**Отчёт по лабораторным работам**

**Вариант 1**

**Лабораторная работа 7.**

**Сетевые модели**

**Студент:**

**Адамович Карианна Павловна, 2-3**

**Преподаватель:**

**Бурмакова Анастасия Владимировна**

**г.Минск, 2025**

**Цель работы:** Приобретение навыков сетевого планирования и составления сетевых графиков, приобретение опыта нахождения критического пути.

**Задание для выполнения:**

Лабораторная работа базируется на исследовании различных тематик в проектировании программных продуктов, составлении сетевых графиков для разных тем, нахождении критических путей в составленных графиках. Каждый проект принять условным или обобщенным, но допустимо делать упор на конкретные примеры.

|  |  |  |
| --- | --- | --- |
| Вариант | Проект для исследования | Время выполнения всех задач |
| Вариант 1, 7, 13 | «Создание десктопного приложения» | 65 дней |

**Задание 1. Структурное планирование.**

Подумайте и выделите в проекте, согласно вашему варианту не менее 4 этапов работ. Также разбейте полученные этапы на задачи, их количество в совокупности по этапам должно быть не менее 12. Пример оформления задания смотрите в приложении ниже и в лекционном материале по теме.

**Задание 2. Календарное планирование.**

Распределите время, отпущенное на ваш проект согласно вариантам, на выделенные вами этапы. Скорректируйте сформулированные вами задачи, если это необходимо.

***Задания 1 и Задания 2 в отчете:***

***Тема «Создание десктопного приложения»***

|  |  |  |  |
| --- | --- | --- | --- |
| Код  операции | Наименование операции | Предшествующие операции | t |
| I. АНАЛИЗ | | | |
| Z1 | Сбор и анализ требований от пользователей |  | 5 |
| Z2 | Проектирование интерфейса приложения (UI/UX) | Z1 | 7 |
| Z3 | Создание технического задания и общей архитектуры приложения | Z1 | 5 |
| II. ПРОЕКТИРОВАНИЕ | | | |
| Z4 | Проектирование базы данных | Z2 | 10 |
| Z5 | Проектирование классов | Z3, Z4 | 15 |
| Z6 | Проектирование пользовательского интерфейса | Z2, Z5 | 10 |
| III. КОДИРОВАНИЕ | | | |
| Z7 | Кодирование интерфейсов пользователей | Z6 | 15 |
| Z8 | Кодирование процедур работы с базой данных | Z4, Z5 | 20 |
| Z9 | Кодирование классов | Z5 | 25 |
| IV. ТЕСТИРОВАНИЕ | | | |
| Z10 | Функциональное тестирование | Z7, Z8, Z9 | 15 |
| Z11 | Структурное тестирование | Z7, Z8, Z9 | 20 |
| V. ВНЕДРЕНИЕ | | | |
| Z12 | Разработка документации | Z10 | 5 |
| Z13 | Обучение пользователей | Z12 | 10 |
| Z14 | Испытание | Z11, Z13 | 15 |
| Z15 | Завершение работ | Z14 | 5 |

**Задание 3. Сетевой график, нахождение критического пути.**

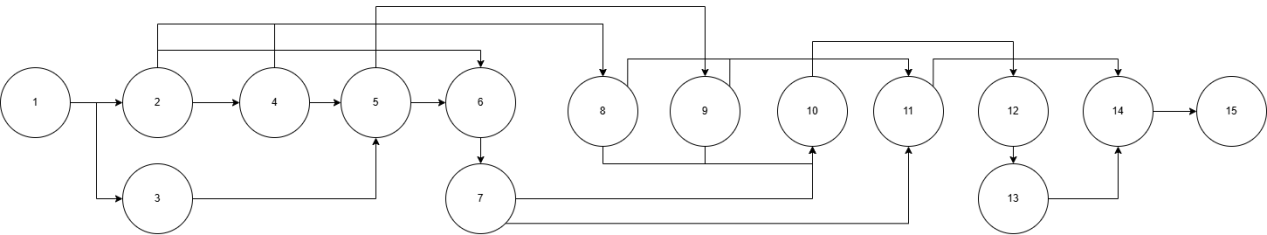
Согласно составленному перечню задач и распределённому времени составьте сетевой график вашего проекта. Помните о правилах составления графика и вводите фиктивные операции и операции ожидания если это необходимо.

При построении сетевых графиков соблюдается ряд правил:

1. в сети не должно быть событий (кроме исходного), в которые не входит ни одна дуга;
2. не должно быть событий (кроме завершающего), из которых не выходит ни одной дуги;
3. сеть не должна содержать замкнутых контуров (циклов);
4. ***любая пара событий сетевого графика может быть соединена не более чем одной дугой;***
5. ***номер*** начального ***события любой операции должен быть меньше номера ее*** конечного ***события.***

Найдите критический путь в составленном вами сетевом графике и обоснуйте его нахождение. Критический путь может быть меньше, чем время, отведенное на выполнение всех задач. Выделите, какие операции принадлежат критическому пути.

Z1 → Z2 → Z3 → Z4 → Z5 → Z6 → Z9 → Z10 → Z14 → Z15

******

**\*Задание 4. Оптимизация**

Предложите варианты оптимизации вашего проекта с привлечением денежных средств или человеческого ресурса.

Оптимизация проекта может быть достигнута за счет грамотного распределения ресурсов (человеческих и финансовых), внедрения эффективных инструментов и улучшения процессов разработки. Вот возможные варианты:

1. Оптимизация с привлечением денежных средств

Варианты:

**Инвестиции в инструменты разработки**:

Приобретение профессиональных средств для проектирования интерфейса (например, Figma, Adobe XD).

Покупка специализированных библиотек или фреймворков, ускоряющих разработку (например, PyQt для Python или WPF для C#).

Использование облачных решений для хранения данных и тестирования (Azure, AWS).

**Нанимайте внешних специалистов:**

Аутсорсинг отдельных задач, таких как проектирование базы данных или тестирование. Это поможет сократить сроки выполнения.

Нанять консультантов для улучшения качества кода и проектирования архитектуры.

**Автоматизация процессов:**

Внедрение CI/CD (Continuous Integration/Continuous Deployment) через такие платформы, как GitHub Actions или Jenkins, чтобы быстрее и качественнее выполнять сборки приложения.

Преимущества:

Снижение времени выполнения задач.

Повышение качества конечного продукта.

Снижение нагрузки на команду.

2. Оптимизация с привлечением человеческого ресурса

Варианты:

**Расширение команды:**

Привлечение специалистов по тестированию для оперативного поиска и исправления ошибок.

Найм UI/UX-дизайнеров для улучшения интерфейсов и пользовательского опыта.

**Обучение команды:**

Проведение тренингов по использованию новых технологий (например, инструментов для проектирования, работы с базами данных или оптимизации процессов разработки).

Семинары по повышению эффективности работы команды, включая методики Agile или Scrum.

**Увеличение гибкости работы:**

Разделение задач между командами для параллельного выполнения этапов. Например, разработка интерфейса и базы данных могут идти одновременно.

Введение роли координатора проекта, чтобы ускорить коммуникацию между командами.

Преимущества:

Эффективное распределение нагрузки.

Ускорение выполнения задач благодаря параллелизации.

Улучшение взаимодействия между разработчиками.

3. Внедрение современных технологий

Варианты:

**Использование готовых решений:**

Интеграция open-source библиотек для ускорения разработки (например, SQLite для базы данных).

Применение шаблонов интерфейсов вместо ручной реализации.

**Оптимизация тестирования:**

Автоматизация тестов с помощью таких инструментов, как Selenium для пользовательских интерфейсов и PyTest для Python.

Использование сервисов облачного тестирования (BrowserStack, Sauce Labs).

**Сокращение времени на документацию:**

Использование инструментов для автогенерации документации (например, Sphinx для Python или JSDoc для JavaScript).