МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное
учреждение высшего образования «Южный федеральный университет»
(ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ)

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ ДЛЯ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ

Программное и аппаратное обеспечение информационных систем

Таганрог 2021

Лабораторная работа № 1.
Применение CASE-средств для проектирования и разработки программного и аппаратного обеспечения ИС и АС.

Цель занятия:
Изучить виды и типы CASE-средств поддержки процесса разработки программного обеспечения, освоить принципы применения CASE-средств на практике, выбрать тему и реализовать первую часть индивидуального проекта в рамках лабораторной работы.

В результате выполнения лабораторной работы студент получает комплект документации, схемы программного обеспечения, выполненные при помощи CASE-средств.

Средства выполнения практической работы:
Проведение занятий предполагается в аудитории, оснащенной:

1. Комплектом персональных ЭВМ в компьютерных классах с выходом в глобальную сеть Интернет.
2. Мультимедийной доской для демонстрации материалов.
3. Установленной ОС Windows или ОС Linux.
4. Установленным текстовым редактором MS Office или Libre Office.
5. Установленным редактором графических схем типа Draw.io.

Результаты освоения материала:
В результате проведения занятия студент должен:
Знать:

- классификацию CASE-средств и основы их применения
- современные инструменты поддержки процесса разработки программного обеспечения, относящиеся к классу CASE-средств

Уметь:

- применять на практике CASE-средства поддержки процесса разработки программного обеспечения

Методика проведения занятия:
Практическое занятие проводится следующим образом:

1. Рассмотрение теоретических вопросов.
2. Разъяснения преподавателя.
3. Диспут или «круглый стол» по тематике занятия.
4. Выполнение заданий преподавателя.

5. Анализ и оценка выполненных работ и степени овладения, обучающихся запланированными компетенциями.

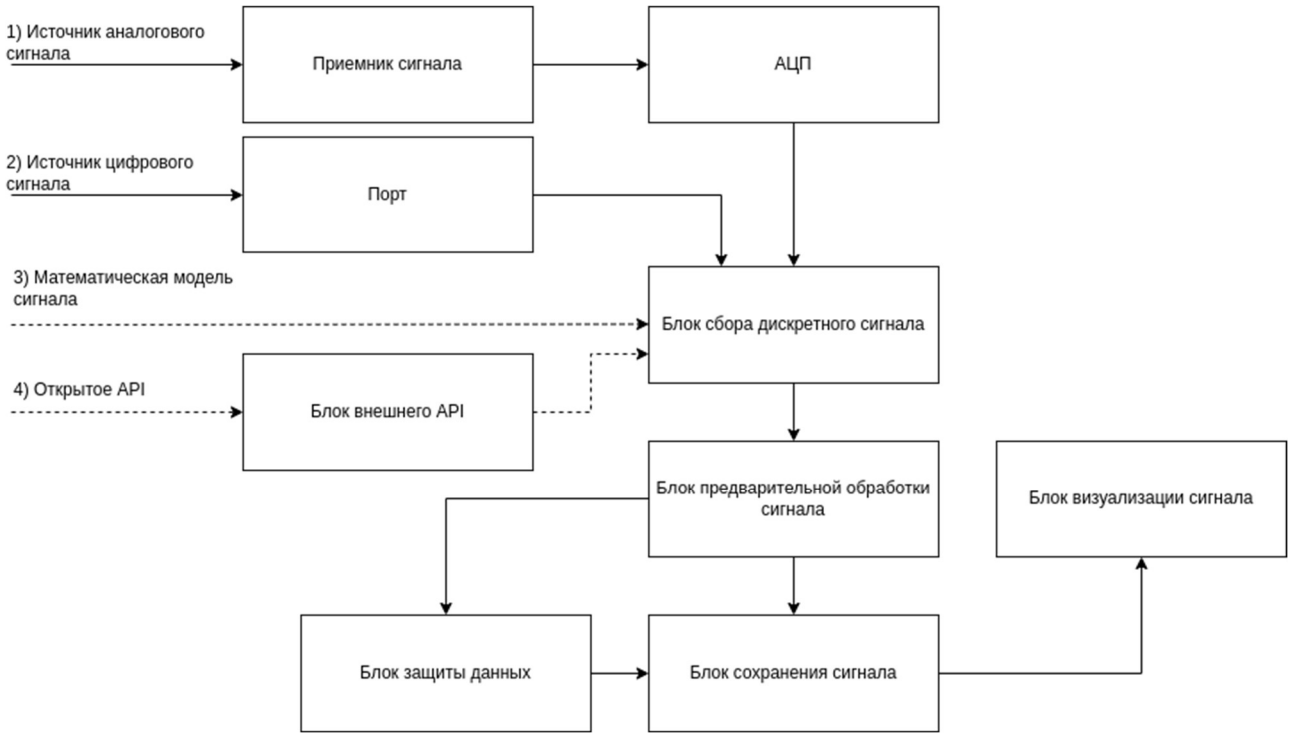
Содержание практического занятия:
В рамках практического занятия подробно рассматриваются и обсуждаются следующие ключевые вопросы

1. Основные группы CASE средств.
2. CASE средства верхнего уровня.
3. CASE средства нижнего уровня.
4. Интегрированные CASE средства.
5. Факторы, которые нужно принимать во внимание при выборе CASE-средств.
6. Методологии разработки программных проектов.
7. CASE-средства для проектирования программного обеспечения.
8. CASE-средства проектирования баз данных.
9. CASE-средства управления проектами и ролями пользователей.
10. CASE-средства моделирования.
11. Инструменты документирования проектов.
12. Основы применения UML-диаграмм.
13. Инструменты для построения UML-диаграмм.

В общем случае, каждому студенту предлагается разработать систему приема данных от аппаратного источника данных с предварительной обработкой, сохранением в базу данных и защитой информации.

Это общее задание, а частные случаи варьируются относительно варианта, а также желания и возможностей студентов.

Схематично такую систему можно изобразить следующим образом.



Студенту формируется следующее задание:

1. Формирование идеи для серии Лабораторных работ № 1–4 в виде комплексного приложения или аппаратно-программной платформы.
2. Выделение основных технических требований к разрабатываемому Приложений.
3. Оформление идеи и технических требований при помощи системы облачного документооборота (формирование комплекта документации).
4. Разработка структурной схемы Приложения с применением CASE-средств.
5. Разработка функциональной схемы Приложения с применением CASE-средств.
6. Формирование структуры модулей Приложения с применением CASE-средств (UML-диаграмм).
7. Формирование структуры базы данных Приложения с применением CASE-средств (UML-диаграмм).
8. Формирование этапов разработки Приложения и составление списка решаемых задач.
9. Оформление списка задач при помощи систем Управления проектами.

Для облачного документооборота предлагается использовать OneDrive или Google Drive, но выбор может быть изменен со стороны студента.

Для составления UML-диаграмм предлагается использовать бесплатный инструмент Draw.io, который существует в виде отдельного приложения, в виде онлайн-ресурса и даже в виде расширения для Google Drive.

Для оформления структуры базы данных предлагается использовать бесплатный ресурс dbdiagram.io или StarUML.

Для Управления проектами предлагается использовать бесплатный вариант JIRA или его аналоги типа Trello.

В идеальном случае, для получения сигналов и их предварительной обработки предполагается использовать аппаратные комплекты разработчиков типа:

1. Arduino.
2. Raspberry Pi.
3. STM SDK.

Но, так как зачастую у студентов отсутствует подобная плата комплекта разработчика или финансовые возможности ее приобретения или, чаще всего, знания в программировании аппаратуры и в области цифровой схемотехники, то студентам предлагается эмулировать процесс получения входных данных одним из следующих способов:

1. Математически.
2. При помощи файлов или базы данных с отсчетами или измерениями.
3. При помощи внешних сетевых API (открытых источников).

В первом случае, для генерации входных сигналов предлагается использовать аддитивную модель системы гармонических сигналов, которые можно генерировать на основе различных амплитуд и частот следующим образом:

X = A1*sin(2*pi*F1) + A2*sin(2*pi*F2) + … + Ak*sin(2*pi*Fk).

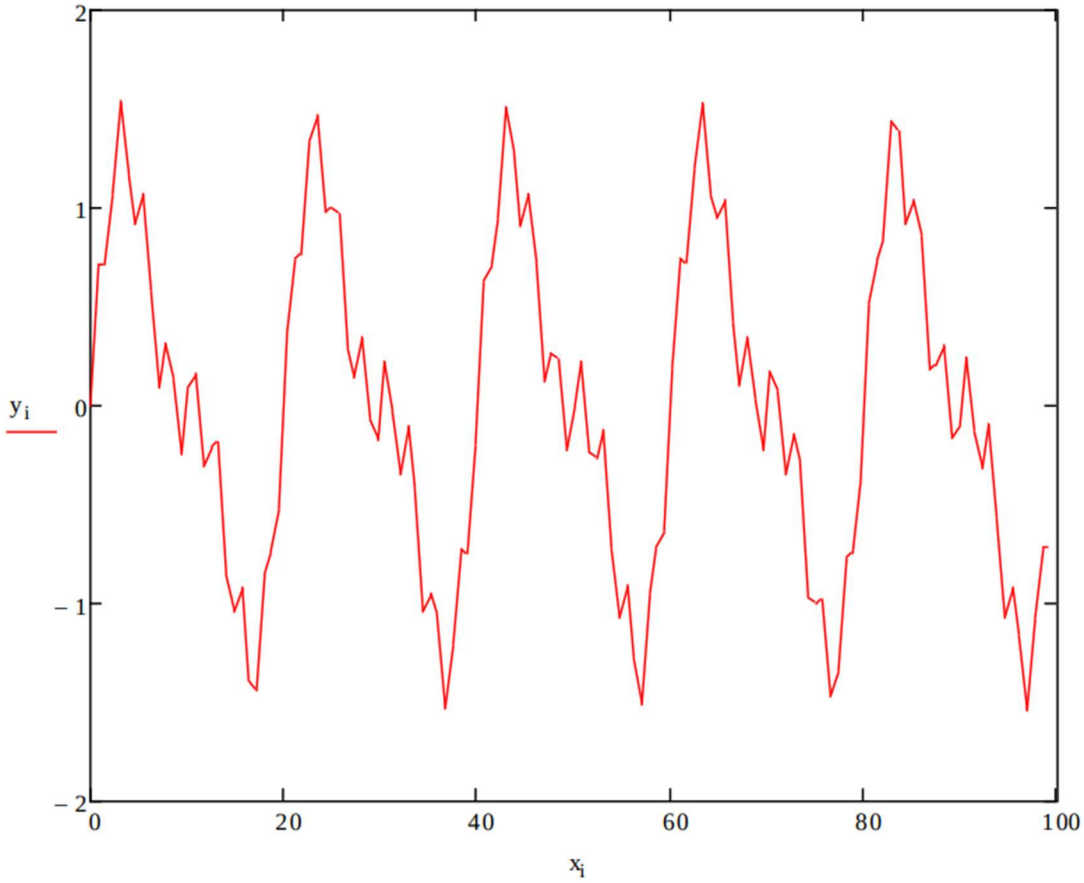
Здесь присутствует вектор амплитуд гармоник A и вектор частот гармоник F. Длины этих векторов одинаковы и равны k.

В результате сложения этих гармоник получается наложение сигналов разной силы и частоты, что позволяет имитировать источник случайных сигналов и получать разные формы.

Например, при следующей комбинации:

$yi \coloneqq \sin 2\pi\ 0{,}05xi + 0{,}5 \sin 2\pi\ 0{,}1xi + 0{,}25 \sin 2\pi\ 0{,}4xi$,

получается достаточно «чистый» сигнал



Имея такие дискретные данные, мы получаем симуляцию внешнего источника данных и процесса дискретизации.

На выходе этого процесса у нас есть массивы готовых цифровых данных, которые можно передавать на следующие этапы работы системы и использовать для выполнения задания к лабораторной работе.

Критерии оценивания:

Максимальное количество баллов за практическую работу: 10 баллов.
9-10 баллов– обучающийся в течение модуля активно принимает участие в устных опросах, даваемые им ответы верны и позволяют высоко оценить уровень владения материалом;

7-8 баллов – обучающийся принимает участие в устных опросах, даваемые им ответы, как правило, верны и позволяют оценить владение материалом на достаточном уровне;

6 баллов – обучающийся не проявляет инициативы для участия в устных опросах, а даваемые им ответы фрагментарны и верны лишь частично, что не позволяет оценить владение материалом на достаточном уровне;

1-5 баллов – обучающийся пытается участвовать в собеседовании, но дает неверные ответы, показывает отсутствие базовых знаний;

0 баллов – обучающийся не принимает участия в устных опросах, либо даваемые им ответы принципиально неверны.

Лабораторная работа № 2.
Разработка программной части ИС и АС.

Цель занятия:
Изучить методы сопровождения исходного кода, создания и использования репозиториев методы получения сигналов и данных, методы предварительной обработки информации перед применением бизнес-логики приложения.

В результате выполнения лабораторной работы студент получает навыки создания репозиториев исходного кода, приема данных и написания программ для предварительной обработки принятых данных.

Средства выполнения практической работы:
Проведение занятий предполагается в аудитории, оснащенной:

1. Комплектом персональных ЭВМ в компьютерных классах с выходом в глобальную сеть Интернет.
2. Мультимедийной доской для демонстрации материалов.
3. Установленной ОС Windows или ОС Linux.
4. Установленным текстовым редактором MS Office или Libre Office.
5. Установленным редактором графических схем типа Draw.io.
6. Установленные средства разработки нативного или кроссплатформенного программного обеспечения для языков JavaScript, Python, Pascal, C++, C# или Java по выбору студента.

Результаты освоения материала:
В результате проведения занятия студент должен:
Знать:

- способы современной разработки программного обеспечения с применением нативных или кроссплатформенных инструментов;
- современные инструменты поддержки процесса разработки программного обеспечения, относящиеся к классу CASE-средств.

Уметь:

- применять на практике CASE-средства разработки программного обеспечения, нативные и кроссплатформенные инструменты разработки.

Методика проведения занятия:
Практическое занятие проводится следующим образом:

1. Рассмотрение теоретических вопросов.
2. Разъяснения преподавателя.

3. Диспут или «круглый стол» по тематике занятия.
4. Выполнение заданий преподавателя.
5. Анализ и оценка выполненных работ и степени овладения, обучающихся запланированными компетенциями.

Содержание практического занятия:
В рамках практического занятия подробно рассматриваются и обсуждаются следующие ключевые вопросы

1. Нативные программные средства, инструменты и библиотеки.
2. Кроссплаатформенные программные средства, инструменты и библиотеки.
3. Принципы выполнения программного кода на уровне микропроцессора ЭВМ.
4. Способы компиляции программ.
5. Принципы работы компиляторов.
6. Принципы работы интерпретаторов скриптовых языков программирования.
7. Способы кросскомпиляции приложений.
8. Использование виртуальных машин и технологий виртуализации при разработке программного обеспечения.
9. Технология работы с языком Си.
10. Технология работы с языком Pascal.
11. Технология работы с веб-стеком для разработки программного обеспечения.
12. Технология работы со стеком Java.
13. Технология работы со стеком C#.
14. Типовая структура проекта.
15. Использование систем контроля версии для управления проектами.

В общем случае, каждому студенту предлагается разработать программную реализацию приема данных от аппаратного источника данных или ее модели с предварительной обработкой полученных данных.

Это общее задание, а частные случаи варьируются относительно варианта, а также желания и возможностей студентов, как и в первой лабораторной работе.

Студенту формируется следующее задание:

1. Доработка задания из первой лабораторной работы, если это необходимо.
2. Доработка структурной схемы программного обеспечения, если это необходимо.

3. Расширение структурной схемы блока приема сигнала от датчика в зависимости от выбранного варианта источника:
   - аппаратный аналоговый датчик,
   - аппаратный цифровой датчик.
   - математическая модель датчика с генератором цифрового сигнала,
   - внешнее API (в сети Интернет).
4. Создание и инициализация репозитория исходного кода в сети Интернет.
5. Реализация блока приема данных от датчика.
6. Реализация буфера принятого сигнала от датчика.
7. Реализация блока дискретизации или синхронизации выборки данных в соответствии с заданной частотой выборки.
8. Реализация блока предварительной обработки данных от датчика согласно модели цифровой фильтрации или усреднения (скользящие средние).

Для облачного документооборота по-прежнему предлагается использовать OneDrive или Google Drive, но выбор может быть изменен со стороны студента.

Для составления или редактирования UML-диаграмм по-прежнему предлагается использовать бесплатный инструмент Draw.io, который существует в виде отдельного приложения, в виде онлайн-ресурса и даже в виде расширения для Google Drive.

Для оформления или редактирования структуры базы данных по-прежнему предлагается использовать бесплатный ресурс dbdiagram.io или StarUML.

Для Управления проектами по-прежнему предлагается использовать бесплатный вариант JIRA или его аналоги типа Trello.

В идеальном случае, для получения сигналов и их предварительной обработки (как и в Лабораторной работе № 1) предполагается использовать аппаратные комплекты разработчиков типа:

1. Arduino.
2. Raspberry Pi.
3. STM SDK.

Если у студентов не появляется возможность использовать физические устройства, то им предлагается эмулировать процесс получения входных данных одним из следующих способов (как это было описано в руководстве к Лабораторной работе № 1):

1. Математически.
2. При помощи файлов или базы данных с отсчетами или измерениями.
3. При помощи внешних сетевых API (открытых источников).

Если в Лабораторной работе 1 предлагается выбрать способ получения исходных данных, то в данной Лабораторной работе 2 требуется уже реализовать блок приема данных из внешнего источника.

Напомним, что большинство студентов может выбрать математическую модель сигнала, поэтому приведем здесь способ обработки таких данных.

Для генерации входных сигналов можно использовать аддитивную модель системы гармонических сигналов, которые можно генерировать на основе различных амплитуд и частот следующим образом:

X = A1*sin(2*pi*F1) + A2*sin(2*pi*F2) + … + Ak*sin(2*pi*Fk).

Здесь присутствует вектор амплитуд гармоник A и вектор частот гармоник F. Длины этих векторов одинаковы и равны k.

В результате сложения этих гармоник получается наложение сигналов разной силы и частоты, что позволяет имитировать источник случайных сигналов и получать разные формы.

Предлагается выбрать вектора амплитуд и частот на базе следующих вариантов.

Вариант	**Вектор A**	**Вектор F**
1	1, 0.5, 0.25	0.1, 0.1, 0.4
2	2.5, 1.5, 0.35	0.55, 0.2, 1.4
3	2, 1.5, 0.75	0.5, 0.7, 2.4
4	1.5, 0.5, 1.25	0.75, 1.1, 1.64
5	1, 5.5, 1.75	1.5, 0.3, 2.2

В качестве алгоритма обработки данных можно использовать разные варианты. Давайте рассмотрим применение скользящих средних для сглаживания сигналов.

Скользящие средние

В основе применения скользящего среднего лежит следующая формула:

EMA = (P * alpha) + (EMAprev * (1 - alpha))

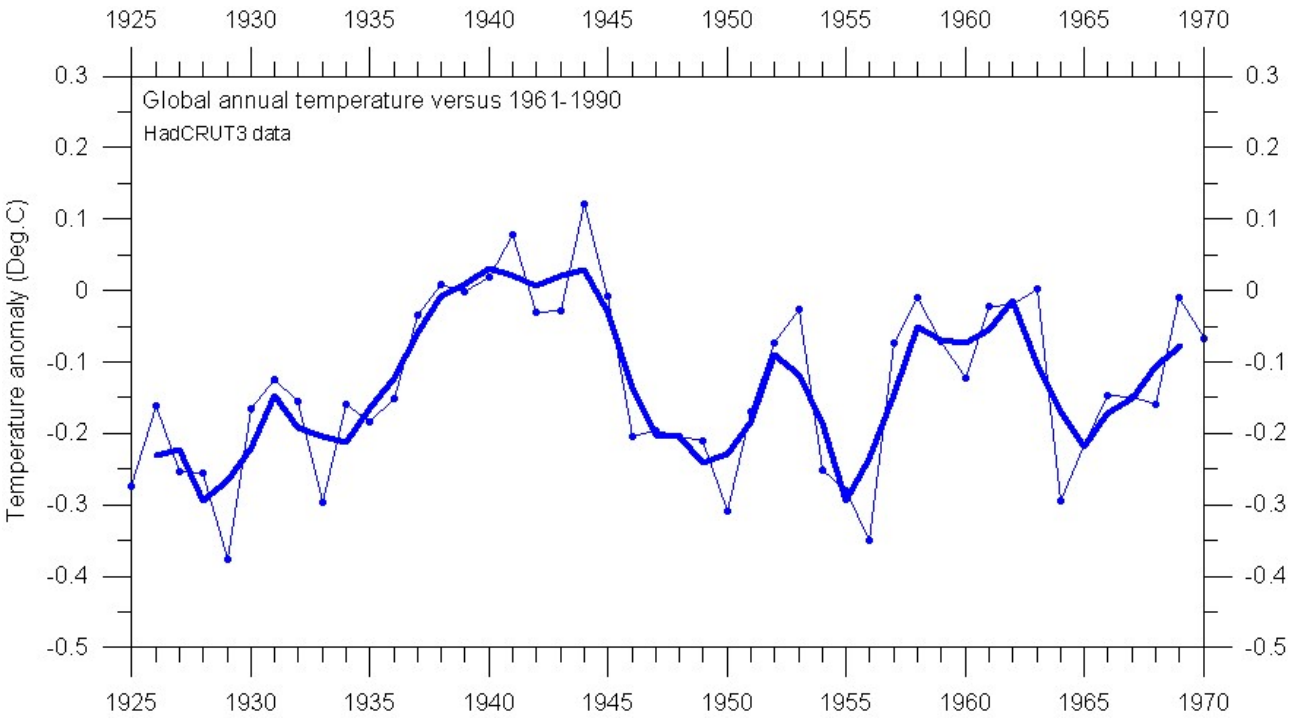
P — текущее значение сигнала
alpha — сглаживающий фактор, например 2 / (1 + N)
N — количество периодов сглаживания

Период N для EMA выбирается экспериментальным путем исходя из исторических данных.

Скользящие средние применяются для предиктивной аналитики или для сглаживания сигналов, - мы вполне можем взять их для примера алгоритма предварительной обработки.

Тем более, что одним из вариантов исходного сигнала для первой лабораторной работы мы предлагаем использовать данных о курсах валют или значения валютных пар, полученных из внешних источников типа API.

Пример использования скользящих средних с температурными данными.



Цифровой фильтр.

Это альтернативный вариант обработки информации. Цифровые фильтры — это сложная тема, но мы можем упростить материал, используя готовую формулу:

Yn = SUM Xn-j * Bj

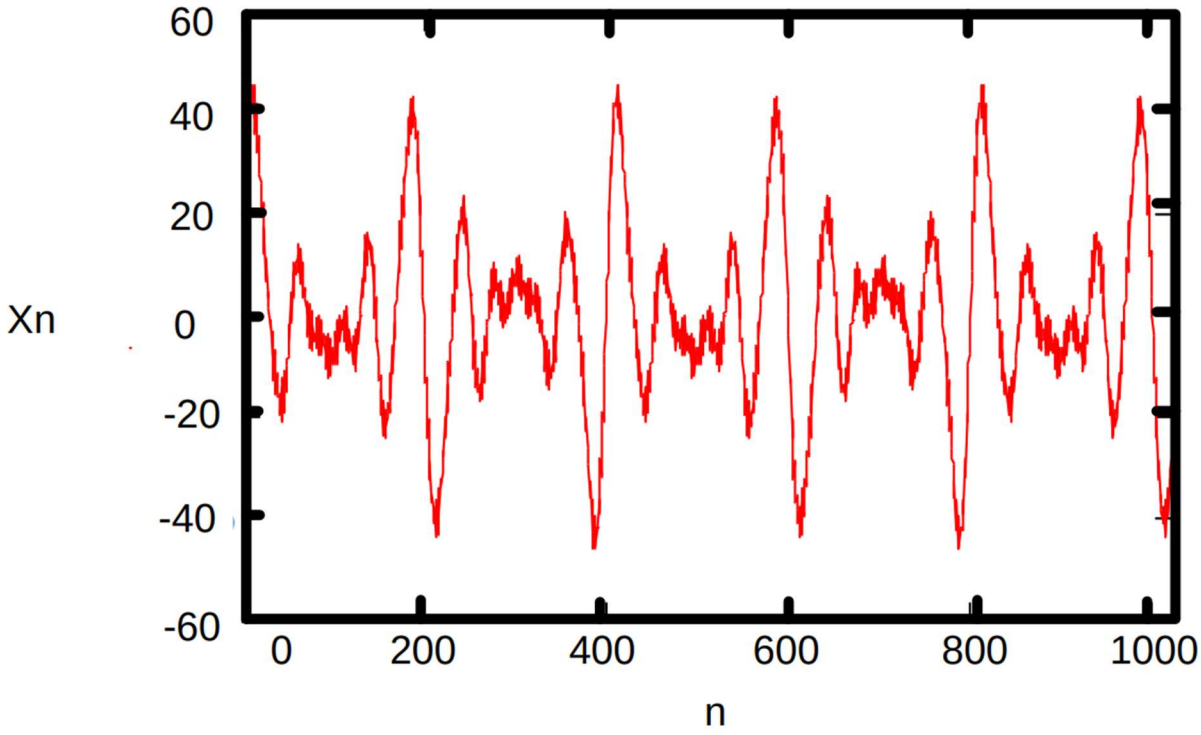
Yn — это значение выходного сигнала в текущей точке

Xn-j — это значение входного сигнала, отстоящего от текущей точки на j позиций в прошлое

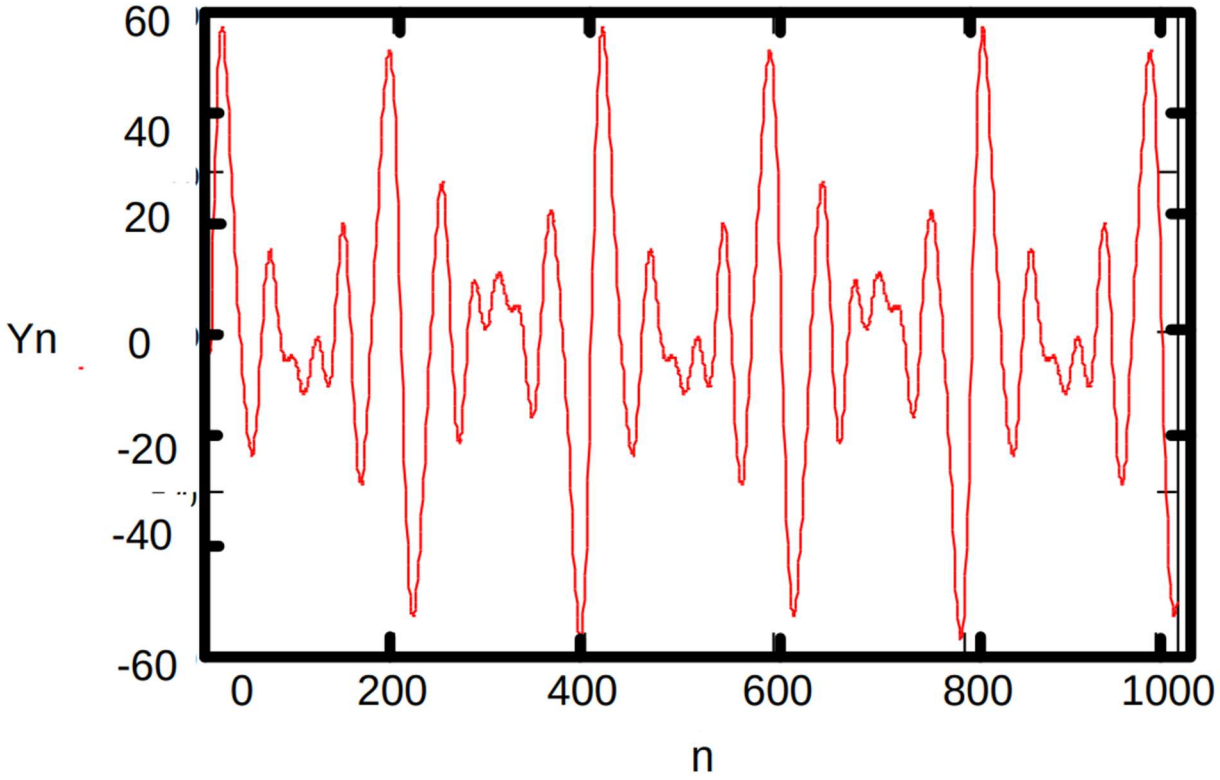
Bj — это коэффициент фильтра с номером j.

Имея обработанные таким образом данные, мы получаем симуляцию блоков работы с сырыми данными.

Пример сгенерированного входного сигнала до фильтрации.



После фильтрации с правильно подобранными коэффициентами получается более «чистая» форма без высокочастотных примесей.



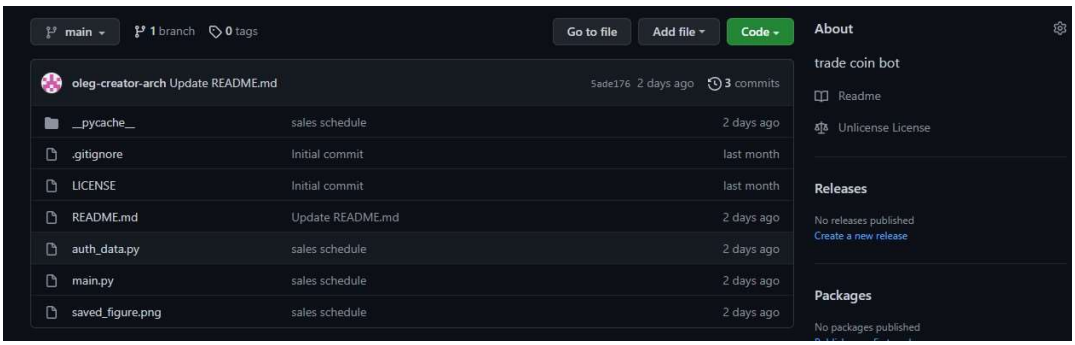
На выходе этого процесса у нас есть массивы готовых цифровых данных, которые можно передавать на следующие этапы работы системы и использовать для выполнения задания к лабораторной работе.

После реализации некоторого количества алгоритмов в виде программного кода, его желательно зафиксировать до следующей лабораторной работы. Для этого лучше использовать репозитории типа Github или Gitlab.

Напомним, что, для работы с репозиториями у нас в системе должна быть установлена программа Git. Для ОС Windows это программное обеспечение нужно устанавливать отдельно.
Для создания репозитория введем команды в консоли

1 git init
2 git add .
3 git commit −m ’Initial commit’
4 git remote −v
5 git remote add
origin ’https://github.com/oleg−creator−arch/trade_coin_bot.git’
6 git push origin main

Созданный проект на сайте Github, может выглядеть следующим образом.



Теперь нам доступны команды:

- Клонирования: git clone https://github.com/repo/repo.git
- Принятия изменений: git pull
- Создания ветви: git checkout –b newbranch
- Переход на существующую ветку: git checkout branchname
- Добавления файлов в индекс: git add .
- Фиксации изменения (коммита): git commit –m “new commit”
- Отправки изменений на сервер Github: git push

И так далее.

Критерии оценивания:

Максимальное количество баллов за практическую работу: 10 баллов.
9-10 баллов– обучающийся в течение модуля активно принимает участие в устных опросах, даваемые им ответы верны и позволяют высоко оценить уровень владения материалом;

7-8 баллов – обучающийся принимает участие в устных опросах, даваемые им ответы, как правило, верны и позволяют оценить владение материалом на достаточном уровне;

6 баллов – обучающийся не проявляет инициативы для участия в устных опросах, а даваемые им ответы фрагментарны и верны лишь частично, что не позволяет оценить владение материалом на достаточном уровне;

1-5 баллов – обучающийся пытается участвовать в собеседовании, но дает неверные ответы, показывает отсутствие базовых знаний;

0 баллов – обучающийся не принимает участия в устных опросах, либо даваемые им ответы принципиально неверны.

Лабораторная работа № 3.
Разработка аппаратной части ИС и АС или ее модели.

Цель занятия:
Изучить способы управления версиями программного кода, разработать каркас приложений, научиться работать с цифровыми данными, применять алгоритмы предварительной обработки информации, подготавливая для дальнейшего применения, применять CASE-средства поддержки процесса разработки программного обеспечения.

Средства выполнения практической работы:
Проведение занятий предполагается в аудитории, оснащенной:

1. Комплектом персональных ЭВМ в компьютерных классах с выходом в глобальную сеть Интернет.
2. Мультимедийной доской для демонстрации материалов.
3. Установленной ОС Windows или ОС Linux.
4. Установленным текстовым редактором MS Office или Libre Office.
5. Установленным редактором графических схем типа Draw.io.
6. Установленные средства разработки нативного или кроссплатформенного программного обеспечения для языков JavaScript, Python, Pascal, C++, C# или Java по выбору студента.

В результате проведения занятия студент должен:
Знать:

- способы современной разработки программного обеспечения с применением нативных или кроссплатформенных инструментов;
- современные инструменты поддержки процесса разработки программного обеспечения, относящиеся к классу CASE-средств;
- методы и алгоритмы обработки сигналов различной физической природы.

Уметь:

- применять на практике CASE-средства разработки программного обеспечения, нативные и кроссплатформенные инструменты разработки.

Методика проведения занятия:
Практическое занятие проводится следующим образом:

1. Рассмотрение теоретических вопросов.
2. Разъяснения преподавателя.
3. Диспут или «круглый стол» по тематике занятия.
4. Выполнение заданий преподавателя.

5. Анализ и оценка выполненных работ и степени овладения, обучающихся запланированными компетенциями.

Содержание практического занятия:
В рамках практического занятия подробно рассматриваются и обсуждаются следующие ключевые вопросы

1. Нативные программные средства, инструменты и библиотеки.
2. Кроссплатформенные программные средства, инструменты и библиотеки для программирования аппаратуры или для разработки моделей.
3. Принципы выполнения программного кода на уровне микроконтроллера.
4. Способы компиляции программ и их прошивки.
5. Принципы работы компиляторов для микроконтроллеров.
6. Способы кросскомпиляции приложений.
7. Использование виртуальных машин и технологий виртуализации при разработке программного обеспечения, в том числе и для аппаратных моделей.
8. Технология работы с языком Си.
9. Технология работы с языком Pascal.
10. Технология работы с веб-стеком для разработки программного обеспечения.
11. Технология работы со стеком Java.
12. Технология работы со стеком C#.
13. Типовая структура проекта.
14. Использование систем контроля версии для управления проектами.
15. Методы предварительной обработки сигналов.
16. Алгоритмы цифровой фильтрации.
17. Алгоритмы агрегации данных и их усреднения.

В общем случае, каждому студенту предлагается разработать программную реализацию приема данных от аппаратного источника данных или ее модели с предварительной обработкой полученных данных.

Это общее задание, а частные случаи варьируются относительно варианта, а также желания и возможностей студентов, как и в первой лабораторной работе.

Студенту формируется следующее задание:

1. Доработка задания из первой и второй лабораторной работы, если это необходимо.
2. Доработка структурной схемы программного обеспечения, если это необходимо.

3. Расширение структурной схемы блока приема сигнала от датчика в зависимости от выбранного в Лабораторной работе № 1 варианта источника данных:
   - аппаратный аналоговый датчик,
   - аппаратный цифровой датчик.
   - математическая модель датчика с генератором цифрового сигнала,
   - внешнее API (в сети Интернет).
4. Доработка блока приема данных от датчика.
5. Доработка буфера принятого сигнала от датчика.
6. Доработка блока дискретизации или синхронизации выборки данных в соответствии с заданной частотой выборки.
7. Реализация блока предварительной обработки данных от датчика согласно модели цифровой фильтрации или усреднения (скользящие средние).

Для облачного документооборота по-прежнему предлагается использовать OneDrive или Google Drive, но выбор может быть изменен со стороны студента.

Для составления или редактирования UML-диаграмм по-прежнему предлагается использовать бесплатный инструмент Draw.io, который существует в виде отдельного приложения, в виде онлайн-ресурса и даже в виде расширения для Google Drive.

Для оформления или редактирования структуры базы данных по-прежнему предлагается использовать бесплатный ресурс dbdiagram.io или StarUML.

Для Управления проектами по-прежнему предлагается использовать бесплатный вариант JIRA или его аналоги типа Trello.

В идеальном случае, для получения сигналов и их предварительной обработки предполагается использовать аппаратные комплекты разработчиков типа:

1. Arduino.
2. Raspberry Pi.
3. STM SDK.

Если в Лабораторной работе 1 предлагается выбрать способ получения исходных данных, в Лабораторной работе 2 требуется реализовать блок приема данных из внешнего источника, а в данной Лабораторной работе 3 требуется

использовать или моделировать аппаратный блок для обработки данных и их защиты.

Напомним, что студенты могут выбрать математическую модель сигнала, поэтому приведем здесь способ обработки таких данных. Однако в этом случае им также необходимо изучить и сымитировать программно методы дискретизации и квантования сигналов.

На выходе этого процесса у нас есть массивы готовых цифровых данных, которые можно передавать на следующие этапы работы системы и использовать для выполнения задания к лабораторной работе.

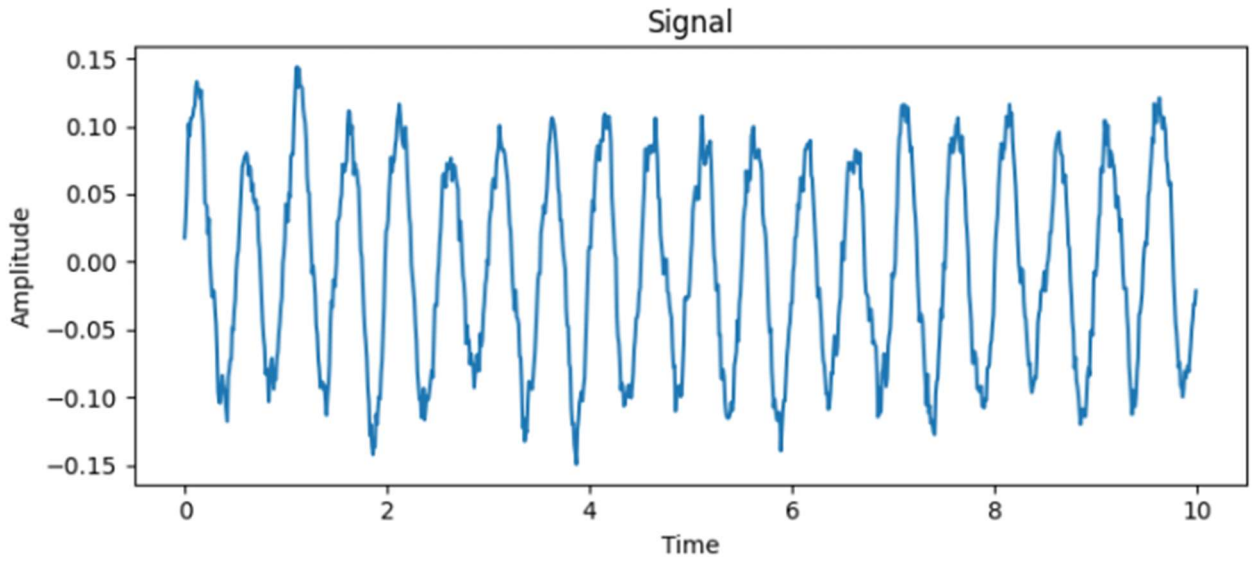
Также на этом этапе работы необходимо выполнить визуализацию полученных данных в виде графика.

Например, при использовании языка Python для разработки можно воспользоваться специальными математическими библиотеками, как показано в примере ниже.

```
import matplotlib.pyplot as plt
import numpy as np

plt.rcParams["figure.figsize"] = [7.50, 3.50]
plt.rcParams["figure.autolayout"] = True

np.random.seed(0)

dt = 0.01 # sampling interval
Fs = 1 / dt # sampling frequency
t = np.arange(0, 10, dt)

# generate noise:
nse = np.random.randn(len(t))
r = np.exp(-t / 0.05)
cnse = np.convolve(nse, r) * dt
cnse = cnse[:len(t)]
s = 0.1 * np.sin(4 * np.pi * t) + cnse
fig, axs = plt.subplots()
```

```
axs.set_title("Signal")
axs.plot(t, s, color='C0')
axs.set_xlabel("Time")
axs.set_ylabel("Amplitude")
```

В результате мы получим график смоделированного сигнала.



Критерии оценивания:

Максимальное количество баллов за практическую работу: 10 баллов.
9-10 баллов– обучающийся в течение модуля активно принимает участие в устных опросах, даваемые им ответы верны и позволяют высоко оценить уровень владения материалом;

7-8 баллов – обучающийся принимает участие в устных опросах, даваемые им ответы, как правило, верны и позволяют оценить владение материалом на достаточном уровне;

6 баллов – обучающийся не проявляет инициативы для участия в устных опросах, а даваемые им ответы фрагментарны и верны лишь частично, что не позволяет оценить владение материалом на достаточном уровне;

1-5 баллов – обучающийся пытается участвовать в собеседовании, но дает неверные ответы, показывает отсутствие базовых знаний;

0 баллов – обучающийся не принимает участия в устных опросах, либо даваемые им ответы принципиально неверны.

Лабораторная работа № 4.
Обеспечение защиты и хранения данных ИС и АС.

Цель занятия:
Разработать структуру хранилища данных, подключить инструменты хранения данных (базе данных) типа SQL или NoSQL и обеспечить безопасное хранение или передачу простым шифрованием.

Средства выполнения практической работы:
Проведение занятий предполагается в аудитории, оснащенной:

1. Комплектом персональных ЭВМ в компьютерных классах с выходом в глобальную сеть Интернет.
2. Мультимедийной доской для демонстрации материалов.
3. Установленной ОС Windows или ОС Linux.
4. Установленным текстовым редактором MS Office или Libre Office.
5. Установленным редактором графических схем типа Draw.io.
6. Установленные средства разработки нативного или кроссплатформенного программного обеспечения для языков JavaScript, Python, Pascal, C++, C# или Java по выбору студента.
7. Локальной базой данных (сервером или просто драйвером SQLite3).

Результаты освоения материала:
В результате проведения занятия студент должен:
Знать:

- способы современной разработки программного обеспечения с применением нативных или кроссплатформенных инстурментов;
- современные инструменты поддержки процесса разработки программного обеспечения, относящиеся к классу CASE-средств;
- способы проектирования баз данных;
- способы обеспечения защиты данных методами шифрования.

Уметь:

- применять на практике CASE-средства разработки программного обеспечения, нативные и кроссплатформенные инструменты разработки, а также проектирования.

Методика проведения занятия:
Практическое занятие проводится следующим образом:

1. Рассмотрение теоретических вопросов.
2. Разъяснения преподавателя.

3. Диспут или «круглый стол» по тематике занятия.
4. Выполнение заданий преподавателя.
5. Анализ и оценка выполненных работ и степени овладения, обучающихся запланированными компетенциями.

Содержание практического занятия:
В рамках практического занятия подробно рассматриваются и обсуждаются следующие ключевые вопросы

1. Современные SQL базы данных.
2. Современные NoSQL базы данных.
3. Методы проектирования структуры баз данных.
4. CASE-средства для разработки структур баз данных.

В общем случае, каждому студенту предлагается разработать программное решение для трансляции принятых и обработанных данных от источника данных или ее модели после предварительной обработки в базу данных с защитой информации.

Это общее задание, а частные случаи варьируются относительно варианта, а также желания и возможностей студентов, как и в первой лабораторной работе.

Студенту формируется следующее задание:

1. Доработка задания из второй лабораторной работы, если это необходимо.
2. Разработка структурной схемы базы данных при помощи CASE-средств.
3. Создание схемы базы данных при помощи драйвера базы данных.
4. Реализация блока приема данных после обработки.
5. Реализация программной прослойки над движком базы данных.
6. Реализация блока защиты информации.
7. Реализация блока сохранения информации в базу данных.
8. Реализация блока чтения данных из базы данных.

Для облачного документооборота по-прежнему предлагается использовать OneDrive или Google Drive, но выбор может быть изменен со стороны студента.

Для составления или редактирования UML-диаграмм по-прежнему предлагается использовать бесплатный инструмент Draw.io, который существует в виде отдельного приложения, в виде онлайн-ресурса и даже в виде расширения для Google Drive.

Для оформления или редактирования структуры базы данных по прежнему предлагается использовать бесплатный ресурс dbdiagram.io или StarUML.

Для Управления проектами по-прежнему предлагается использовать бесплатный вариант JIRA или его аналоги типа Trello.

В идеальном случае, для сохранения данных после их предварительной обработки предполагается по-прежнему использовать в качестве источников аппаратные комплекты разработчиков типа:

1. Arduino.
2. Raspberry Pi.
3. STM SDK.

Но, по-прежнему, допускается эмулировать процесс получения входных данных одним из следующих способов:

1. Математически.
2. При помощи файлов или базы данных с отсчетами или измерениями.
3. При помощи внешних сетевых API (открытых источников).

В завершении лабораторной работы мы получаем структуру базы данных, таблицы, поля и типы полей. Несколько таблиц необходимо связать друг с другом.

В качестве примера мы можем предложить следующую структуру, сформированную при помощи dbdiagram.io.

При помощи псевдо-языка описания структуры базы данных мы можем реализовать следующую схему:

1. Сущность Пользователь, имеющий поля id, username, email, created, active.

```
Table User {
  id int [pk]
  username varchar [not null]
  email varchar [not null, unique]
  created datetime
   active boolean
}
```

2. Сущность Платформа, имеющая поля id, name, created, owner_id, address_id.

```
Table Platform {
  id int [pk]
  name varchar [not null]
  created datetime
  owner_id int [ref: > User.id]
  address_id int [ref: - Address.id]
}
```

3. Сущность Адрес, где установлена Платформа с полями id, line_1, line_2, city, country, town, latitude, longitude, ip

```
Table Address {
  id int [pk]
  line_1 varchar [not null]
  line_2 varchar
  city varchar
  town varchar
  country varchar
  latitude float4
  longitude float4
  ip ip
}
```

4. Сущность Сенсор, имеющий поля id, name, uuid, created, platform_id

```
Table Sensor {
  id int [pk]
  name varchar [not null]
  uuid uuid
  created datetime
  platform_id int [ref: > Platform.id]
}
```

5. Сущность Измерение, имеющее поля id, value, created, sensor_id

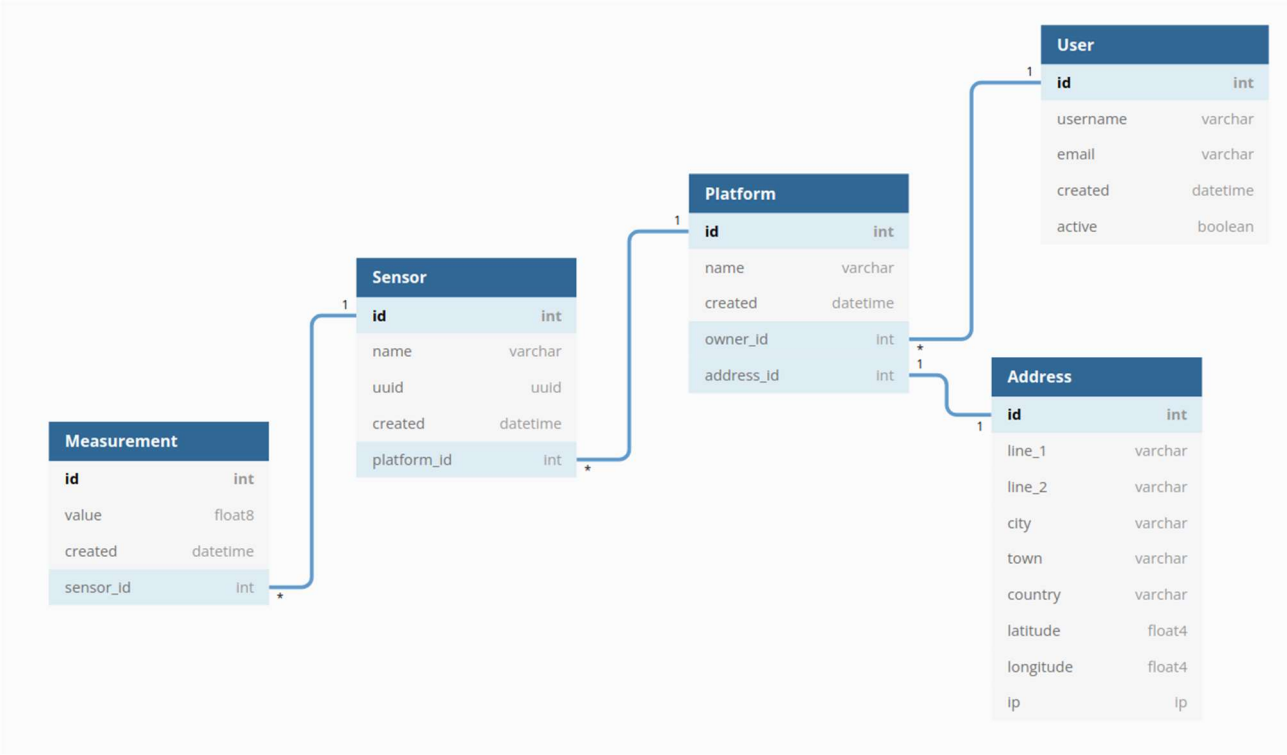
```
Table Measurement {
```

```
  id int [pk]
  value float8
  created datetime
  sensor_id int [ref: > Sensor.id]
}
```

Сущности связаны следующими отношениями:

1. Один пользователь может иметь несколько Платформ, но Платформа принадлежит одному Пользователю (один ко многим).
2. На одной Платформе может быть установлено несколько Сенсоров, но каждый Сенсор может быть только на одной Платформе (один ко многим).
3. Каждая Платформа имеет только один Адрес (один к одному).
4. Каждый Сенсор формирует Измерения (один ко многим).

Схематично, при помощи CASE-средств мы можем сформировать структуру БД следующим образом.



По заданию лабораторной работы данные должны быть защищены при хранении или при передаче. Если мы говорим о работе с внешним API по протоколу HTTPS, то это итак происходит благодаря алгоритмам, заложенным в сам протокол.

Если мы говорим о защите на уровне базы данных, то можно использовать специальные расширения типа pg_crypto для базы данных PostgreSQL (то есть решение на этом уровне зависит от типа базы данных используемой в проекте).

Если нужно защищать данные на программном уровне, то можно воспользоваться стандартным алгоритмом шифрования, входящим в используемый фреймворк или реализовать его самостоятельно, пользуясь примерами и документацией в сети Интернет.

Главное, разобраться во всех этапах создания системы или программно-аппаратных комплексов, в современных правилах формирования и сопровождения проектов.

Итак, в заключении цикла лабораторных работ 1 — 4 мы получили систему, которая выполняет следующие действия:

1. Получает данные от аналогового или цифрового датчика (или от их модели/внешнего API).
2. Проводит дискретизацию аналогового сигнала или семплирование цифровых данных.
3. Выполняет цифровую обработку одним из предложенных алгоритмов, например, скользящими средними или цифровым фильтром, в зависимости от типа данных.
4. Сохраняет информацию в базе данных.
5. Защищает информацию одним из методов шифрования.

Мы можем считать подобную систему полнофункциональным элементом системы Интернета Вещей, состоящим из аппаратной части (или ее модели) и программной части, базы данных и каналов связи.

Критерии оценивания:

Максимальное количество баллов за практическую работу: 10 баллов.
9-10 баллов– обучающийся в течение модуля активно принимает участие в устных опросах, даваемые им ответы верны и позволяют высоко оценить уровень владения материалом;

7-8 баллов – обучающийся принимает участие в устных опросах, даваемые им ответы, как правило, верны и позволяют оценить владение материалом на достаточном уровне;

6 баллов – обучающийся не проявляет инициативы для участия в устных опросах, а даваемые им ответы фрагментарны и верны лишь частично, что не позволяет оценить владение материалом на достаточном уровне;

1-5 баллов – обучающийся пытается участвовать в собеседовании, но дает неверные ответы, показывает отсутствие базовых знаний;

0 баллов – обучающийся не принимает участия в устных опросах, либо даваемые им ответы принципиально неверны.