Министерство образования Российской Федерации

Тверской государственный технический университет

Кафедра Программного обеспечения

Программа для работы с личными делами студентов

Выбор платформы и декомпозиция работ

Выполнил: Бакеев А.В.

Проверил: Мальков А.А

Тверь, 2024

Оглавление

[Введение 3](#_Toc181049493)

[Постановка задачи 3](#_Toc181049494)

[Пункт первый 4](#_Toc181049495)

[Пункт второй 4](#_Toc181049496)

[Пункт третий 5](#_Toc181049497)

[Пункт четвертый 5](#_Toc181049498)

# Введение

В современных условиях программирование является ключевым навыком для успешной реализации проектов. Выбор платформы для разработки программного обеспечения влияет на эффективность работы команды и качество продукта.

Данная лабораторная работа освещает процесс выбора платформы и декомпозиции задач, необходимых для успешного завершения проекта. Это может научить анализировать требования, подбирать подходящие технологии и распределять задачи, что способствует развитию навыков управления проектами и планирования разработки.

# Постановка задачи

**Вариант 2.** Разработать программный модуль «Личные дела студентов». Программный модуль предназначен для получения сведений о студентах сотрудниками деканата, профкома и отдела кадров. Сведения должны храниться в течение всего срока обучения студентов и использоваться при составлении справок и отчетов.

1. Составить перечень технологий, используемых для реализации варианта задания

2. Обосновать выбор данных технологий и удобство их использования

3. Декомпозировать разрабатываемую систему, описать модули, необходимые для реализации ПО согласно заданию.

4. Сформулировать набор задач, необходимых для реализации ПО согласно варианту задания, определить порядок выполнения и приоритет каждой из них.

# Пункт первый

Для реализации моего варианта я буду использовать следующий технологии:

Основная среда разработки – PyCharm (из под macOS)

Язык программирования – python (3.12)

Библиотека для реализации GUI (graphical user interface) – PyQt5 или tkinter

# Пункт второй

PyCharm является самой популярной и удобной средой разработки для создания проектов на языке программирования python. Это так же и место, содержащее множество мощных инструментов для написания и тестирования кода. Новые технологии-помощники сильно упрощают работу – предлагают где какой блок кода нужно дописать, при передаче в функцию каких-либо параметров подсвечивают серым, что есть что, дабы не перепутать. А удобный интерфейс оказывается весьма дружелюбным и понятным, чего нельзя сказать о некоторых других средах (например - VS Code, который так же подходит для задачи куда менее удобен, т.к. не является полноценной средой разработки, а лишь прокаченной версией редактора кода).

Python – простой и приятный синтаксически язык, способный легко и быстро решать множество задач, не требующих сверх быстрой работы кода или очень маленького пространства, занимаемого программой на жестком диске. Наш случай не является ни первым, ни вторым. К тому же на Пайтон существует множество библиотек, которые легко и удобно устанавливаются на любую версию интерпретатора, имеют хорошую совместимость друг с другом и достаточно полную документацию.

PyQt5 – библиотека для того, чтобы можно было работать на пайтоне с фреймворком Qt – достаточно популярным способ создания кроссплатформенных графических интерфейсов для десктопных приложений.

Tkinter – так же библиотека для написания несложных gui. Простая, не требовательная к мощностям машины, на которой будет запускаться программа, понятный и простой синтаксис команд.

# Пункт третий

Декомпозиция с описанием каждого модуля:

1. Продумать модуль хранения данных студента (какой тип данных у какого поля будет, каким способом будут хранить – с базой данных или нет, а если с базой данных – то какой)
2. Продумать функционал части сотрудника профкома – какие данные сможет получать о каждом студенте и в каком формате
3. Продумать функционал части сотрудника деканата– какие данные сможет получать о каждом студенте и в каком формате
4. Продумать функционал части сотрудника отдела кадров – какие данные сможет получать о каждом студенте и в каком формате
5. Так же продумать общую часть интерфейса – изначально программа будет одинаковая для всех – что со стороны университета, что со стороны деканата. Но в стартовом меню будет несколько вопросов, утоняющих, какой вид программы должен увидеть пользователь после ответа на них. Для стороны университета будут уникальные id, без которые нельзя будет авторизоваться в качестве сотрудника. Для студента же – паспортные данные.
   1. Id всегда целое число (не имеет значение знак, лишь отсутствие дробной части).
   2. Если id будет четным, значит это аккаунт студента
   3. Если id нечестный – это аккаунт сотрудника
   4. Стартовая форма просит ввести ваш id и на его основании определяет, какой интерфейс вы увидите дальше, после проверки.
   5. Студент – свои личные данные и возможность их редактировать
   6. Сотрудник – Список студентов и возможно получить данные о каждом конкретном из них

# Пункт четвертый

Набор необходимых задач:

1. Импортировать библиотеки для работы с GUI в среду разработки PyCharm
2. Сделать mvp\* версию приложения с хранением всех данных в файлах. У файла будет строго определенная система – у каждой строки свое предназначение
   1. Фамилия Имя Отчество
   2. Год рождения
   3. Средний балл за предыдущий семестр
   4. Статус о работе
3. Связать эту версию с интерфейсом и проработать систему id. Провести тесты, цель которых выявить ограничения функционала приложения, в зависимости от статуса пользователя (студент или сотрудник)
4. \*mvp подразумевает наличие меню, которое, по введенному id будет определять ваш уровень доступа (студент или сотрудник университета). В зависимости от этого уровня будет открываться разное меню приложения. В меню студента можно будет посмотреть свои текущие данные, а в меню сотрудника – список всех студентов.