, поскольку человек слишком сложен.

Есть ли альтернатива ИМТ?

Некоторые исследователи говорят, что большой информативностью обладают обхват талии или соотношение талии и бёдер. Дело в том, что жир вокруг печени и других органов брюшной полости (также его называют висцеральным) считается наиболее опасным. Он вырабатывает жирные кислоты, воспалительные агенты и гормоны, которые повышают уровень холестерина низкой плотности, глюкозы и триглицеридов в крови, увеличивают кровяное давление.

В любом случае, какие бы замеры вы не делали самостоятельно, лучшую их интерпретацию может дать только врач. Берегите себя и будьте здоровы!

## 1.4 Важное о УБМ

Базальная скорость метаболизма (BMR) — это скорость расхода энергии в единицу времени эндотермическими животными в состоянии покоя. Она выражается в единицах энергии в единицу времени в диапазоне от ватта (джоуль/сек) до млO2/мин или джоуля в час на кг массы тела Дж/(ч·кг). Правильное измерение требует соблюдения строгого набора критериев. Эти критерии включают в себя нахождение в физически и психологически ненарушенном состоянии и нахождение в термически нейтральной среде в постабсорбционном состоянии (т.е. без активного переваривания пищи). У животных, таких как рыбы и рептилии, применяется эквивалентный термин «стандартная скорость метаболизма» (SMR). Он следует тем же критериям, что и BMR, но требует документирования температуры, при которой была измерена скорость метаболизма. Это делает BMR вариантом стандартного измерения скорости метаболизма, который исключает данные о температуре, что привело к проблемам в определении «стандартных» скоростей метаболизма для многих млекопитающих.

Метаболизм включает в себя процессы, необходимые организму для функционирования. Базальная скорость метаболизма — это количество энергии в единицу времени, которое необходимо человеку для поддержания функционирования организма в состоянии покоя. Некоторые из этих процессов включают дыхание, кровообращение, контроль температуры тела, рост клеток, работу мозга и нервов, а также сокращение мышц. Базальная скорость метаболизма влияет на скорость, с которой человек сжигает калории, и, в конечном счете, на то, поддерживает ли этот человек вес, набирает или теряет его. Основной уровень метаболизма составляет от 60 до 75% ежедневного расхода калорий людьми. На это влияет несколько факторов. У людей BMR обычно снижается на 1-2% за десятилетие после 20 лет, в основном из-за потери безжировой массы, хотя вариабельность между людьми высока.

## 1.5 Обзор литературы

* Mifflin M.D. ,St Jeor S.T., Hill L.A., Scott B.J., Daugherty S.A., Koh Y.O.

A new predictive equation for resting energy expenditure in healthy individuals

,1990

В данной статье был проведён множественный регрессионный анализ,

в ходе которого было выявлено, что такие параметры как вес, рост, пол и

возраст оказывают наибольшее влияние на точность оценочной модели

скорости базового расхода калорий (RMR). В результате было разработано

прогностическое уравнение Миффлина-Сент-Джеора для расчёта скорости

базового расхода калорий (RMR) на основании данных признаков.

* Frankenfield D., Roth-Yousey L., Compher C. Comparison of predictive

equations for resting metabolic rate in healthy nonobese and obese adults: a

systematic review, 2005В данной работе был осуществлён сравнительный анализ наиболее

популярных уравнений для измерения скорости метаболизма в покое (RMR).

При исследовании точности на основе различных метрик уравнение

Миффлина-Сент-Джеора было выявлено как наиболее надежное,

прогнозирующее RMR с точностью в пределах 10% от измеренного у

большего числа людей, что является наилучшим результатом в сравнении с

другими решениями.

# 2. Постановка задачи

Целью курсовой работы является одним из немногих Desktop приложений связанных с физической активностью тела. Для достижения поставленной цели решаются следующие основные задачи

* Исследование и понятие расчета калорий при определенной активности человека
* Изучение таких понятий как УБМ, ИМТ тела
* Реализация алгоритмов по расчету калорий с учетом дополнительных параметров
* Разработка Desktop приложения, осуществляющего расчёт расхода калорий и хранение актуальной информации о физической активности пользователя

# 3. Глава 1. Стадии и этапы разработки

* Анализ и разработка требований к приложению
* Планирование проекта
* Реализация продукта
* Тестирование программного продукта
* Написание руководства по эксплуатации

## 3.1 Анализ и разработка требований к приложению

Анализ и разработка требований к приложению физической активности является важным этапом в разработке любого приложения. Вам потребуется определить основные цели и функциональность приложения, чтобы удовлетворить потребности пользователей. Программа будет включать различных видов расчеты такие как ИМТ тела и расчет калорий являются основными из них. Также основной задачей будет различного рода активности пользователя пользователь может выбирать под себя, все его основные тренировки будут записываться в БД. Разработчикам важно точно выявить желание заказчика программного обеспечения, а также необходимо тесное взаимодействие разработчика с заказчиком.

## 3.2 Планирование проекта

Планирование проекта - поможет вам определить все необходимые задачи, ресурсы, сроки и этапы, чтобы успешно достичь поставленных целей. Также непрерывный процесс определения наилучшего способа действий для достижения поставленных целей проекта с учетом складывающейся обстановки. Планирование является наиболее важным процессом управления проектом, определяющим во времени всю деятельность по осуществлению проекта.

## 3.3 Реализация продукта

это некий системный процесс или цепь взаимосвязанных процессов, который/которые состоят из определённых действий, привязанных ко времени и направленных на достижение заданного результата.

Сами по себе проекты могут быть частью других, более сложных, или, наоборот, разбиваться на более мелкие.

Цели не всегда бывают фиксированными, они могут требовать уточнения в процессе выполнения действий или после некоторых этапов. Внутри проекта могут быть промежуточные цели и т.д.

Результат любого проекта – конкретный продукт, одна его единица или множество. Вот только продукты не всегда могут быть материальными.

## 3.4 Тестирование программного продукта

Тестирование – процесс исследования, испытания программного продукта, проверка созданного программного обеспечения на соответствие заданным требованиям, и на отсутствие дефектов.

Чтобы программный продукт имел как можно меньше дефектов, этап тестирования стоит начинать как можно раньше, чтобы была возможность выявить проблемы уже на ранних стадиях разработки.

## 3.5 Написание Руководства по эксплуатации

Руководство по эксплуатации – это документ на бумажном носителе или электронное приложение, в котором отражается информация о параметрах и характеристиках изделия, способах его безопасного применения, критериях оценки его текущего состояния и определения необходимости отправки в ремонт, сведения о порядке утилизации.

# 4.Идея создания приложения

Приложение физической активности в наше время набирает все больше и больше популярности, кроме того они очень полезны в наше время так как поддержания физического здоровья очень сказывается на последующую жизнь. Дело в том что большинство приложений созданных для мониторинга здоровья это обычно мобильные приложения. А Desktop приложение поможет пользоваться или записывать данные или вводит расчет уже сделанной активности, не подключаясь при этом к телефону.Такое приложение может быть полезным как для начинающих спортсменов, так и для опытных спортсменов, которые хотят отслеживать свои достижения и поддерживать мотивацию. Для разработки потребовалось изучить интернет ресурсы и тщательно изучить анатомию человека для точного подсчета.

## 4.1 Язык программирования

Для данного Desktop приложения использовался наиболее распространённый язык программирования как “Python” который имеет популярность среди разработчиков.

Python (МФА: [ˈpʌɪθ(ə)n]; в русском языке встречаются названия пито́н или па́йтон) — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным в том плане, что всё является объектами. Необычной особенностью языка является выделение блоков кода отступами. Синтаксис ядра языка минималистичен, за счёт чего на практике редко возникает необходимость обращаться к документации. Сам же язык известен как интерпретируемый и используется в том числе для написания скриптов. Недостатками языка являются зачастую более низкая скорость работы и более высокое потребление памяти написанных на нём программ по сравнению с аналогичным кодом, написанным на компилируемых языках, таких как C или C++.

Python является мультипарадигменным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование функциональное программирование и асинхронное программирование . Задачи обобщённого программирования решаются за счёт динамической типизации. Аспектно-ориентированное программирование частично поддерживается через декораторы, более полноценная поддержка обеспечивается дополнительными фреймворками. Такие методики как контрактное и логическое программирование можно реализовать с помощью библиотек или расширений. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора (GIL), высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Эталонной реализацией Python является интерпретатор CPython, который поддерживает большинство активно используемых платформ и являющийся стандартом де-факто языка. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. CPython компилирует исходные тексты в высокоуровневый байт-код, который исполняется в стековой виртуальной машине. К другим трём основным реализациям языка относятся Jython (для JVM), IronPython (для CLR/.NET) и PyPy. PyPy написан на подмножестве языка Python (RPython) и разрабатывался как альтернатива CPython с целью повышения скорости исполнения программ, в том числе за счёт использования JIT-компиляции. Поддержка версии Python 2 закончилась в 2020 году. На текущий момент активно развивается версия языка Python 3. Разработка языка ведётся через предложения по расширению языка PEP (англ. Python Enhancement Proposal), в которых описываются нововведения, делаются корректировки согласно обратной связи от сообщества и документируются итоговые решения.

Стандартная библиотека включает большой набор полезных переносимых функций, начиная с возможностей для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на Си или C++, при этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках. Существует и специализированный репозиторий программного обеспечения, написанного на Python, — PyPI. Данный репозиторий предоставляет средства для простой установки пакетов в операционную систему и стал стандартом де-факто для Python. По состоянию на 2019 год в нём содержалось более 175 тысяч пакетов.

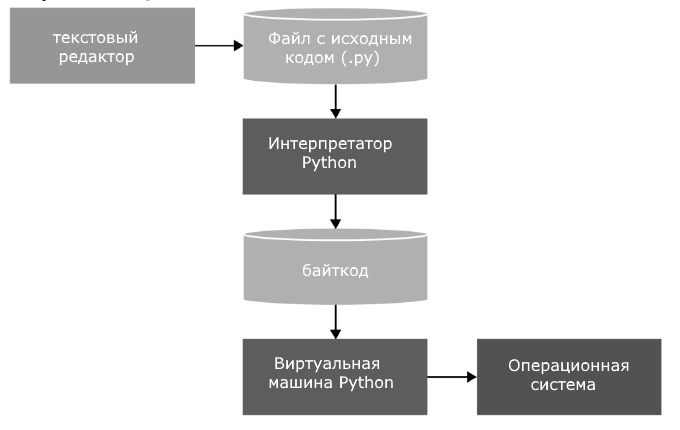
Python стал одним из самых популярных языков, он используется в анализе данных, машинном обучении, DevOps и веб-разработке, а также в других сферах, включая разработку игр. За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм. Отладка же и экспериментирование в значительной степени облегчаются тем фактом, что язык является интерпретируемым. Применяется язык многими крупными компаниями, такими как Google или Faceboo.

## 4.2 Особенности данного языка

Особенности языка python:

Скриптовый язык, код программы определяется в виде скриптов. Язык поддерживает различные парадигмы программирования: ООП и Функциональное.

Интерпретация программы, для работы со скриптами нам нужен интерпретатор, который будет запускать скрипты.



Портативность и кроссплатформенность, автоматическая управление памяти, Динамическая типизация (переменная в процессе выполнения кода разные значения).

Программа на python состоит из набора инструкций, каждая инструкция помещается на новую строку, большую роль в ЯП python играет табуляция. Некоторые конструкции языка могут состоять из нескольких строк. Регистрозависимость.

Чтобы пометить что делает тот или иной участок кода даётся комментарий и тогда строку интерпретатор не прочтёт, всего 2 вида комментариев: строчный (# перед строкой) и блочный ( ставятся три одинарные кавычки [‘’’] ).

В python есть 2 основные для консоли команды: print и input (отвечает за ввод информации с консоли, введённую строку мы можем сохранить как значение переменной)

Переменные и типы данных

Название переменной в python должно начинаться с алфавитного символа или со знака подчеркивания, название переменных не должны совпадать с ключами:

False, await, else, import, pass, None, break, except, in, raise, True, class, finally, is return, and, continue, for, lambda, try, as, def, from, nonlocal, while, assert, del, global, not, with, async, elif, if, or, yield

Переменные нужно называть 2 способами: camel case (подтип переменных начинается с большой буквы, userName = "Bob") и underscore notation (user\_name="Ann" - с подчёркиванием).

Типы данных:

bool - истинность, int - целочисленные, float - с плавающей точкой, complex - комплексные числа, str - строковые данные.

По умолчанию стандартные числа рассматриваются в десятичной системе, python поддерживает и другие СС, для указание чтобы указать что число находится в двоичной системе нужно поставить префикс “0b”, префикс для восьмеричной СС “0o”, для шестнадцатеричной нужно ставить префикс “0x” .

Строка - некая последовательность символов, заключенная в двойные или одинарные кавычки со стандарта 3.0 и выше, строки представлены в кодировке Unicode, если строка имеет много символов, то мы можем её разбить на части и разместить на разных строках кода. Чтобы так оформить надо заключить всю строку в круглые скобки, а подстроки в кавычки.

Поразрядные операции с числами и двоичное представление чисел.

Логические операции:

Логическое умножение (& - амперсанд).

Логические операции производятся над отдельными разрядами числа, происходит умножение поразрядно и если у обоих операндов значение

разрядов равна 1, то операция возвращает 1, в любом другом случае возвращается 0.

( | ) - Логическое сложение поразрядно и выводит 1, если хотя бы у одного числа проверяемого в разряде есть 1.

Логическое исключающее или - ( ^ ), иногда называют операцией XOR, если значение разрядов чисел разные, то выводят 1, иначе 0.(иногда применяют для простого шифрования).

Инверсия аналогична выражению” -(x+1)” и обозначается ( ~ ) и инвертирует наше число.

## 4.3 История появления языка

Задумка по реализации языка появилась в конце 1980-х годов, а разработка его реализации началась в 1989 году сотрудником голландского института CWI Гвидо ван Россумом. Для распределённой операционной системы Amoeba требовался расширяемый скриптовый язык, и Гвидо начал разрабатывать Python на досуге, позаимствовав некоторые наработки для языка ABC (Гвидо участвовал в разработке этого языка, ориентированного на обучение программированию). В феврале 1991 года Гвидо опубликовал исходный текст в группе новостей alt.sources. С самого начала Python проектировался как объектно-ориентированный язык.

Гвидо ван Россум назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона», поскольку автор был поклонником этого телешоу, как и многие другие разработчики того времени, а в самом шоу прослеживалась некая параллель с миром компьютерной техники.

Наличие дружелюбного, отзывчивого сообщества пользователей считается, наряду с дизайнерской интуицией Гвидо, одним из факторов успеха Python. Развитие языка происходит согласно чётко регламентированному процессу создания, обсуждения, отбора и реализации документов PEP (англ. Python Enhancement Proposal) — предложений по развитию Python.

3 декабря 2008 года, после длительного тестирования, вышла первая версия Python 3000 (или Python 3.0, также используется сокращение Py3k). В Python 3 устранены многие недостатки архитектуры с максимально возможным (но не полным) сохранением совместимости со старыми версиями Python.

Дата окончания срока поддержки Python 2.7 первоначально была установлена на 2015 год, а затем перенесена на 2020 год из опасения, что большая часть существующего кода не может быть легко перенесена на Python 3. Поддержка Python 2 была направлена лишь на уже существующие проекты, новые проекты должны были использовать Python 3. Официально Python 2.7 не поддерживается с 1 января 2020 года, хотя последнее обновление вышло в апреле 2020. Больше никаких исправлений безопасности или других улучшений для Python 2.7 не будет выпущено. С окончанием срока службы Python 2.x поддерживаются только Python 3.6.x и более поздние версии.

## 4.4 Возможности Python

Вот лишь некоторые вещи, которые умеет делать python:

* Работа с xml/html файлами
* Работа с http запросами
* GUI (графический интерфейс)
* Создание веб-сценариев
* Работа с FTP
* Работа с изображениями, аудио и видео файлами
* Робототехника
* Программирование математических и научных вычислений

Таким образом, python подходит для решения львиной доли повседневных задач, будь то резервное копирование, чтение электронной почты, либо же какая-нибудь игрушка. Язык программирования Python практически ничем не ограничен, поэтому также может использоваться в крупных проектах. К примеру, python интенсивно применяется IT-гигантами, такими как, например, Google и Yandex. К тому же простота и универсальность python делают его одним из лучших языков программирования.

Веб-разработка

Чаще всего Python используют в веб-разработке. Для него написано множество фреймворков: FastAPI, Flask, Tornado, Pyramid, TurboGears, CherryPy и, самый популярный, Django.

Ещё на Python пишут парсеры для сбора информации с веб-страниц. Делают это с помощью таких популярных библиотек:

* Scrapy — кросс-платформенная высокопроизводительная библиотека с большим сообществом пользователей.
* Beautiful Soup — простая и мощная библиотека для парсинга HTML- и XML-документов. Отличается хорошей документацией.
* Selenium — инструмент для автоматизации и тестирования онлайн-приложений, который позволяет имитировать работу браузера.

Научные исследования и машинное обучение

Благодаря лаконичности, простому синтаксису и богатому набору инструментов Python стал любимым языком учёных. Есть специальные библиотеки для Python, которые позволяют строить графики, проводить исследования и вычисления:

* SciPy — набор инструментов для научных вычислений.
* NumPy — расширение, которое позволяет оперировать матрицами и многомерными массивами.
* pandas — библиотека для аналитики данных.
* Matplotlib — библиотека для построения графиков различной сложности.

Также для Python есть множество свободных ML-инструментов:

* TensorFlow;
* scikit-learn;
* Keras;
* pandas.

Ещё одна область применения Python — автоматизация тестирования. Многие специалисты по автоматизации QA выбирают Python из-за его простоты. Он отлично подходит тем, кто имеет небольшой опыт в разработке приложений. Развитое сообщество, логичный синтаксис и удобочитаемость упрощают процесс обучения.

Хоть язык не компилируется, с его помощью всё же можно создать десктопные программы. Вот, например, небольшой список популярных приложений, которые написаны на Python:

* GIMP — визуальный редактор на Linux;
* Ubuntu Software Centre — центр приложений в ОС Ubuntu (один из дистрибутивов Linux);
* BitTorrent до шестой версии — менеджер торрент-закачек (позже программу переписали на C++, но сети peer-to-peer всё ещё работают на Python);
* Blender — программа для создания 3D-графики;
* Ansible — механизм управления конфигурацией, позволяющий развёртывать многоузловое программное обеспечение;
* Mercurial — кросс-платформенный инструмент управления распределённым исходным кодом.

Многие компьютерные игры полностью или частично написаны на Python. Существует заблуждение, что этот язык не подходит для серьёзных проектов, но на самом деле он использовался в разработке таких хитов, как:

* Battlefield 2;
* World of Tanks;
* Civilization IV;
* The Sims 4;
* EVE Online.

Главное преимущество Python — его простота. Не нужно тратить месяцы на изучение сложного синтаксиса, как в C++, и переживать об утечках памяти. Язык всё сделает за вас.

Другие достоинства Python:

Это интерпретируемый язык. До запуска программа на Python представляет собой обычный текстовый файл, который можно запустить на любой платформе, где установлен интерпретатор.

Он отлично подходит новичкам. Python хорошо спроектирован и логичен. Для его изучения даже не нужно знать английский язык. Благодаря его простоте разработка идёт намного быстрее, потому что программист пишет меньше кода.

У языка мощное сообщество. Сообщество пользователей Python настолько большое, что если вы столкнётесь с непонятной ошибкой, то, скорее всего, быстро найдёте её решение в интернете. Ведь до вас кто-то уже столкнулся с похожей и выложил решение на Stack Overflow.

# Глава 2. Разработка приложения физической активности

Создание приложения физической активности взяло начало с такого языка программирования как “Python”. В данной главе рассмотрим этапы и процессы, связанные с началом создания приложения для физической активности. Разработка приложений стала неотъемлемой частью нашей современной жизни. Они предоставляют удобный доступ к различным сервисам и возможности автоматизировать и улучшить многие аспекты нашей повседневной деятельности.

## Алгоритм работы основного функционала приложения

Программа состоит из расчета базовых введенных значений человека это рост, вес. Также имеется возможность записать имя и фамилию при желании. Далее идет расчет по формуле 0.9 \* вес (61кг) \* 0.1 (км). Тем самым можно вычесть сколько вы сожгли калорий при определенной активности. Есть в общем 4 вида активности с которыми может столкнуться каждый человек.

Физическими активностями считаются:

* Бег
* Езда на велосипеде
* Быстрая ходьба
* Плавание

Для каждой формулы предоставлен свой формат подсчета:

**activity == "Бег":**

**calories = 0.9 \* weight \* distance**

**activity == "Езда на велосипеде":**

**calories = 0.5 \* weight \* distance**

**activity == "Быстрая ходьба":**

**calories = 0.7 \* weight \* distance**

**activity == "Плавание":**

**calories = 0.6 \* weight \* distance**

## Использование программных библиотек

В языке «Python» есть некоторые библиотеки. Библиотека — это набор часто используемых кодов, которые разработчики могут включать в свои программы Python, чтобы не писать код с нуля. По умолчанию в Python доступна стандартная библиотека, которая содержит большое количество многократно используемых функций. Кроме того, доступно более 137 000 библиотек Python для различных задач, в числе которых интернет-разработка, наука о данных и машинное обучение (ML).

В моем же проекте Было использовано 7 видов различных библиотек для упрощенной работы с приложением. Вот некоторые из них:

* import sys
* import os
* import customtkinter
* import PIL

и др.

## Архитекутра приложения

В данном приложении испоьзуется монолитная архитектура создания. Монолитная архитектура — это традиционная модель разработки программного обеспечения, в которой одна база кода используется для выполнения нескольких бизнес-функций. Все программные компоненты монолитной системы взаимозависимы из-за использования встроенных механизмов обмена данными внутри системы. Модификация монолитной архитектуры возможна лишь частично и занимает много времени, поскольку даже небольшие изменения затрагивают большие области базы кода. Напротив, архитектурный подход на основе микросервисов подразумевает, что программное обеспечение состоит из небольших независимых компонентов (сервисов). Каждый сервис выполняет одну функцию и взаимодействует с другими сервисами через четко определенные интерфейсы. Поскольку они работают независимо, вы можете обновлять, изменять, развертывать или масштабировать каждую службу отдельно по мере необходимости.

# Заключение

В результате работы было создано Desktop приложение, отличающееся от аналогов внедрением дополнительных параметров учёта веса и роста при расчёте расхода калорий за пройденное расстояние, основанный на физической активности данного пользователя в указанный период времени. Эта возможность была реализована за счет применения иных формул расчёта интенсивности физической активности, в которых производится комбинирование значений веса и роста. Созданное приложение также предоставляет функциональность,позволяющую пользователю получить информацию о базовом расходе калорий, его текущем ИМТ тела и тд

# 

# Список источников:

<https://ru.wikipedia.org/wiki/Python>

<https://shop.actenzo.ru/articles/fizicheskaya-aktivnost-i-zdorove>

<https://nord-med.ru/news/news-company/kto-i-zachem-pridumal-schitat-indeks-massi-tela-imt/>

<https://en.wikipedia.org/wiki/Basal_metabolic_rate>

<https://www.python.org/>

<https://skillbox.ru/media/code/dlya_chego_nuzhen_python/>