*(1И, Половинкин, И/З №1, 16 вариант)*

**Задание 1. Разработка технического задания.**

**1.**

**a.** Цель: Разработать приложение, которое позволит пользователям находить друг друга на основе их общих интересов, создавая удобное пространство для общения и обмена опытом.

Задачи:

* Создание пользовательского интерфейса для регистрации, создания профиля и поиска других пользователей.
* Разработка алгоритма сопоставления пользователей на основе их интересов.
* Интеграция функционала общения, такого как чаты или форумы.
* Обеспечение безопасности данных пользователей.
* Оптимизация приложения для мобильных устройств.

**b. Программные продукты, решающие задачу:**

* Tinder
* Bumble
* OkCupid
* Meetup

Данные программные продукты были выбраны именно потому, что они успешно решают поставленную задачу и предоставляют ряд функций и особенностей, которые могут быть полезны и вдохновляющими для нашего проекта.

**c. Набор характеристик, влияющих на качество решения задачи:**

* Удобство использования интерфейса
* Эффективность алгоритма сопоставления пользователей
* Безопасность данных
* Наличие дополнительных функций (например, фильтры по интересам, геолокация и т. д.)
* Поддержка мобильных устройств

**d. Сравнительный анализ программных средств:**

| **Характеристика** | **Tinder** | **Bumble** | **OkCupid** | **Meetup** |
| --- | --- | --- | --- | --- |
| Удобство интерфейса | + | ++ | ++ | ++ |
| Эффективность алгоритма сопоставления | ++ | ++ | ++ | + |
| Безопасность данных | ++ | ++ | ++ | ++ |
| Дополнительные функции | ++ | ++ | ++ | ++ |
| Поддержка мобильных устройств | ++ | ++ | ++ | ++ |

**e. Характеристики предполагаемого программного продукта:**

* Интуитивно понятный и привлекательный пользовательский интерфейс.
* Алгоритм сопоставления, учитывающий как общие интересы, так и геолокацию пользователей.
* Защита данных пользователей с использованием современных методов шифрования.
* Дополнительные функции, такие как возможность создания событий и групп по интересам.
* Поддержка как мобильных устройств, так и веб-версии приложения.

**2.**

**a.** **Жизненный цикл бизнес-процесса, автоматизируемого программным продуктом:**

**1. Инициация:**

* **Цель:** Определение потребностей пользователей в знакомствах и общении, сбор информации о конкурентах.
* **Деятельность:**
  + Проведение анализа потребностей пользователей.
  + Исследование конкурентов.

**2. Планирование:**

* **Цель:** Определение ресурсов и плана развития приложения.
* **Деятельность:**
  + Проектирование интерфейса.
  + Разработка функций поиска.
  + Реализация механизмов общения.
  + Создание плана тестирования и отладки.

**3. Разработка:**

* **Цель:** Создание программного продукта, соответствующего требованиям.
* **Деятельность:**
  + Реализация проектированного интерфейса.
  + Реализация алгоритмов поиска.
  + Реализация системы обмена сообщениями.
  + Проведение тестирования и отладки.

**4. Внедрение:**

* **Цель:** Запуск приложения и привлечение пользователей.
* **Деятельность:**
  + Развертывание приложения.
  + Проведение маркетинговой кампании.

**5. Эксплуатация:**

* **Цель:** Обеспечение бесперебойной работы приложения и поддержка пользователей.
* **Деятельность:**
  + Проведение технической поддержки.
  + Обработка обратной связи от пользователей.

**6.** **Оптимизация:**

* **Цель:** Улучшение приложения на основе обратной связи от пользователей и изменений в отрасли.
* **Деятельность:**
  + Анализ обратной связи от пользователей.
  + Внедрение новых функций.
  + Обновление приложения.

**b. IDEF0 процесса:** ****



Анализ потребностей пользователей



Проектирование интерфейса



Разработка функц. поиска

****

Реализация механизмов общения общени





Тестирование и отладка



Развертывание и маркетинг

Поддержка и обновление







**c. Выявление пользователей системы:**  
Пользователи приложения для знакомств и общения по интересам могут быть разделены на следующие категории:

* **Конечные пользователи:** Люди, которые используют приложение для поиска новых знакомств и общения по интересам.
* **Администраторы:** Модераторы, которые отвечают за поддержание порядка и безопасности в приложении, а также за управление контентом.
* **Рекламодатели:** Лица, размещающие рекламу в приложении для монетизации.
* **Разработчики:** Люди, занимающиеся созданием и поддержкой технической части приложения.

**3.**

1. **Общие сведения**:
   * Название проекта: Приложение для знакомств и общения по интересам.
   * Назначение: Создание средства для пользователей, позволяющего находить людей с общими интересами и обмениваться сообщениями.
   * Владелец: [ItSight].
   * Стадия разработки: Первая версия.
2. **Цели и назначение создания автоматизированной системы**:
   * Создание средства обеспечения коммуникации между пользователями с общими интересами.
   * Обеспечение удобства поиска и нахождения потенциальных собеседников.
   * Предоставление безопасной и привлекательной среды для общения.
3. **Характеристика объектов автоматизации**:
   * Пользовательские профили.
   * Механизм поиска по интересам.
   * Система обмена сообщениями и чатов.
   * Интерфейс приложения на различных платформах.
4. **Требования к автоматизированной системе**:

**Функциональные требования:**

* + Регистрация и создание пользовательского профиля.
  + Возможность поиска других пользователей по интересам и параметрам.
  + Обмен сообщениями между пользователями.
  + Механизм управления настройками конфиденциальности.
  + Интерфейс, обеспечивающий удобство использования.

**Нефункциональные требования:**

* + Безопасность: защита данных пользователей, авторизация и аутентификация.
  + Производительность: оптимизация скорости работы системы при большом количестве пользователей и сообщений.
  + Совместимость: поддержка различных операционных систем и устройств.
  + Надежность: минимизация ошибок и сбоев, обеспечение доступности сервиса.
  + Масштабируемость: способность системы масштабироваться для обработки роста числа пользователей.
  + Удобство использования: интуитивный интерфейс, понятная навигация.

1. **Состав и содержание работ по созданию автоматизированной системы**:

* Анализ потребностей и формирование требований.
* Проектирование системы, включая архитектуру, базу данных и интерфейс.
* Разработка функциональности, включая фронтенд и бэкенд.
* Тестирование на всех этапах разработки: модульное, интеграционное, системное.
* Развертывание системы на серверах и мобильных платформах.
* Маркетинговые мероприятия для привлечения пользователей.
* Поддержка и обновление системы после выпуска.

1. **Порядок разработки автоматизированной системы**:
   * Анализ требований и определение приоритетов.
   * Проектирование интерфейса и базы данных.
   * Разработка поэтапно с учетом приоритетов и рисков.
   * Тестирование каждого этапа перед переходом к следующему.
   * Развертывание и настройка на тестовых и рабочих серверах.
   * Проведение заключительного тестирования перед выпуском.
2. **Порядок контроля и приемки автоматизированной системы**:

* Планирование и проведение тестирования на всех этапах разработки.
* Участие заказчика в проверке выполненных этапов и принятии решений.
* Фиксация ошибок и их исправление в течение разработки.
* Финальное тестирование перед релизом, включая нагрузочное и безопасности.
* Проведение приемочных испытаний заказчиком и подписание акта приемки.

1. **Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу автоматизированной системы в действие**:
   * Разработка отчетов о выполненных работах на каждом этапе разработки, включая:
     + Описание проделанных действий и достигнутых результатов.
     + Отчеты о проведенных тестах с указанием результатов.
     + Документация по проведенным изменениям в системе.
   * Подготовка документации для внутреннего использования командой разработчиков и технической поддержки, включая:
     + Описания архитектуры системы.
     + Инструкции по развертыванию и настройке.
     + Руководства пользователя.
   * Обеспечение полной трассируемости изменений в системе для последующего анализа и поддержки.
2. **Требования к документированию**:
   * Составление полной технической документации по процессу разработки и функциональности системы, включая:
     + Техническое задание.
     + Спецификации функциональных и нефункциональных требований.
     + Диаграммы базы данных и архитектуры системы.
     + Отчеты о тестировании и результаты испытаний.
   * Регулярное обновление документации в соответствии с изменениями в системе и ее окружении.
3. **Источники разработки**:
   * Систематическое использование следующих источников для обеспечения качества и надежности разработки:
     + Стандарты и регламенты, установленные заказчиком или соответствующими органами.
     + Результаты анализов и исследований, проведенных командой разработчиков.
     + Требования, выявленные в процессе взаимодействия с конечными пользователями и тестировщиками.
     + Опыт и лучшие практики, актуальные для сферы разработки приложений для знакомств и общения.
     + Обратная связь от пользователей и заказчика, направленная на постоянное улучшение продукта.
     + Данные о безопасности и конфиденциальности, актуальные для сферы социальных сетей и коммуникации онлайн.

**Задание 2. Разработка программного продукта**

**1. Постановка задачи**

**Условие задачи:**

Разработать программу для сортировки индексов автомобилей по цене и покупки автомобилей с учётом имеющегося бюджета.

**1.1. Допущения:**

* Предполагается, что список цен на автомобили и бюджет для покупки передаются программе в виде вектора целых чисел.
* Все цены и бюджет представлены положительными целыми числами.
* Программа должна быть способна корректно обрабатывать ситуации, когда список автомобилей пуст или бюджет недостаточен для покупки хоть одного автомобиля.

**1.2. Пример работы ("черный ящик"):**

Пример 1:

* Входные данные:
  + Цены на автомобили: [15, 5, 11, 10, 12]
  + Бюджет: 30
* Ожидаемый результат:
  + Количество купленных автомобилей: 3
  + Индексы купленных автомобилей: [1, 3, 2]

Пример 2:

* Входные данные:
  + Цены на автомобили: [5, 10, 1, 2, 1, 20]
  + Бюджет: 18
* Ожидаемый результат:
  + Количество купленных автомобилей: 4
  + Индексы купленных автомобилей: [0, 2, 3, 4]

**2. Декомпозиция**

**2.1. Диаграммы IDEF0:**

****

****

**Ввод данных**

****

****

****

****

**Расчёт бюджета**

****

****

****

**Выбор автомобилей**

****

****

****

**Вывод результатов**

****

**2.2. Спецификации модулей:**

* **Модуль сортировки индексов sortIndexes**:
  + **Вход**: Число N (количество автомобилей), вектор цен на автомобили.
  + **Выход**: Отсортированный вектор индексов автомобилей по возрастанию цены.
  + **Архитектура:** Алгоритм сортировки, основанный на стандартной функции **std::sort.**
  + **Обоснование:** **std::sort** - это высокоэффективный и оптимизированный алгоритм сортировки, доступный в стандартной библиотеке C++. В данном случае удобнее использовать готовый алгоритм библиотеки, чем реализовывать свой с нуля.
* **Модуль покупки автомобилей buyCars**:
  + **Вход**: Число N (количество автомобилей), бюджет S, вектор цен на автомобили.
  + **Выход**: Пара из количества купленных автомобилей и вектора индексов купленных автомобилей.
  + **Архитектура**: Функция buyCars покупает максимальное количество автомобилей с учётом бюджета и возвращает количество купленных автомобилей и их индексы. (т.е, использование жадного алгоритма).
  + **Обоснование:** Жадный алгоритм в данной задаче является оптимальным, поскольку он позволяет приобрести максимальное количество автомобилей с заданным бюджетом.

**2.3. Тесты ("черный ящик"):**

**Тесты для sortIndexes**

1. **Проверка сортировки в порядке возрастания:**
   * Вход: [15, 5, 11, 10, 12]
   * Ожидаемый выход: [1, 3, 4, 2, 0] (индексы отсортированы по возрастанию цен)
2. **Проверка сортировки в случае дубликатов:**
   * Вход: [5, 10, 1, 2, 1, 20]
   * Ожидаемый выход: [2, 3, 4, 0, 1, 5] (индексы отсортированы по возрастанию цен, при равных ценах сохранен исходный порядок)

**Тесты для buyCars**

1. **Проверка покупки автомобилей для примеров из условия задачи:**
   * Вход: N = 5, S = 30, A = [15, 5, 11, 10, 12]
   * Ожидаемый выход: Количество купленных автомобилей: 3, Номера купленных автомобилей: 2 3 4
   * Вход: N = 6, S = 18, A = [5, 10, 1, 2, 1, 20]
   * Ожидаемый выход: Количество купленных автомобилей: 4, Номера купленных автомобилей: 1 3 4 5
2. **Дополнительные тесты:**
   * Вход: N = 0, S = 100, A = [] (пустой список автомобилей)
   * Ожидаемый выход: Количество купленных автомобилей: 0, Номера купленных автомобилей: []
   * Вход: N = 5, S = 1, A = [10, 10, 10, 10, 10] (недостаточный бюджет)
   * Ожидаемый выход: Количество купленных автомобилей: 0, Номера купленных автомобилей: []
   * Вход: N = 5, S = -10, A = [1, 2, 3, 4, 5] (отрицательный бюджет)
   * Ожидаемый выход: Количество купленных автомобилей: 0, Номера купленных автомобилей: []
   * Вход: N = 5, S = 10, A = [-1, -2, -3, -4, -5] (отрицательные цены)
   * Ожидаемый выход: “Ошибка ввода: цена автомобиля не может быть отрицательной”.

**3. Алгоритмы**

**3.1. Основной алгоритм:**

1. Сортировка индексов автомобилей по цене с помощью функции sortIndexes.
2. Покупка автомобилей с использованием отсортированных индексов и учётом бюджета с помощью функции buyCars.

**3.2. Алгоритмы модулей (псевдокод):**

// Модуль 1: sortIndexes

алг sortIndexes

арг N, A

рез indexes

нач

indexes = [0, 1, 2, ..., N-1]

Сортировать indexes по возрастанию A[i] для каждого элемента i в indexes.

Вернуть indexes

кон

// Модуль 2: buyCars

алг buyCars

арг N, S, A

рез boughtCars, boughtIndexes

нач

indexes = sortIndexes(N, A)

boughtCars = 0

boughtIndexes = []

для каждого индекса i в indexes:

если A[i] <= S:

S = S - A[i]

boughtCars = boughtCars + 1

добавить i в boughtIndexes

иначе:

прервать цикл

Вернуть boughtCars, boughtIndexes

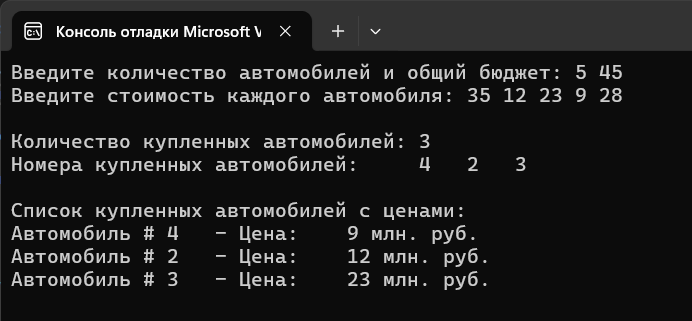
кон

**4. Разработка модулей**

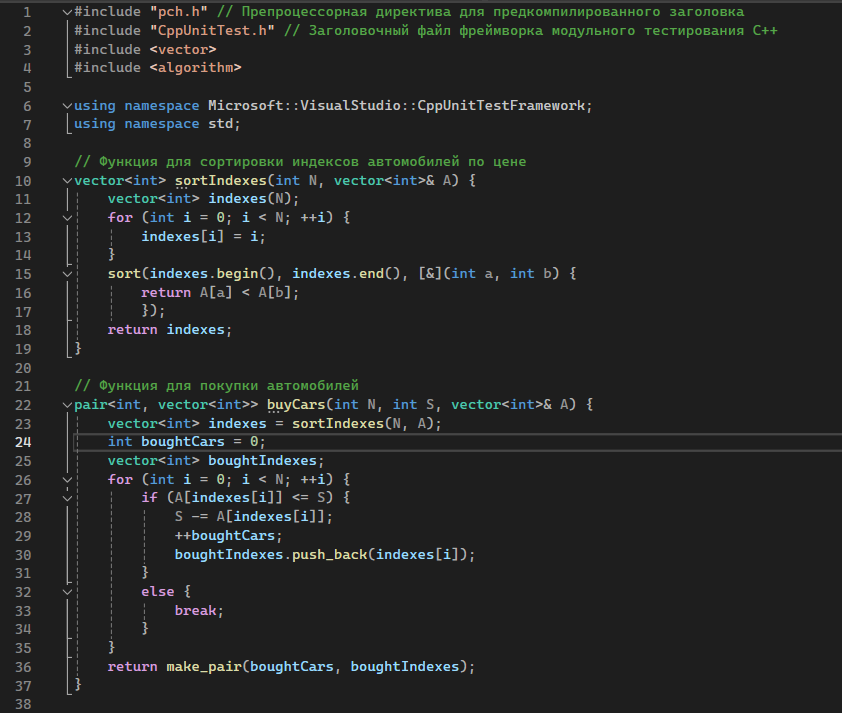
**4.1.** **Кодирование модулей с соответствующей документацией**

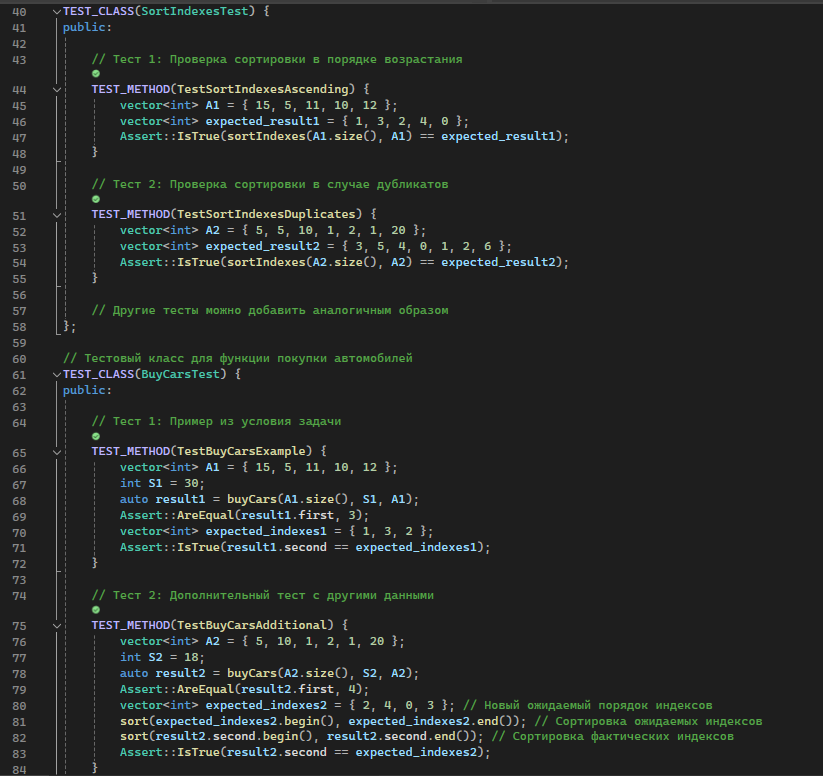
****

****

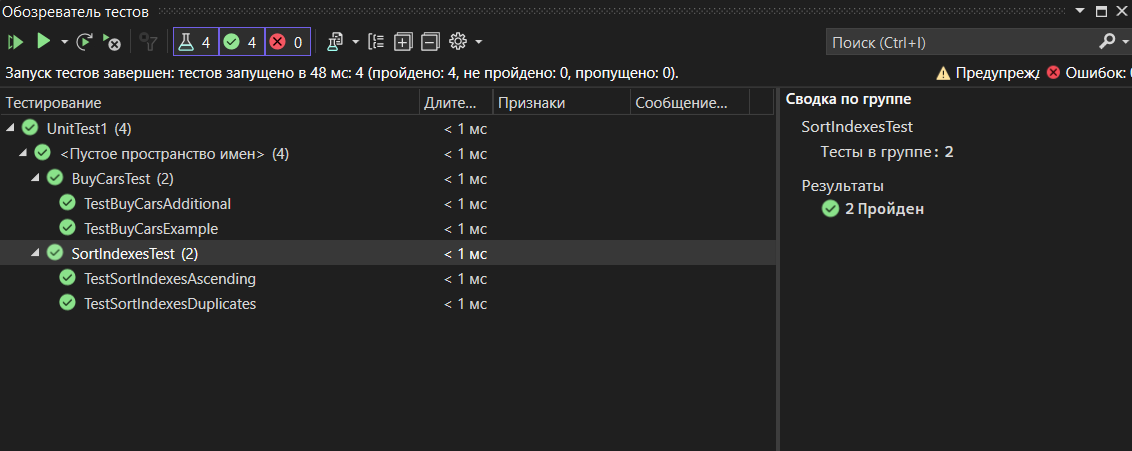
****

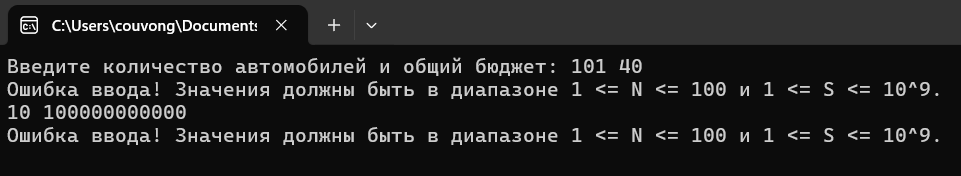
**4.2.**

****

****

**4.3.**

****

****