

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

**ОТЧЕТ ПО УЧЕБНОЙ (ТЕХНОЛОГИЧЕСКОЙ
(ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ)) ПРАКТИКЕ**
(2024/2025 учебный год)

_____ Культиясова Ярослава Кирилловича

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 2 года

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 20.06.2025 по 03.07.2025

Кафедра «Вычислительная техника» _____

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики к.т.н., доцент, Карамышева Н.С.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ
(ТЕХНОЛОГИЧЕСКОЙ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ))
ПРАКТИКИ
(2024/2025 учебный год)**

_____ Культиясова Ярослава Кирилловича

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 2 года

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 20.06.2025 по 03.07.2025

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики к.т.н., доцент, Карамышева Н.С.

(должность, ученая степень, ученое звание)

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя практики от вуза
1	Ознакомление с программой практики требованиями и порядком защиты отчета по практике, выдача задания	6	20.06.25 - 21.06.25	
2	Сбор, обработка, систематизация материала, выполнение содержательной части индивидуального задания	68	21.06. 25 – 26.07. 25	
3	Анализ результатов выполнения индивидуального задания, оформление отчёта	22	26.07.25 – 02.07. 25	
4	Проведение мероприятий промежуточной аттестации	12	02.07.25 – 03.07. 25	
	Общий объём часов	108		

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЁТ
О ПРОХОЖДЕНИИ УЧЕБНОЙ (ТЕХНОЛОГИЧЕСКОЙ
(ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ)) ПРАКТИКИ

(2024/2025 учебный год)

Культиясова Ярослава Кирилловича

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 2 года

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 03.07.2025

Кафедра «Вычислительная техника»

Культиясов Я.К. организовывал и планировал работу команды бакалавров: Максим С.Н., Панькин М.Д., распределял задачи между участниками команды, мотивировал их, доложил о результатах руководителю практики. Для проведения практики были изучены и проанализированы учебно-методические материалы по учебной программе. Согласован с руководителем практики список вариантов заданий для студентов. Дистанционно произвел организацию групповой работы с помощью Scrum методологии, системы контроля версий Git, включая отправку заданий, методического материала, проверка корректности выполненных заданий и оформления отчетов.

Магистр Культиясов Я.К. " " 2025 г.

Руководитель Карамышева Н.С. " " 2025 г.
практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

**ОТЗЫВ
О ПРОХОЖДЕНИИ УЧЕБНОЙ (ТЕХНОЛОГИЧЕСКОЙ
(ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ)) ПРАКТИКИ**

(2023/2024 учебный год)

Культиясова Ярослава Кирилловича

Направление подготовки 09.04.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Прикладной искусственный интеллект»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 2 года

Год обучения 1 семестр 2

Период прохождения практики с 20.06.2025 по 03.07.2025

Кафедра «Вычислительная техника»

В процессе прохождения практики Культиясов Я.К. закрепил и расширил теоретические знания в сфере информатики и вычислительной техники.

За период прохождения практики магистром были проведены подготовительные работы для организации командной работы (дистанционно). А также произведен анализ и выявление типовых ошибок при выполнении заданий и оформлении отчетов. Во время практики показал себя ответственным, добросовестным руководителем, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию, знаниями о современных технологиях образовательного процесса. Культиясов Я.К. свободно ориентировался во всех аспектах методики управления групповой работой во время практики.

Практика Культиясова Я.К. заслуживает оценки « ».

Научный

руководитель к.т.н., доцент, Карамышева Н.С. « » 2025 г.

ОГЛАВЛЕНИЕ

<u>Введение.....</u>	<u>7</u>
1. <u>Постановка задачи.....</u>	<u>8</u>
2. <u>Учебное задание.....</u>	<u>9</u>
3. <u>Методология организации командной деятельности.....</u>	<u>13</u>
4. <u>Организация дистанционного рабочего пространства.....</u>	<u>18</u>
5. <u>Выполнение работы.....</u>	<u>21</u>
<u>Заключение.....</u>	<u>25</u>
<u>Список литературы.....</u>	<u>26</u>
<u>Приложение № 1.....</u>	<u>27</u>
<u>Приложение № 2.....</u>	<u>31</u>

ВВЕДЕНИЕ

Разработка программного обеспечения – это групповой процесс. Большинство программных продуктов рождается в результате коллективных усилий. Люди являются одной из главных составляющих, необходимых для производства программного продукта. При организации работы программистов важно учитывать не только их технические навыки, но и взаимодействие между ними.

Прохождению учебной практики должно предшествовать изучение студентами дисциплин: «Технология разработки программного обеспечения», «Мировые тенденции развития вычислительной техники и информационных технологий».

Для успешного прохождения учебной практики студент должен:

знать:

- основные технологии разработки программного обеспечения;
- технические и программные средства реализации взаимодействия программного обеспечения;

уметь:

- разрабатывать программное обеспечение на языках высокого уровня;
- работать с информацией в глобальных компьютерных сетях;

владеть:

- методикой решения прикладных задач по профилю своей специальности;
- методами и средствами разработки и оформления технической документации.

В ходе прохождения данной практики была проведена организация дистанционного учебного процесса для бакалавриата 1 курса с использованием современных технологий коммуникаций.

1. ПОСТАНОВКА ЗАДАЧИ

Целями учебной (технологической (проектно-технологической)) практики являются закрепление и углубление теоретической подготовки обучающегося, знакомство с практическими аспектами профессиональной деятельности по интеграции программного обеспечения в рамках одного проекта для решения практических задач. А так же изучение принципов формирования команды для коллектива программистов и особенностей командной работы.

Задачами учебной (проектно-технологической) практики являются:

- получение умений по объединению нескольких программных модулей в области решения профессиональных задач с применением программных средств;
- углубление навыков по сбору, обработке, анализу и систематизации информации по теме задания;
- углубление навыков по документированию программных средств, подготовке отчётов;
- получение навыков работы с современными программными и аппаратными средствами информационных технологий.

При прохождении учебной практики используются подходы группового разделения сфер исследования существующих источников, включая научную, профессионально-техническую, и учебно-методическую литературу, анализа полученных результатов и консолидации результатов индивидуальных исследований в обобщенный учебно-методический материал.

При выполнении данной работы учитывается индивидуальный опыт, полученный в ходе выполнения группового задания и анализа учебно-методического материала в течение практики.

2. УЧЕБНОЕ ЗАДАНИЕ

Необходимо реализовать и продемонстрировать работу алгоритмов сортировки или поиска элементов из списка, представленного ниже. Конечная программа должна реализовывать непосредственно алгоритм, а также выполнять считывание входных данных из файла и запись результатов в файл, и отображать все данные в графическом интерфейсе.

Форматы входных и выходных файлов определяются произвольно, для массивов желательно использование формата CSV (значения разделены запятой). Программа должна обрабатывать данные, предоставленные в корректно сформированных входных файлах.

Конечные программы не должны использовать реализации алгоритмов сортировки или поиска элементов, предоставляемые стандартными или сторонними библиотеками, включая системные.

В случае наличия реализации требуемого алгоритма в стандартных библиотеках языка программирования выполнить сравнение производительности реализованного алгоритма со стандартным. Сделать выводы.

Отчет должен содержать следующие элементы:

- краткое описание алгоритма;
- блок-схема алгоритма;
- преимущества и недостатки алгоритма;
- типичные сценарии применения;
- непосредственная реализация алгоритма без использования стандартных или сторонних библиотек;
- блок-схема программы;
- анализ работы алгоритма, включая количество базовых операций (для алгоритмов сортировки – число перестановок), производительность для

наилучшего и наихудшего сценариев в секундах (для алгоритмов сортировки – предварительно отсортированный, инвертированный и случайный наборы данных, для алгоритмов поиска – поиск элементов, заведомо находящихся вначале или конце набора данных, включая случай, когда набор данных состоит из многократно повторяющихся элементов);

- анализ и реализация графического интерфейса;
- раздел тестирования готовой программы.

В качестве результатов выполнения работы предоставляются: отчет с описанием реализованного алгоритма и сферы его применения, исполняемая программа, включая ее исходный код, наборы тестовых данных (в зависимости от варианта).

Анализ работы алгоритма должен выполняться на нескольких различных наборах данных.

Учебно-методические материалы содержат 20 вариантов заданий на реализацию алгоритмов сортировки массивов, поиска элементов и работы с множествами чисел. Для заданий определены общие требования к реализации программ и содержанию отчетов о результатах их выполнения.

Для выполнения задания необходимо разделить группу бакалавриата на подгруппы по 2-3 человека. Каждая подгруппа самостоятельно выбирает один из 20 вариантов заданий и разделяет его выполнение следующим образом:

1. Студенты выполняют анализ работы алгоритма, строят его блок-схему.
2. Один студент разрабатывает алгоритм работы программы и реализует его на любом языке программирования.
3. Второй студент разрабатывает интерфейс, для отображения всех возможностей реализованной программы первого студента.
4. Если есть третий студент в подгруппе, то он сопоставляет всё вместе и производит тестирование для проверки корректности работы готовой программы (иначе это выполняет второй студент).

Для выполнения предложенных заданий учебной практики необходимы навыки, полученные в ходе изучения курса «Программирование», а также базовые знания по алгоритмизации. Для реализации программ допустимо использовать любые доступные языки программирования, за исключением случаев использования готовых или стандартных библиотек. Предпочтительным является использование языка Си.

Также необходимо выполнить реализацию программы с интерфейсом пользователя, позволяющим отображать входные и выходные данные программы.

При проверке качества реализации программы необходимо учитывать корректность реализации основного алгоритма, а также реакцию программы на «граничные» наборы данных, при которых возможно выявление ошибок реализации основного алгоритма или кода ввода и вывода данных.

Часть вариантов заданий для бакалавров:

1. Сортировка пузырьком (bubble sort);
2. Шейкерная сортировка (cocktail shaker sort);
3. Сортировка выбором (selection sort);
4. Сортировка вставками (insertion sort);
5. Сортировка Шелла (Shell sort);
6. Быстрая сортировка (quicksort);
7. Двоичная сортировка (binary tree sort);
8. Сортировка слияниями (merge sort);
9. Поразрядная сортировка (radix sort);
10. Пирамидальная сортировка (heapsort);
11. Поиск в неупорядоченных таблицах;
12. Поиск в упорядоченных таблицах;
13. Поиск подстроки в строке: алгоритм Бауэра-Мура (Bauer-Moore);

14. Поиск подстроки в строке: алгоритм Кнута-Мориса-Пратта (Knuth-Morris-Pratt);
15. Линейный и бинарный поиск (linear/binary search algorithm);
16. Сформировать все возможные подмножества из k элементов заданного множества;
17. Сформировать все возможные пары подмножеств при объединении которых получается исходное множество;
18. Сформировать все возможные подмножества из заданного множества, элементы которых располагаются по возрастанию в коде Грея.

3. МЕТОДОЛОГИЯ ОРГАНИЗАЦИИ КОМАНДНОЙ ДЕЯТЕЛЬНОСТИ

При разработке программного обеспечения выполняемые задачи распределяются по ролям. Участники команды, выполняющие конкретную роль, должны рассматривать проект с точки зрения своей специализации и обладать необходимой квалификацией.

При формировании групп учитывалась психологическая совместимость, входящих в них студентов. Задача магистров помочь студентам получить навыки конструктивной работы в коллективе. При формировании и утверждении групп следует избегать как существующих, так и потенциальных конфликтов внутри одной группы. Студенты, не сумевшие сформировать группы, разделены на группы магистром, либо введены в уже существующие группы. Сформированные группы утверждаются магистром.

Студенты на основе своих личных предпочтений и желаний организуют рабочие команды. После выбора варианта задания распределены роли между участниками, достигнута договоренность о способах взаимодействия, а так же определены количество часов в неделю, выделяемых на работу над заданием каждым участником. Данный момент является очень важным, поскольку одной из распространенных причин провала проекта является невыполнение рабочих обязательств одним или несколькими членами группы.

Определена схема и частота проведения собраний внутри группы, а так же с магистрами. Встреча коллектива, совместно работающего над проектом проводилась с помощью мессенджера Discord. Онлайн конференции являлись точкой интеграции полученных результатов выполнения задания и определения дальнейших действий. На конференциях обсуждались и вырабатывались различные решения, касающиеся задания. Например, распределение задач, определение приоритетов, назначение ответственных, изменение плана и т.п.

В случае недостаточности утвержденных собраний для командной работы разработчиков, команда может использовать дополнительное время. Это было полезно на начальных этапах.

Студенты сочетают командную работу с индивидуальной. В процессе разработки каждому из студентов поручены конкретные задачи для решения. Зоны ответственности, роли студентов и другие правила, регулирующие индивидуальную работу и ее сочетание с командной, отслеживались магистром.

Методологией управления проекта выбран Scrum. Методология позволяет максимально эффективно использовать имеющиеся в наличии трудовые и материальные ресурсы команды, весь её технический потенциал, когда разработчики разбиваются на группы, каждая из которых отвечает за своё конкретное направление.

Соответственно, руководитель проекта контролирует и координирует работу всех команд, руководит процессами выработки, обсуждения идей и контролирует их доведение до конечного результата. Такой метод позволяет грамотно собирать все необходимые данные, осуществлять полный контроль и владение ситуацией по выполнению рабочих процессов, при необходимости задать правильную мотивацию.

Scrum адаптивен, позволяет изменять требования в любой момент времени.

- не сложен в изучении, дает возможность экономить время, отказываясь от некритичных действий. С ним можно получить потенциально рабочий продукт по завершении каждой итерации.

- важной частью Scrum является самостоятельно организующаяся, многофункциональная команда, способная вести разработку продукта при минимальном внешнем контроле.

- планирование и управление перечнем требований, предъявляемых к продукту.

- планирование этапов работы.
- взаимодействие между участниками проектной группы.
- анализ и изменение процесса разработки.



Рисунок 1. Процесс работы Scrum

Scrum-разработка предполагает три вида практик:

- встречи на ежедневной основе (Daily Scrum Meeting);
- встречи по обзору итерации (Sprint Review Meeting);
- аварийная остановка спринта (Sprint Abnormal Termination).

Ежедневные скрам-встречи

Такие встречи проводят, прежде чем команда приступит к работе. Они позволяют каждому члену группы представлять, кто и чем конкретно занят на пути к цели в настоящее время. На подобные встречи выделяют не больше 15 минут, и этого вполне достаточно, ведь перед специалистами стоит задача поделиться информацией, а не решить проблемы. Все вопросы обсуждаются отдельно.

Встречи проводит скрам-мастер и задает всем специалистам стандартные вопросы:

- 1) Что ты сделал вчера?
- 2) Что ты сделаешь сегодня?
- 3) С какими проблемами ты столкнулся?
- 4) Все открытые вопросы заносятся в список «Пункты действий».

Встречи по обзору спринта

После итерации проводят демонстрационную встречу, на которой происходит обзор спринта.

В первую очередь демонстрируют результаты спринта. Определить, какие требования из журнала спринта удалось выполнить, а также обсудить результаты с командой. Далее планируются задачи на новую итерацию.

Вторая половина встречи посвящается анализу завершившегося спринта. Команда распределяет методы работы на эффективные и неэффективные, анализирует их, делает выводы, необходимые для повышения продуктивности работы.

В финале встречи резюмируются итоги, планируется новый этап разработки продукта. Далее все повторяется и так до завершения проекта.

Аварийная остановка спринта

К ней прибегают лишь в особых ситуациях. Например, спринт может быть прерван до дедлайна в том случае, когда команда понимает: в этой итерации не удастся добиться поставленных задач по разработке продукта. Еще один вариант – остановку производят если пропала необходимость в поставленной на этот этап цели.

После остановки спринта все участники проекта собираются, чтобы обсудить причины произошедшего и спланировать дальнейшую работу. Далее приступают к планированию нового спринта.

Подводя итоги, можно сделать вывод, что, методология Scrum побуждает команду к более активной и плодотворной работе в проекте, способствует находить новые точки роста и стремиться к постоянному совершенствованию своих знаний и навыков.

4.ОРГАНИЗАЦИЯ ДИСТАНЦИОННОГО РАБОЧЕГО ПРОСТРАНСТВА

Для коммуникации со студентами в условиях дистанционного режима обучения был выбран сервис Discord. Discord – это бесплатная программа, которая позволяет вести общение с помощью простых текстовых сообщений и голосовой связи. Весь функционал мессенджера заточен, под людей которые любят играть в команде.

В случае необходимости можно устроить видео конференцию. Запустить можно на любой операционной системе компьютера, ноутбука, телефона или планшета. Так же есть отдельная веб-версия, которая запускается в браузере. Есть море встроенных функций, при желании можно подключить сторонние модули. При всей своей красоте, она потребляет минимум ресурсов, что позволяет запустить даже на слабых компьютерах.

Возможности Discord:

- Приложение способно организовывать голосовые конференции с настройкой канала связи и работать по принципу push-to-talk, создавать публичные и приватные чаты для обмена текстовыми сообщениями. Программа имеет браузерную версию, отличием от настольной версии является то, что режим push-to-talk работает только если в браузере активна вкладка с приложением;
- В текстовом чате поддерживается присоединение файлов, картинок, вставка ссылок (для некоторых сайтов работает предпросмотр), форматирование текста и смайлы, размер вложения ограничен объёмом 8 МБ;
- Организация выделенного сервера, где позволяет создавать необходимое количество текстовых каналов и голосовых/видео каналов (позволяет транслировать экран или отдельное приложение на компьютере).

Перед началом практики был подготовлен специальный сервер в Discord с названием «19BB1». Его основные характеристики и настройки:

1. Роли:

1.1. Студент – роль выдаётся каждому студенту, который подтвердил свою личность на сервере, позволяет просматривать учебные материалы и разговаривать в голосовых каналах «аудитория»;

1.2. Магистры – роль выдаётся магистрам руководящим практикой, которые были предоставлены для текущей учебной группы, позволяет публиковать учебные материалы и подключаться к голосовым каналам «преподавательская»;

1.3. Модератор – роль может выдаваться только владельцем сервера, позволяет управлять любыми настройками сервера.

2. Голосовые каналы:

2.1. Аудитории – голосовые каналы, где могут проводиться собрания и лекции со студентами, голос магистра приглушает голос всех остальных участников беседы, также позволяет каждому участнику использовать транслировать видео с камеры или экрана своего рабочего стола/приложения;

2.2. Преподавательская – голосовой канал, к которому не могут подключаться студенты, применяется для обсуждения магистрами организационных вопросов.

3. Текстовые каналы:

3.1. Объявления – магистр оставляет здесь свои объявления (например о сборе на организационное собрание, его кратких итогах и прочих организационных объявлений);

3.2. Учебные материалы – все необходимые учебные материалы для прохождения учебной (ознакомительной) практики;

3.3. Варианты – канал с доской задач, где каждый может увидеть в какой находится подгруппе тот или иной студент и какое ему поручено задание, а также здесь публикуются ещё свободные варианты, запросы на которые студент может оставить в этом же канале, позже в любое удобное время магистр может их рассмотреть;

3.4. Чат – свободный канал для общения и обсуждения, открытый для всех пользователей сервера.

5.ВЫПОЛНЕНИЕ РАБОТЫ

В первый день практики было проведено собрание, на котором были оговорены все основные организационные вопросы о прохождении практики и выдано первое задание – поделиться на подгруппы по 2-3 человека и подготовить файл для будущего отчёта, представлены учебные материалы для ознакомления. Магистры так же работали в команде из двух человек. Была проведена лекция о настройке стилей в соответствии с требованиями оформления отчётов из учебных материалов кафедры в программе Microsoft Word, в которой после студенты создавали свои отчёты.

На второй день практики студенты первого курса бакалавра группы 24BOA2 предоставили разделение на подгруппы и варианты:

№ подгруппы	ФИО	Вариант
2	Максин С.Н.	2
2	Панькин М.Д.	2
5	Силявко М.А.	5
5	Семенов Н.Д.	5
5	Юсупов В.Р.	5

В этот день так же были оговорены организационные вопросы, разделение задач внутри подгруппы. Было проведено краткое руководство по работе в Git, рассмотрены основные команды. На данном этапе было решено, что Git будет использоваться по желанию студентов, как задание с повышенной сложностью. Изучение структуры GIT проводилось в среде “Learn Git Branching” (Приложение 1).

Все студенты группы работали бригадами по два и три человека, то есть в бригаде. Каждая бригада провела анализ алгоритма сортировки из полученного задания.

Для студентов, реализующих алгоритм работы на языке программирования, анализ был необходим для понимания механизма работы

алгоритма, чтобы корректно и оптимально написать кода программы. Для студентов, разрабатывающих интерфейс и тестирующих работу программы, анализ был необходим для составления эргономичного графического интерфейса, а также для подбора наборов входных данных для тестирования.

Бригада 2.

1. Студент Максин С.Н. разработал и реализовал алгоритм шейкерной сортировки (cocktail shaker sort) на языке программирования C.

2. Студент Панькин М.Д. разработал и реализовал графический интерфейс для разработанной программы.

Бригада 5.

1. Студент Силявко М.А. разработал и реализовал алгоритм сортировки Шелла (Shellsort) на языке программирования C.

2. Студент Семенов Н.Д. разработал и реализовал графический интерфейс для разработанной программы.

3. Студент Юсупов В.Р. произвел тестирование конечного варианта программы.

Подводя итоги, большая часть половины студентов группы 24ВОА2 для реализации программы использовали язык программирования C. Реже встречаются такие языки как Python и C#.

Студенты проявили свои знания в предметных областях инженерных направлений, а также знания и навыки, полученные за первый год обучения в бакалавриате.

В течении всей практики студенты имели возможность обращаться за помощью к магистрам в личные сообщения Discord, получая советы и необходимые правки.

Не все студенты выполнили возложенные на них задачи при разделении их внутри на подгруппы (бригады).

ЗАКЛЮЧЕНИЕ

В результате прохождения учебной практики было организовано дистанционное пространство для учебного процесса, проведено краткое обучение работы с системой контроля версий Git, проведены необходимые организационные собрания и промежуточные аттестационные мероприятия.

При определении заданий учитывалась необходимость постановки групповой работы, чтобы в процессе ее решения студенты выполнили исследование требуемого к реализации алгоритма, его скоростные характеристики и требования к ресурсам, разработка и реализация корректного, «дружелюбного» интерфейса, и проверка корректности работы реализованного алгоритма с интерфейсом. В процесс выполнения этой объемной работы бакалавриата первого курса расширили познания в сфере анализа и реализации алгоритмов, разработки и построения интерфейса, а также получение навыков тестирования.

Использованные в ходе выполнения учебной практики подходы к групповому разделению задач и делегированию ролей потребовали высокой координации деятельности, особенно на этапе консолидации результатов и их анализа.

В результате прохождения практики команда магистров приобрела навыки:

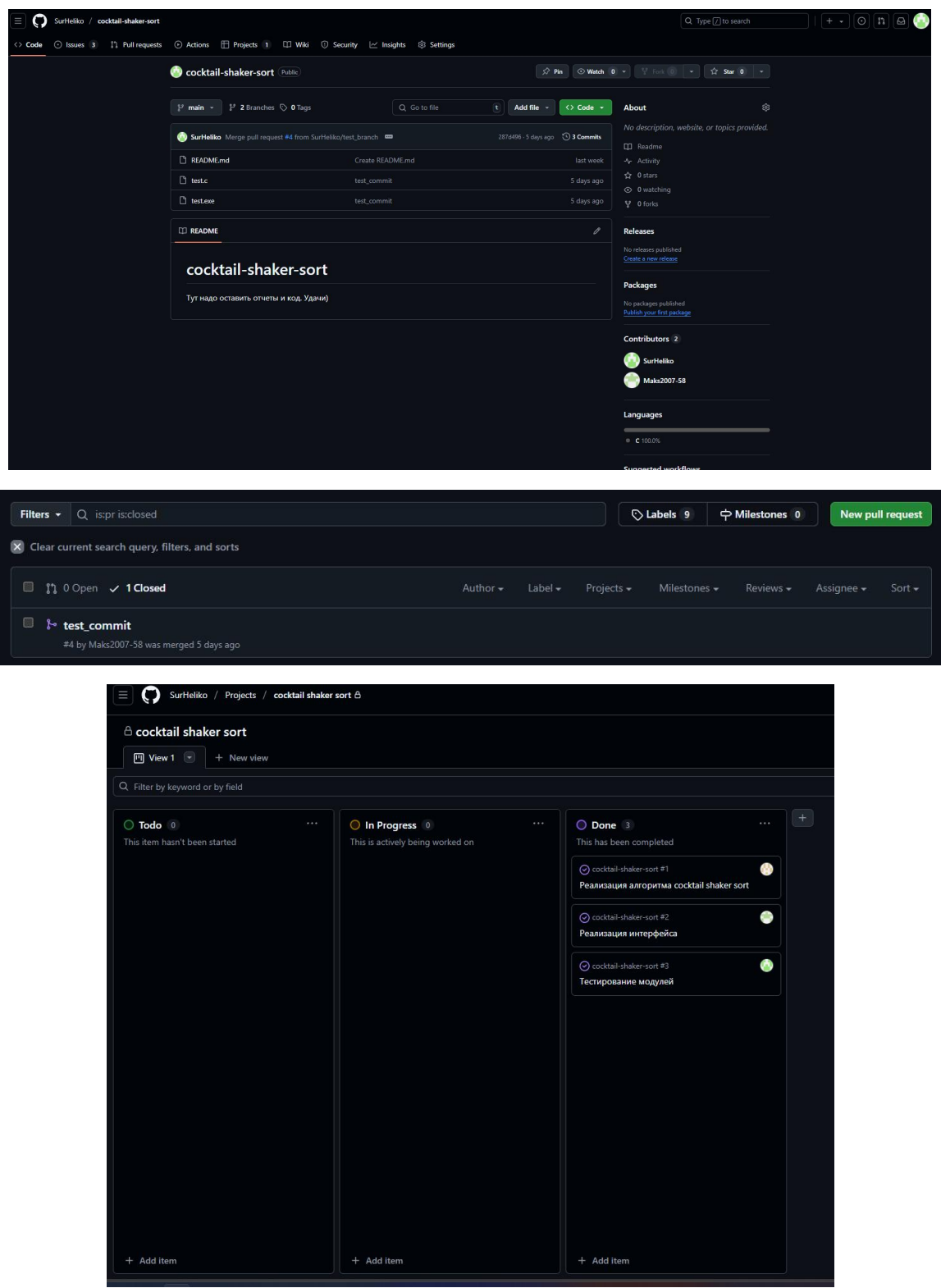
- организация и руководство работой команды, вырабатывая командную стратегию для достижения поставленной цели, организация и корректировка работы команды, в том числе на основе коллегиальных решений;
- способность осуществлять эффективное управление разработкой программных средств и проектов, составления планов, распределения задач, тестирования и оценки качества программных средств.

Цели выполнения учебной практики считаем достигнутыми.

СПИСОК ЛИТЕРАТУРЫ

1. ГОСТ 7.32–2001 «Отчет о научно-исследовательской работе. Структура и правила оформления»;
2. ГОСТ 2.105–95 «Общие требования к текстовым документам»;
ГОСТ Р 7.0.5–2008 «Библиографическая ссылка. Общие требования и правила составления»;
3. ГОСТ 7.1–2003 «Библиографическая запись. Библиографическое описание. Общие требования и правила составления»;
4. ГОСТ 7.80–2000. «Библиографическая запись. Заголовок. Общие требования и правила составления».
5. Корячко В. П., Таганов А. И. Процессы и задачи управления проектами информационных систем. Учебное пособие [Электронный ресурс] : учебное пособие Корячко В. П., Таганов А. И. — М.: Гор. линия-Телеком, 2014. - 376 с
6. Андрианова, А.А. Алгоритмизация и программирование. Практикум [Электронный ресурс] : учебное пособие / А.А. Андрианова, Л.Н. Исмагилов, Т.М. Мухтарова. — Электрон. дан. — Санкт-Петербург : Лань, 2019. — 240 с.
7. Обучение Git [Электронный ресурс] <https://learngitbranching.js.org/>

РАБОТА С GIT



ПРОЕКТ СТУДЕНТОВ ПЕРВОГО КУРСА БАКАЛАВРИАТА
ГРУППЫ 24ВОА2
МАКСИНА С.Н., ПАНЬКИНА М.Д.

Введение

Сортировка данных на сегодняшний день при современном развитии компьютерных технологий является одним из наиболее распространенных процессов современной обработки данных. Задачи на сортировку данных встречаются очень часто в различных профессиональных сферах деятельности.

Алгоритмы сортировки очень широко распространяются практически во всех задачах обработки информации. Они образуют отдельный класс алгоритмов, применяются с целью осуществления последующего более быстрого поиска.

Важность сортировки основана на том факте, что на ее примере можно показать многие основные фундаментальные приемы и методы построения алгоритмов. Сортировка является хорошим примером огромного разнообразия алгоритмов, которые выполняют одну и ту же задачу. Кроме того, многие из них имеют определенные преимущества друг перед другом. За счет усложнения алгоритма можно добиться существенного увеличения эффективности и быстродействия алгоритма по сравнению с более простыми методами. Как правило, термин сортировка понимают, как процесс перестановки объектов некоторого множества в определенном порядке.

Шейкерная сортировка используется редко в реальных приложениях из-за своей неэффективности на больших данных. Она может быть полезна в учебных целях для понимания основных принципов сортировки и структур данных. В реальных приложениях для сортировки массивов обычно применяют более эффективные алгоритмы, такие как быстрая сортировка, сортировка слиянием или сортировка пирамидой.

1 Постановка задачи

Поставленная задача: необходимо заполнить массив из n -ого количества элементов случайными числами. После этого выполнить сортировку вставками над данными, находящимися в массиве, записать отсортированные данные в файл, посчитать время выполнения сортировки.

Использовать сервис GitHub для совместной работы. Создать и выложить коммиты, характеризующие действия, выполненные каждым участником бригады.

Оформить отчет по проведенной практике.

1.1 Достоинства алгоритма шейкерной сортировки

- алгоритм прост в понимании и реализации;
- алгоритм эффективен при работе с небольшими массивами, где большая часть элементов уже находится в правильном порядке;
- алгоритм производит небольшое количество обменов элементов.

1.2 Недостатки алгоритма шейкерной сортировки

- в отличие от более сложных алгоритмов, не является стабильным;
- высокая алгоритмическая сложность $O(n^2)$;
- не рекомендуется для сортировки больших массивов.

1.3 Типичные сценарии применения данного алгоритма

- использование в учебных целях для понимания основных принципов сортировки структур данных.

2 Выбор решения

Для написания данной программы будет использован язык программирования Си. Этот язык является распространённым языком программирования. При разработке языка Си был принят компромисс между низким уровнем языка ассемблера и высоким уровнем других языков. Си – это язык программирования общего назначения, хорошо известный своей эффективностью, экономичностью и переносимостью. Указанные преимущества Си обеспечивают хорошее качество разработки почти любого вида программного продукта.

В качестве среды программирования была выбрана программа Microsoft Visual Studio Code. Microsoft Visual Studio Code — это программная среда по разработке приложений для ОС Windows, как консольных, так и с графическим интерфейсом.

Для удобства совместной разработки был использован сервис GitHub. GitHub — сервис для управления личными и командными проектами. Проект динамично разрабатывается, регулярно расширяя функционал и возможности.

3. Описание программы

При запуске программы выводится сообщение: «Введите размер массива:» и пользователю предлагается ввести количество значений для сортировки.

После того как данные были введены, генерируется массив из случайных чисел, которые выводятся на терминал.

Далее над этими данными выполняется шейкерная сортировка, при которой массив перебирается сначала слева направо, а потом справа налево. При этом сравниваются соседние элементы и меняются местами так, чтобы оказаться в подходящем месте среди ранее упорядоченных элементов. Так происходит до тех пор, пока набор входных данных не будет исчерпан.

После этого отсортированный массив записывается в файл shakerSort.txt.

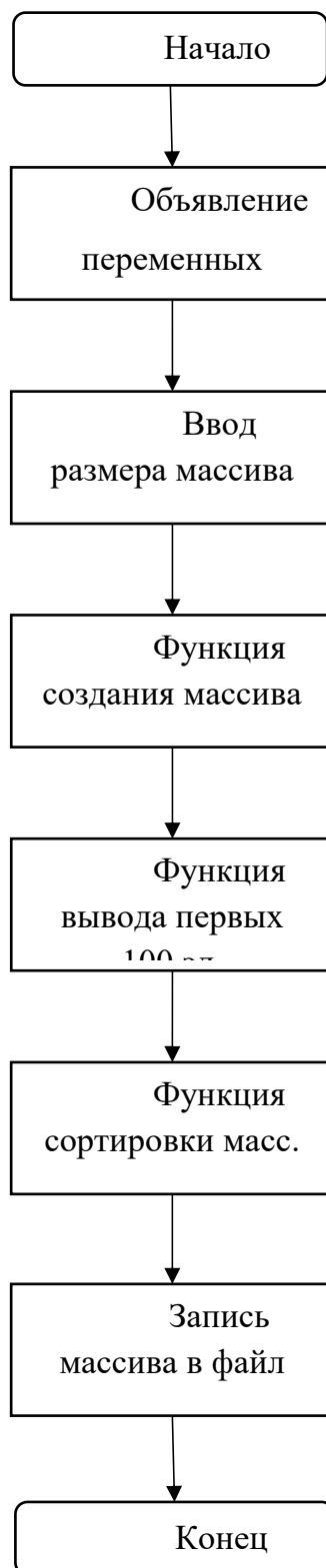
Программа так же осуществляет подсчет времени, затраченного на сортировку.

Подробный алгоритм работы программы и функции сортировки представлен в подразделах 4.1 и 4.2 на рисунке 1.

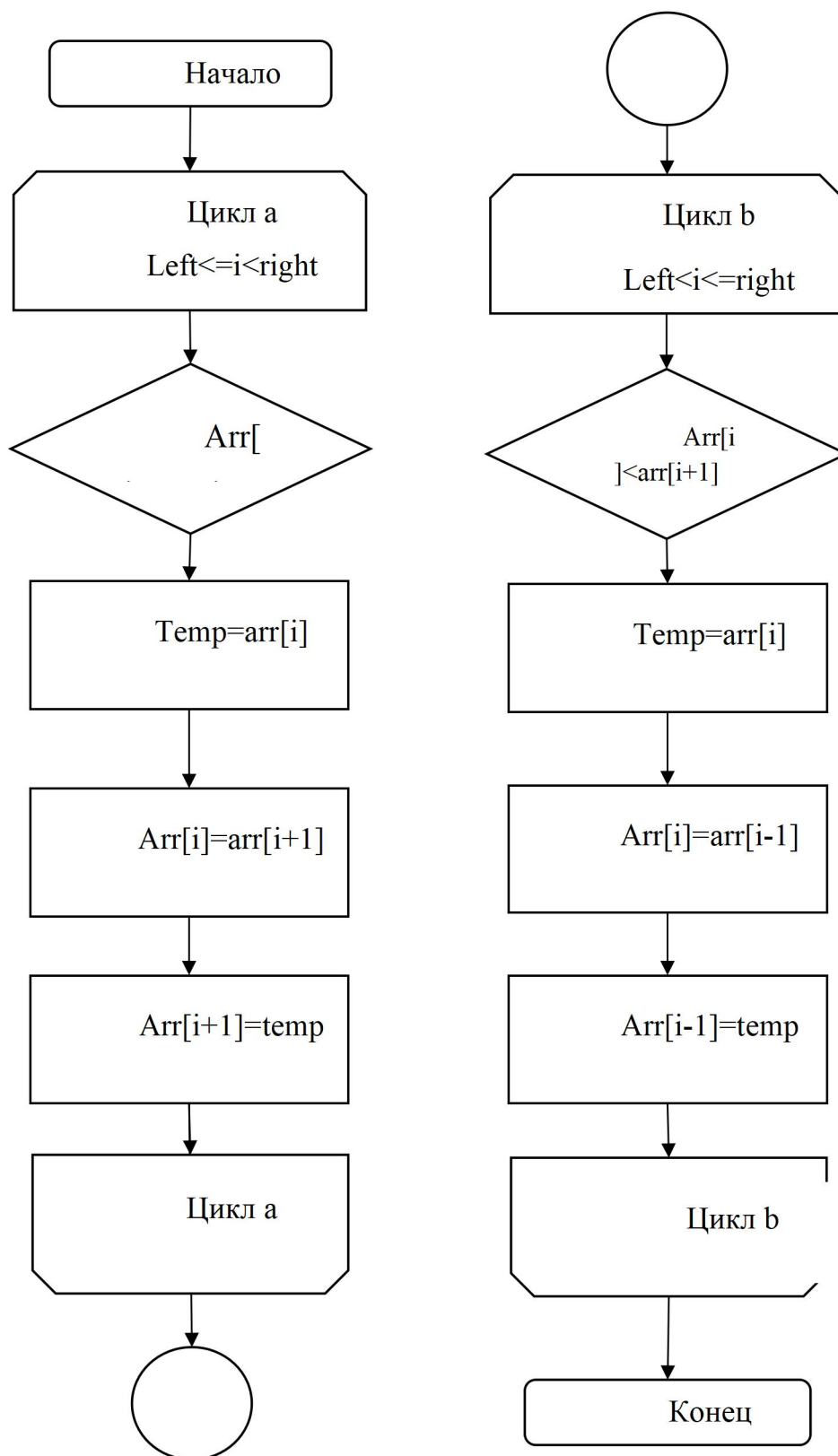
Листинг программы приведен в приложении Б.

4 Схемы программы

4.1 Блок-схема программы



4.2 Блок-схема алгоритма сортировки



5. Тестирование программы

5.1 Тестирование на разных наборах данных

Тестовый набор данных представлен в таблице 1. Результаты тестирования приведены в Приложении А на рисунках А.1-А.11.

Таблица 1 – Тестовый набор данных

	Размер size	массива	Время выполнения сортировки в секундах
1	10000		0.105
2	20000		0.471
3	30000		1.159
4	40000		2.08
5	50000		3.335
6	60000		4.842
7	70000		6.604
8	80000		9.125
9	90000		11.107
10	100000		14.533
11	110000		16.589

5.2 Анализ полученных результатов тестирования (анализ работы алгоритма)

На основании анализа данных, полученных в результате тестирования алгоритма шейкерной сортировки, можно сделать вывод, что время, затраченное на работу программы относительно количества элементов увеличивается линейно, то есть с увеличением количества элементов пропорционально увеличивается время работы программы.

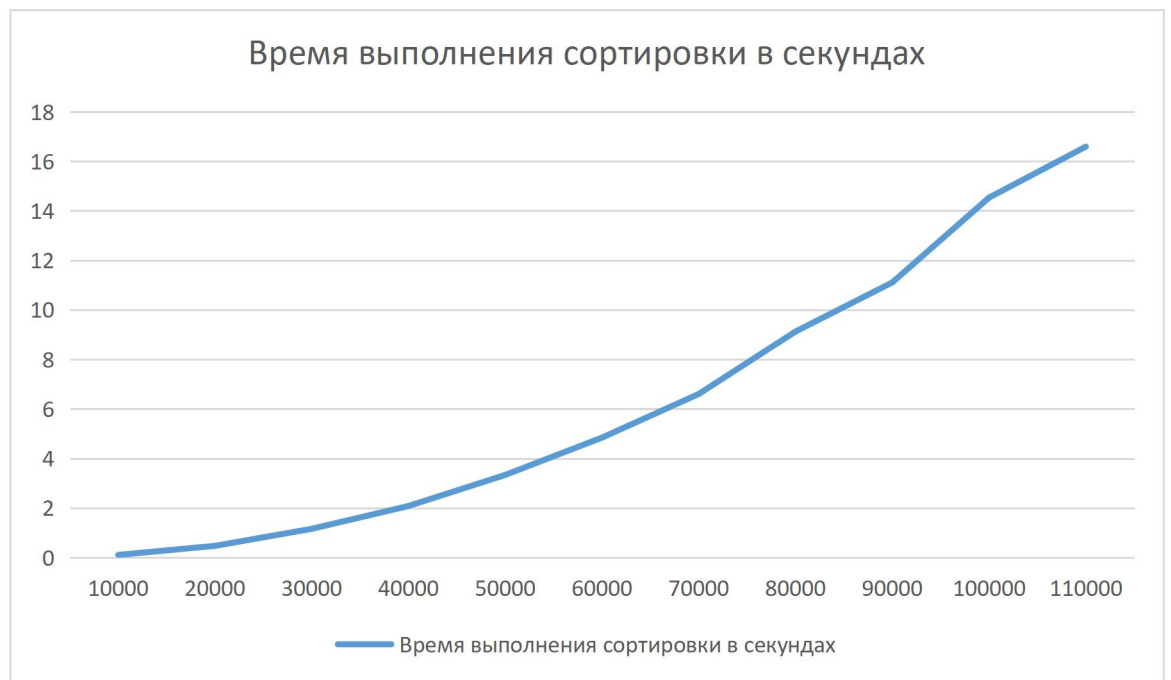


Рисунок 2 – Результаты тестирования

6 Отладка

В качестве среды разработки была выбрана программа Microsoft Visual Studio Code, которая содержит в себе все необходимые средства для разработки и отладки модулей и программ.

Для отладки программы использовались точки останова и пошаговое выполнение кода программы, анализ содержимого локальных переменных. Точки останова – это прерывание выполнения программы, при котором выполняется вызов отладчика. Отладчик является инструментом для поиска и устранения ошибок в программе, с помощью которого можно исследовать состояние программы.

Был использован метод бинарного поиска, он включает в себя разделение частей кода для упрощения процесса отладки. Это может быть особенно полезно, если причина ошибки находится в начале языка программирования, а фактическая ошибка ближе к концу.

Команда шаг с заходом (step into) выполняет следующую инструкцию в обычном пути выполнения программы, а затем приостанавливает выполнение программы, чтобы мы могли проверить состояние программы с помощью отладчика. Если выполняемый оператор содержит вызов функции, шаг с заходом заставляет программу перескакивать в начало вызываемой функции, где она приостанавливается.

Тестирование проводилось в рабочем порядке, в процессе разработки, после завершения написания программы. После завершения написания программы, мною были выявлены и исправлены ошибки.

7 Совместная работа

Во время работы над данной практикой наша бригада осуществляла совместную работу в GitHub.

Определили задачи проекта, назначили приоритет задачам. Разделили роли, назначили исполнителей задачам. Обсуждали выполнение задачи на доске. Корректировали статус задач по мере выполнения. Данная программа была загружена на компьютер, с помощью `git clone` <ссылка>.

Мной был разработан алгоритм шейкерной сортировки и написан изначальный код программы. Я загрузил этот код на репозиторий GitHub.

Ссылка на репозиторий: <https://github.com/SurHeliko/cocktail-shaker-sort>

Заключение

При выполнении данной работы были получены навыки совместной работы с помощью сервисов GitHub, навыки использования программы Git Bash. Был изучен алгоритм сортировки вставками.

Мною был написан алгоритм шейкерной сортировки. Было выполнено тестирование программы на разных наборах данных и отладка данной программы.

При выполнении практической работы были улучшены базовые навыки программирования на языке C. Улучшены навыки отладки, тестирования программ и работы со сложными типами данных.

В дальнейшем программу можно улучшить путем подключения упрощающих реализацию данной сортировки библиотек и улучшения графического интерфейса.

Список используемой литературы

1. ГОСТ 19.701 – 90 Схемы алгоритмов, программ, данных и систем.
2. Керниган, Брайан У., Ритчи, Деннис М. Язык программирования С, 2-е издание.: Пер. с англ. – М., 2009.
3. Шейкерная сортировка [Электронный ресурс] – URL: <https://ru.wikipedia.org>
(дата обращения: 30.06.2025 г)



ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ
ТЕХНИКИ КАФЕДРА «ВЫЧИСЛИТЕЛЬНАЯ
ТЕХНИКА»

УЧЕБНАЯ (ТЕХНОЛОГИЧЕСКАЯ
(ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ))
ПРАКТИКА
(2024/25 учебный год)

ОТЧЕТ
об организации учебной (ознакомительной) практики для
студентов 1 курса кафедры «Вычислительная техника»
учебных групп 24BOA2

Сведения об организаторе:

ФИО	Культиясов Ярослав Кириллович				
Группа	24ВВИм1	Курс	1	Семестр	2
Период прохождения практики			20.06.25		03.07.25
Метод управления проектом			методология Scrum		

Сведения о составе бригад:

№ студента	№ бригады	ФИО студента	Группа
15	2	Максин С.Н.	<u>24BOA2</u>
16	2	Панькин М.Д.	<u>24BOA2</u>
18	5	Силявко М.А.	<u>24BOA2</u>
17	5	Семенов Н.Д.	<u>24BOA2</u>
20	5	Юсупов В.Р.	<u>24BOA2</u>

Сведения о выполнении работы:

№ студента	Роль в бригаде (комментарий)	Демонстрация программы (отметка, дата)	Демонстрация отчета (отметка, дата)	Рек. оценка	Ссылка на репозиторий
	Программная реализация алгоритма	30.06.2025	02.07.2025		https://github.com/SurHeliko/cocktail-shaker-sort.git
	Тестирование и отладка	X	X		нет
	Разработчик графического интерфейса	01.07.2025	03.07.2025		https://github.com/SurHeliko/cocktail-shaker-sort.git

Отзыв о прохождении практики

№ студента	Краткая характеристика обучающегося
15	Работа выполнена в срок и в полном объеме. Имеются недочеты в оформлении отчета.
16	Работа выполнена в срок и в полном объеме. Имеются недочеты в оформлении отчета.

Студент-организатор: Культиясов Я.К. _____ «__» __ 20__ г.
(подпись)

Научный руководитель: Карамышева Н.С. _____ «__» __ 20__ г.
(подпись)