

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

РАБОЧАЯ ПРОГРАММА

МДК.04.01 Распределённые системы управления данными

ПМ.04 Управление и работа с большими данными в системах искусственного интеллекта

программы подготовки специалистов среднего звена

специальность

09.02.13 Интеграция решений с применением технологий искусственного интеллекта

Форма обучения - очная

Владивосток 2026

Рабочая программа МДК.04.01 Распределённые системы управления данными разработана на основе федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 Интеграция решений с применением технологий искусственного интеллекта, квалификация специалист по работе с искусственным интеллектом, утвержденного приказом Министерства просвещения Российской Федерации от 24 декабря 2024 г. № 1025

Разработчик(и): П.В. Калашников, преподаватель

Рассмотрено и одобрено на заседании цикловой методической комиссии
Протокол № 9 от «20 » мая 2026 г.

Председатель ПЦК  П.В. Калашников

СОДЕРЖАНИЕ

- 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ
УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ
ДИСЦИПЛИНЫ**
- 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ МЕЖДИСЦИПЛИНАРНОГО КУРСА

1.1. Цель и место междисциплинарного курса в структуре образовательной программы

Междисциплинарный курс МДК.04.01 Распределённые системы управления данными входит в профессиональный цикл учебного плана образовательной программы в структуру профессионального модуля ПМ.04 «Управление и работа с большими данными в системах искусственного интеллекта».

Цель: формирование у обучающихся профессиональных компетенций, необходимых для проектирования, разработки, тестирования и документирования программных модулей, предназначенных для реализации алгоритмов машинного обучения и других компонентов систем искусственного интеллекта.

1.2. Планируемые результаты освоения междисциплинарного курса

По итогам освоения курса, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

Код	Наименование видов деятельности и профессиональных компетенций
ВД 4	Управление и работа с большими данными в системах искусственного интеллекта
ПК 4.1	Осуществлять подбор и настройку готовых моделей искусственного интеллекта
ПК 4.2	Разрабатывать и настраивать сценарии и процесс обучения
ПК 4.3	Проводить оценку эффективности обученных моделей

В результате освоения междисциплинарного курса обучающийся должен:

Код ПК	Уметь	Знать	Владеть навыками
ПК 4.1	проектировать и развертывать отказоустойчивые системы для работы с Big Data с использованием сервисов облачных платформ разрабатывать эффективный код на Python и SQL, обеспечивающий высокую производительность обработки данных.	принципы архитектуры отказоустойчивых Big Data-систем, методы обеспечения надежности и возможности облачных платформ для масштабирования инфраструктуры особенности написания эффективного кода на Python и SQL для обработки больших данных	Проектирование и развертывание отказоустойчивых систем для работы с Big Data
ПК 4.2	Проектировать и реализовывать аналитические системы	архитектуру аналитических систем и рекомендательных	Проектирование и реализация аналитических систем и рекомендательных

	и рекомендательные сервисы с применением алгоритмов машинного обучения использовать Python и Apache Spark для обработки больших данных, обучения моделей и внедрения их в производственную среду.	сервисов, алгоритмы машинного обучения для персонализации и прогнозирования возможности экосистемы Python и фреймворка Apache Spark (PySpark, MLlib) для распределенной обработки больших данных	сервисов с применением алгоритмов машинного обучения
ПК 4.3	Проектировать и настраивать конвейеры потоковой обработки данных с использованием очередей сообщений для обеспечения надёжности, масштабируемости и низкой задержки реализовывать компоненты-производители и потребители сообщений с корректной обработкой ошибок, повторных доставок и гарантиями подтверждения приёма данных	принципы архитектуры потоковой обработки данных (event-driven, publish-subscribe) и механизмы работы очередей сообщений (гарантии доставки, идемпотентность, балансировка нагрузки) возможности и особенности применения популярных платформ для работы с потоками данных и очередями (Apache Kafka, RabbitMQ, Apache Pulsar, Redis Streams)	Проектирование и настройка конвейеров потоковой обработки данных с использованием очередей сообщений

1. СТРУКТУРА И СОДЕРЖАНИЕ МЕЖДИСЦИПЛИНАРНОГО КУРСА

2.1 Объем МДК и виды учебной работы

Вид учебной работы	Объем в часах	В т.ч. в форме практ. подготовки
Учебные занятия	82	48
Самостоятельная работа	36	
Консультации	2	
Промежуточная аттестация в форме экзамена	4	

Всего	124	48
-------	-----	----

2.2. Тематический план и содержание МДК

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы
1	2	3	4
Раздел 1	Распределённые системы управления данными	118	
Тема 1.1. Архитектура распределённых систем данных и CAP-теорема	Содержание учебного материала Понятие распределённой системы данных. Принципы горизонтального масштабирования и sharding. CAP-теорема: консистентность, доступность, устойчивость к разделению. BASE vs ACID. Архитектурные паттерны: shared-nothing, репликация. Обзор экосистемы Big Data в контексте ИИ-проектов.	4	ПК 4.1
	Практические занятия № 1. Сравнительный анализ архитектур PostgreSQL (ACID) и Cassandra (BASE). Моделирование CAP-ограничений на примере распределённого кластера. Установка стендового окружения: Docker Compose с HDFS, Spark, Kafka. Проверка отказоустойчивости при имитации сетевого разделения.	8	
Тема 1.2 Apache Hadoop: HDFS и YARN	Содержание учебного материала Архитектура HDFS: NameNode, DataNode, блоки, репликация, rack awareness. YARN: Resource Manager, Node Manager, ApplicationMaster. MapReduce: принцип Map/Reduce, фазы shuffle и sort. Форматы файлов: ORC, Parquet, Avro — сравнение для ИИ-нагрузок. Balancer и квоты HDFS.	4	ПК 4.1 ПК 4.2 ПК 4.3
	Практические занятия № 2 Развёртывание Hadoop-кластера в Docker (3 DataNode). Работа с HDFS: hdfs dfs — mkdir, put, get, ls, du. Запуск MapReduce-задания WordCount и анализ контрсчётчиков. Web UI HDFS и YARN: мониторинг блоков и задач. Загрузка тренировочного датасета в HDFS, проверка репликации.	6	
Тема 1.3 Apache Spark: RDD, DataFrame и Spark SQL	Содержание учебного материала Архитектура Spark: Driver, Executor, DAG-планировщик, стадии и задачи. Типы данных: RDD, Dataset, DataFrame. Ленивые вычисления, трансформации и действия. Spark SQL: синтаксис, Catalyst-оптимизатор, Tungsten-движок. MLlib: пайплайн API. Сравнение Spark и MapReduce по модели вычислений.	4	ПК 4.1 ПК 4.2
	Практическое занятие № 3 Разработка PySpark-приложений: обработка датасета через DataFrame API.	8	

	Spark SQL-запросы к данным через Hive Metastore. Оптимизация через explain() и broadcastJoin. Чтение и запись Parquet-файлов из HDFS. Анализ Spark UI: DAG, Stage view, Storage.		
Тема 1.4. Оптимизация и конфигурирование Apache Spark	Содержание учебного материала Тюнинг Spark: spark.executor.cores, spark.executor.memory, spark.default.parallelism. Кэширование: persist(), cache(), уровни StorageLevel. Проблемы производительности: data skew, excessive shuffle, OOM. Конфигурация Spark для GPU-ускоренных задач обучения ИИ. Spark History Server и диагностика.	4	ПК 4.1 ПК 4.2
	Практическое занятие № 4. Профилирование Spark-задания через Spark UI: анализ Stage, Task, Shuffle Read/Write. Устранение data skew методом salting. Настройка параметров executor для обработки обучающего датасета ИИ. Сравнительный тест производительности: с кэшированием и без..	8	
Тема 1.5. Apache Hive и медальонная архитектура Data Lake	Содержание учебного материала Apache Hive: архитектура, HiveQL, Metastore. Концепция Data Lake vs Data Warehouse. Партиционирование и бакетирование таблиц Hive. Медальонная архитектура: зоны Bronze / Silver / Gold. Apache Iceberg и Delta Lake: ACID-транзакции и time-travel для Data Lake. Применение в Feature Store ИИ-систем.	4	ПК 4.1 ПК 4.2
	Практическое занятие № 5. Создание партиционированных Hive-таблиц поверх HDFS-данных. HiveQL-запросы для агрегации обучающих данных по классам/меткам. Организация тренировочного датасета по зонам Bronze→Silver→Gold с трансформациями очистки.	4	
	Самостоятельная работа обучающихся: подготовка докладов на тему «Методы разработки медальонной архитектуры»	18	ПК 4.1 ПК 4.2
Тема 1.6 Apache Flink и потоковая обработка данных	Содержание учебного материала Архитектура Flink: JobManager, TaskManager, DAG-задание. Event Time vs Processing Time vs Ingestion Time. Ключевые операции: map, flatMap, keyBy, window (tumbling/sliding/session). Checkpoint и Savepoint. Сравнение Spark Structured Streaming и Flink: задержка, семантика, применение в онлайн-инференсе ИИ.	4	
	Практическое занятие № 6. Развёртывание Flink-кластера. Разработка streaming-задания для обработки событий инференса в реальном времени. Оконные агрегации с Event Time и watermarks. Настройка checkpointing для отказоустойчивости. Тест производительности при пиковой нагрузке.	6	
Тема 1.7 ClickHouse — колоночная СУБД для	Содержание учебного материала Архитектура ClickHouse: MergeTree-семейство движков, репликация,		

аналитических нагрузок	ClickHouse Keeper. Партиционирование, первичный ключ, индексы разреженного типа. Материализованные представления. Оконные функции и агрегации. Сравнение ClickHouse с другими OLAP СУБД. Применение для хранения метрик экспериментов ИИ и логов инференса		
	Практическое занятие № 7. Развёртывание ClickHouse-кластера с репликацией (2 шарда × 2 реплики). Проектирование схемы для хранения логов инференса ML-модели. Партиционирование по дате события. Нагрузочный тест: сравнение скорости аналитических SELECT-запросов с PostgreSQL.	4	
Тема 1.8 MinIO и объектное хранение артефактов ИИ	Содержание учебного материала Объектное хранение: концепция, S3 API-совместимость. MinIO: архитектура, erasure coding, версионирование, bucket policy, шифрование (SSE-S3/SSE-KMS). Интеграция с MLflow как artifact store. Использование MinIO с DVC, LakeFS для версионирования датасетов. Жизненный цикл объектов: tiering, lifecycle rules. Мультиарендность в MinIO Operator.	4	
	Практическое занятие № 8. Развёртывание MinIO в режиме Erasure Coding. Работа с boto3/minio SDK: загрузка датасетов, моделей, артефактов обучения. Интеграция MinIO с MLflow как artifact store. Настройка политик доступа IAM, lifecycle rules и bucket notifications.	4	
Тема 1.9 Оркестрация рабочих процессов данных: Apache Airflow	Содержание учебного материала Apache Airflow: архитектура, DAG, Operator, Executor, XCom. TaskFlow API. Планирование и зависимости задач. Паттерны: dynamic DAG, sensor, branching. Интеграция с Spark, Kubernetes. Airflow vs Prefect vs Dagster: сравнение для ML-пайплайнов. Мониторинг и обработка повторных попыток.	4	
	Практическое занятие № 9. Разработка DAG для ML-пайплайна: извлечение данных → предобработка в Spark → обучение → сохранение модели в MinIO. Настройка Connections, Variables, XCom. Реализация SLA и email-уведомлений. Мониторинг через Airflow UI и Grafana.	4	
	Самостоятельная работа обучающихся: подготовка докладов на тему «Методы оркестрация рабочих процессов данных»	18	
Консультации		2	
Промежуточная аттестация в форме экзамена (6-й семестр)		4	
Всего		124	

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

3. УСЛОВИЯ РЕАЛИЗАЦИИ МЕЖДИСЦИПЛИНАРНОГО КУРСА

3.1. Материально-техническое обеспечение

Для реализации программы МДК.04.01 Распределённые системы управления данными предусмотрено наличие следующих помещений:

Лаборатория организации и принципов построения информационных систем.

Основное оборудование: Компьютеризированные рабочие места обучающихся с выходом в сеть "Интернет" - 20 шт., стулья - 20 шт., комплект мебели (стол и стул) для преподавателя - 1 шт., графические станции ПК №2 WorkStation WS 917 20 шт., автоматизированное рабочее место преподавателя ПК №2 WorkStation WS 917, графический планшет Wacom - 20 шт., мультимедийное оборудование - 1 шт., телевизор, экран, многофункциональное устройство (МФУ) формат А3 -1 шт., доска маркерная - 1 шт.

Программное обеспечение: PyCharm, Visual Studio Code, IntelliJ IDEA, JetBrains Rider, Python (Anaconda3) с TensorFlow/PyTorch, Microsoft Visio, Draw.io, Figma, Figma, Adobe Illustrator, MySQL Workbench, HeidiSQL, MySQL, PostgreSQL, Git, Docker Desktop, Microsoft Project, Draw.io, Microsoft SQL Server Management Studio, Microsoft Office..

Основное оборудование: Рабочие места на базе компьютерной техники с возможностью подключения к информационно-телекоммуникационной сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду ВВГУ: комплекты учебной мебели (столы и стулья) – 19 шт., персональные компьютеры (облачные мониторы) - 19 шт; доска маркерная - 1шт., лазерный копир-принтер-сканер с 2 лотками Xerox WorkCentre 3345 DNI.

Программное обеспечение: Microsoft Windows 7 Professional; Microsoft Office ProPlus 2010 Russian Acdmc; СПС КонсультантЮрист: Версия Проф; Adobe Acrobat Reader; Google Chrome; Adobe Flash Player; 7-Zip 18.01 (x64).

3.2. Учебно-методическое обеспечение

Для реализации программы учебной дисциплины библиотечный фонд ВВГУ укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

Основные источники

1. Нестеров, С. А. Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. — 2-е изд. — Москва : Издательство Юрайт, 2026. — 258 с. — (Профессиональное образование). — ISBN 978-5-534-18087-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587742>

2. Стружкин, Н. П. Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2026. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/587735>

3. Стасышин, В. М. Базы данных: технологии доступа: учебник для среднего профессионального образования / В. М. Стасышин, Т. Л. Стасышина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2026. — 164 с. — (Профессиональное образование). — ISBN 978-5-534-09888-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/586799>

4. Гордеев, С. И. Организация баз данных : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2026. — 691 с. —

(Высшее образование). — ISBN 978-5-534-21115-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/590130>

Дополнительные источники

1. Толстобров, А. П. Управление данными : учебник для вузов / А. П. Толстобров. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2026. — 272 с. — (Высшее образование). — ISBN 978-5-534-14162-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/5885222>.
2. Казарин, О. В. Программно-аппаратные средства защиты информации. Защита программного обеспечения : учебник и практикум для среднего профессионального образования / О. В. Казарин, А. С. Забабурин. — Москва : Издательство Юрайт, 2026. — 312 с. — (Профессиональное образование). — ISBN 978-5-534-13221-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/588246>.

Электронные ресурсы

1. Информационно-справочная система «Консультант Плюс» — <http://www.consultant.ru/>
2. Профессиональная база данных: "Открытая база ГОСТов"/ Режим доступа: <http://standartgost.ru/>, доступ свободный
3. Электронная библиотечная система «РУКОНТ» - Режим доступа: <https://lib.rucont.ru/>
4. Электронная библиотечная система ZNANIUM.COM - Режим доступа: <https://znanium.com/>
5. Электронно-библиотечная система издательства "Юрайт" - Режим доступа: <https://urait.ru/>
6. Электронно-библиотечная система издательства "Лань" - Режим доступа: <https://e.lanbook.com/>
7. Президентская библиотека им. Б.Н. Ельцина (база данных различных профессиональных областей) Режим доступа: <https://www.prilib.ru/>, доступ свободный

4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ И РЕАЛИЗАЦИИ МЕЖДИСЦИПЛИНАРНОГО КУРСА

Начиная изучение дисциплины, студенту необходимо:

- ознакомиться с программой, изучить список рекомендуемой литературы;
- внимательно разобраться в структуре курса, в системе распределения учебного материала по видам занятий, формам контроля, чтобы иметь представление о курсе в целом.

Самостоятельная работа студентов (СРС) — это деятельность учащихся, которую они совершают без непосредственной помощи и указаний преподавателя, руководствуясь сформировавшимися ранее представлениями о порядке и правильности выполнения операций. Цель СРС в процессе обучения заключается, как в усвоении знаний, так и в формировании умений и навыков по их использованию в новых условиях на новом учебном материале. Самостоятельная работа призвана обеспечивать возможность осуществления студентами самостоятельной познавательной деятельности в обучении, и является видом учебного труда, способствующего формированию у студентов самостоятельности. Кроме того, для расширения и углубления знаний по дисциплине целесообразно использовать: публикации в тематических журналах; полнотекстовые базы данных библиотеки; имеющиеся в библиотеках вуза. Успешное освоение дисциплины предполагает активную работу студентов на всех занятиях аудиторной формы: лекций и практических занятий, выполнение аттестационных мероприятий, эффективную самостоятельную работу.

4.1 Методические рекомендации обучающимся по обеспечению самостоятельной работы

Самостоятельная работа проводится с целью: систематизации, закрепления, углубления и расширения теоретических знаний и практических умений, приобретаемых студентами в ходе аудиторных занятий; формирования умений использовать специальную литературу; развития познавательных способностей и активности обучающихся; формирования самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации; развития исследовательских умений.

Для лучшего освоения материала и систематизации знаний по дисциплине необходимо отвечать на вопросы для самоконтроля. Во время самостоятельной проработки лекционного материала особое внимание следует уделять возникшим вопросам, непонятным терминам, спорным точкам зрения. В случае необходимости обращаться к преподавателю за консультацией. Студент должен четко уяснить, что именно с лекции начинается его подготовка к практическому занятию. Вместе с тем, лекция лишь организует мыслительную деятельность, но не обеспечивает глубину усвоения программного материала.

При подготовке к практическому занятию особое внимание необходимо обратить на содержание основных положений и выводов, объяснение явлений и фактов, уяснение практического приложения рассматриваемых теоретических вопросов. В процессе этой работы студент должен стремиться понять и запомнить основные положения рассматриваемого материала, примеры, поясняющие его, а также разобраться в иллюстративном материале. В процессе подготовки рекомендуется взаимное обсуждение материала, во время которого закрепляются знания, а также приобретает практика в изложении и разъяснении полученных знаний, развивается речь.

При необходимости следует обращаться за консультацией к преподавателю. Перед консультацией, необходимо хорошо продумать вопросы, которые требуют разъяснения.

В процессе изучения дисциплины студенту необходимо ориентироваться на самостоятельную проработку лекционного материала, подготовку к практическим занятиям, выполнение тестов, кейсовых заданий, самостоятельное изучение некоторых разделов курса. Для проведения занятий лекционного типа используются учебно-наглядные пособия в форме презентационных материалов, обеспечивающих тематические иллюстрации, соответствующие темам лекций.

4.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

5 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ КУРСА

Код ПК	Критерии оценки результата (показатели освоённости компетенций)	Формы контроля и методы оценки
ПК 4.1	Оценка "отлично": алгоритмы разработаны в полном соответствии с ТЗ, оптимизированы и понятны. Оценка "хорошо": алгоритмы разработаны в соответствии с ТЗ, но допускают незначительные отклонения.	Защита отчёта с демонстрацией разработанных алгоритмов и устное собеседование.

	Оценка "удовлетворительно": алгоритмы разработаны с частичным соответствием ТЗ.	
ПК.4.2	Оценка "отлично": программные модули разработаны в полном соответствии с ТЗ, тесты проходят успешно. Оценка "хорошо": программные модули разработаны с минимальными несоответствиями, тесты в целом успешны. Оценка "удовлетворительно": программные модули разработаны с существенными доработками.	Защита отчёта по разработанным модулям, проверка выполнения ТЗ и прохождения тестов.
ПК 4.3	Оценка "отлично": алгоритмы разработаны в полном соответствии с ТЗ, оптимизированы и понятны. Оценка "хорошо": алгоритмы разработаны в соответствии с ТЗ, но допускают незначительные отклонения. Оценка "удовлетворительно": алгоритмы разработаны с частичным соответствием ТЗ.	Защита отчёта с демонстрацией разработанных алгоритмов и устное собеседование.

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

для проведения текущего контроля и промежуточной аттестации

по **МДК.04.01 Распределённые системы управления данными**

ПМ.04 Управление и работа с большими данными в системах
искусственного интеллекта

программы подготовки специалистов среднего звена

**09.02.13 Интеграция решений с применением технологий искусственного
интеллекта**

Форма обучения: очная

Владивосток 2026

1. Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу МДК.04.01 «Распределённые системы управления данными» в соответствии с техническим заданием и оценка их качества

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта (с использованием оценочного средства - устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных заданий, тестирование и т.д.).

2. Планируемые результаты обучения, обеспечивающие результаты освоения образовательной программы

Код ПК	Код результата обучения	Наименование результата обучения
ПК 4.1	31	Основные методы и подходы к построению алгоритмов (типичные поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы).
	32	Принципы эффективной обработки данных
	У1	Анализировать технические задания и выявлять требования к алгоритмам.
	У2	Разрабатывать оптимальные алгоритмы для решения задач в области ИИ.
	П1	Разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ.
	П2	Использования библиотек и инструментов для работы с алгоритмами и данными (например: Pandas, NumPy, Scikit-learn).
ПК 4.2	31	Принципы модульного программирования.
	32	Стандартные фреймворки и библиотеки для работы с ИИ.
	У1	Реализовывать программные модули на основе требований технического задания.
	У2	Соблюдать при разработке принципы «чистого кода».
	У3	Использовать стандартные библиотеки и фреймворки для ускорения разработки.
	П1	Разработки модульных ИИ-систем, соответствующих требованиям производительности и безопасности.
	П2	Внедрения разработанных ИИ-модулей в комплексные программные системы.
	П3	Оптимизации кода и работы с интерфейсами для взаимодействия между модулями.
ПК 4.3	31	Основные методы и подходы к построению алгоритмов (типичные поисковые алгоритмы, жадные алгоритмы, динамическое программирование, рекурсивные подходы).
	32	Принципы эффективной обработки данных
	У1	Анализировать технические задания и выявлять требования к алгоритмам.
	У2	Разрабатывать оптимальные алгоритмы для решения задач в области ИИ.
	П1	Разработки, оптимизации и оценки сложности алгоритмов для ИИ-программ.
	П2	Использования библиотек и инструментов для работы с

Код ПК	Код результата обучения	Наименование результата обучения
		алгоритмами и данными (например: Pandas, NumPy, Scikit-learn).

3. Описание процедуры оценивания

Результаты обучения по модулю, уровень сформированности компетенций оцениваются по четырём бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» (по бальной системе. Максимальная сумма баллов по дисциплине равна 100 баллов.)

Текущая аттестация по модулю проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по модулю результатов обучения, обеспечивающих результаты освоения образовательной программы в целом. Оценка на зачете / экзамене выставляется с учетом оценок, полученных при прохождении текущей аттестации.

Критерии оценивания устного ответа

(оценочные средства: контрольная работа, собеседование, устное сообщение.)

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

(оценочные средства: реферат, конспект, контрольная работа, письменный отчет по практической работе, портфолио, доклад (сообщение), в том числе выполненный в форме презентации, творческое задание, курсовая работа).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы. Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии оценивания тестового задания

Оценка	Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
Количество правильных ответов	91 % и $\geq$	от 81% до 90,9 %	не менее 70%	менее 70%

Критерии выставления оценки студенту на зачете/ экзамене

(оценочные средства: устный опрос в форме ответов на вопросы билетов, устный опрос в форме собеседования, выполнение письменных разноуровневых задач и заданий, комплексная расчетно-графическая работа, творческое задание, кейс-задача, портфолио, проект и т.п.)

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними

	навыками и приемами выполнения практических задач.
«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

4. Примеры оценочных средств для проведения промежуточной аттестации

4.1 Вопросы для собеседования (устного опроса):

1. Принципы создания и функционирования распределённых систем.
2. Классификация подходов к построению распределённой информационной системы: по месту обработки данных, по архитектуре, по нахождению необходимой функциональности.
3. Понятие прикладных протоколов в распределённых системах.
4. Что такое распределённая система управления базами данных (РСУБД).
5. Требования, лежащие в основе распределённых баз данных (локальная автономия, независимость от центрального узла, непрерывное функционирование и др.).
6. Локальная автономия узлов в распределённой системе: сущность и значение.
7. Независимость от расположения, фрагментации и репликации в распределённых БД.
8. Обработка распределённых запросов: особенности и задачи.
9. Управление распределёнными транзакциями: основные проблемы и подходы.
10. Аппаратная, операционная и сетевая независимость в распределённых СУБД.
11. Архитектурные модели распределённых систем: клиент-сервер, многоуровневая архитектура, peer-to-peer.
12. Методы обеспечения отказоустойчивости и высокой доступности в распределённых системах.
13. Синхронизация времени и согласованность данных в распределённых системах.
14. Протоколы двухфазной и трёхфазной фиксации транзакций.
15. Модели консистентности данных в распределённых хранилищах.
16. Репликация данных: виды, преимущества и недостатки.
17. Фрагментация данных: горизонтальная, вертикальная, смешанная — цели и реализация.
18. Безопасность и разграничение доступа в распределённых информационных системах.
19. Примеры современных распределённых СУБД и их архитектурные особенности.
20. Проблемы масштабируемости распределённых систем и пути их решения

4.2 Примеры тестовых заданий

ТЕСТ № 1

Производится пузырьковая сортировка массива из 8 элементов, заполненного равномерно случайными числами. Сколько будет выполнено перестановок?

- a) 8
- b) 56
- c) 14

- d) 28
- e) 9

2) В процессе сортировки сравниваются соседние элементы. По какому алгоритму выполняется эта сортировка?

- a) Пузырьковая
- b) Вставками
- c) Быстрая
- d) Шелла
- e) Отбором

3) Из каких позиций очереди можно извлекать элементы?

- a) Только из конца очереди
- b) Только из начала очереди
- c) Только из начала или конца очереди
- d) Из любой позиции, кроме конца очереди
- e) Из любой позиции

4) В двоичное дерево поиска заносились упорядоченные по убыванию символы. Результат просмотра в нисходящем порядке следующий: F, E, D, C, B, A. Какое дерево получилось?

- a) Байера
- b) Красно-чёрное
- c) Мультивариантное
- d) Сбалансированное
- e) Вырожденное

5) Какая структура данных используется для сохранения и восстановления содержимого регистров общего назначения центрального процессора при вызове процедур?

- a) Список
- b) Таблица
- c) Очередь
- d) Двоичное дерево
- e) Стек

6) На какой сортировке основана сортировка Шелла?

- a) Бэтчера
- b) Перестановками
- c) Вставками
- d) Отбором
- e) Хоара

7) Основное требование, предъявляемое к массиву для возможности выполнения двоичного поиска:

- a) Малый размер

- b) Неупорядоченность
- c) Упорядоченность
- d) Большой размер
- e) Нет особых требований

8) Имеется некоторая структура данных, в которую заносятся упорядоченные по убыванию целые числа. Считывание данных из этой структуры даёт результат: 1, 3, 5, 7, 9, 11. Чем является эта структура данных?

- a) Дерево
- b) Связный список
- c) Граф
- d) Очередь
- e) Стек

9) Каким выражением определяется количество сравнений для пузырьковой сортировки?

- a) N^2
- b) N
- c) $N(N-1)/2$
- d) $(N-1)/2$
- e) $N-1$

10) Некоторый массив размером N был отсортирован за время, пропорциональное $\log^2 N$. По какому алгоритму выполнялась сортировка?

- a) Перестановками
- b) Хоара
- c) Шелла
- d) Отбором
- e) Вставками

11) Имеется двоичное дерево (не являющееся деревом поиска), содержащее произвольные символы. Нисходящий просмотр дерева даёт следующий результат: A, a, +, *, 1, \$, x. Какой узел является корнем дерева?

- a) x
- b) 1
- c) A
- d) *
- e) +

12) Имеется неупорядоченный массив целых чисел. Для нахождения ключа используется последовательный поиск. Гарантируется ли в этом случае истинность результата поиска?

- a) Да
- b) Нет
- c) Гарантируется при условии, что значение ключа не превышает размера массива
- d) Гарантируется при условии, что в процедуре поиска используется цикл while

e) Гарантируется при условии, что в процедуре поиска используется цикл for

13) Какие позиции списка с выделенным ведущим звеном доступны для занесения новых элементов (при условии, что используются наиболее простые и унифицированные процедуры работы со списком)?

- a) Все позиции, кроме ведущего и последнего звена
- b) Все позиции
- c) Все позиции, кроме ведущего звена
- d) Только ведущее звено
- e) Только последнее звено

14) Имеется идеально сбалансированное двоичное дерево поиска, содержащее целые числа. Просмотр дерева даёт следующий результат: 2, 4, 6, 8, 10, 12, 14. Какой способ просмотра дерева использовался?

- a) Поуровневый
- b) Пузловой
- c) Последовательный
- d) Нисходящий
- e) Восходящий

15) Имеется неупорядоченный массив целых чисел из 10 элементов. Сколько операций сравнения потребуется для установления факта отсутствия искомого элемента в этом массиве?

- a) 0
- b) 10
- c) 5
- d) 1
- e) 9

16) В процессе сортировки выполняется поиск наименьшего элемента. По какому алгоритму выполняется эта сортировка?

- a) Отбором
- b) Пузырьковая
- c) Вставками
- d) Быстрая
- e) Шелла

17) Показанная на рисунке структура данных является ...

- a) 1-связным линейным списком
- b) Стеком
- c) 2-связным линейным списком
- d) 1-связным кольцевым списком
- e) 2-связным кольцевым списком

18) Каким выражением определяется количество перестановок для пузырьковой сортировки в среднем случае?

- a) $N(N-1)/2$
- b) $N-1$
- c) $N(N-1)/4$

- d) 0
e) N2

19) Имеется упорядоченный массив целых чисел из 9 элементов. Сколько операций сравнения потребуется при двоичном поиске для нахождения искомого ключа, если он находится в точно в середине массива?

- a) 1
b) 0
c) 8
d) 5
e) 9

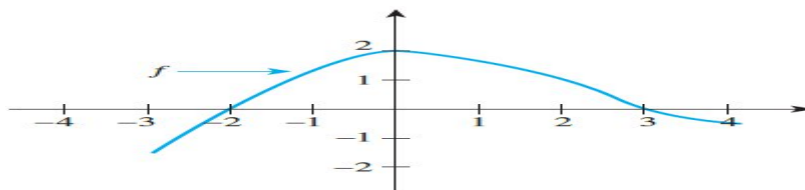
20) Производится пузырьковая сортировка массива из 6 элементов, причём массив уже упорядочен в требуемом порядке. Сколько будет выполнено перестановок?

- a) 30
b) 15
c) 6
d) 7
e) 0

5.4 Примеры заданий для практической работы

Практическая работа № 1

Для функции заданной графиком определить:



- А) $f(0) > 0$?
Б) Для каких значений x выполняется $f(x) > 0$?
В) Приблизительно найдите значения для $x_1 \neq x_2$ такие, что $f(x_1) = f(x_2) = 1,5$.
Г) Возрастает или убывает функция на интервале от 2 до 3?
Д) Возрастает или убывает функция на интервале от 0 до 4?

Задача 2.

- А) Построить график функции $y = 2 \lfloor x \rfloor$ для каждого действительного числа x
Б) Найти $y(3,5)$, $y(5,1)$
В) Найти знак выражения $y(2,1)$, $y(2,7)$.
Г) Определить значения аргумента x , при котором $y(x) > 0$
Д) Найти нули функции

Задача 3

Проверить истинность утверждения

- А) $n = o(n^2)$; В) $n^2 - 2 = o(n^4)$ Д) $10n + 100 = O(30n)$
Б) $n + 3 = o(n^3)$ Г) $5n + 100 = O(25n)$

Задача 4

Определить порядок сложности алгоритма по фрагменту кода

- А)
p:=0, a:=2
for i:=2 to n

```

p:=(p-i)*a
next i
Б)
for i:=  $\lfloor \frac{n}{2} \rfloor$  to n
a=n+i
next i

```

Практическая работа № 2

Задача 1

Найти наименьший корень рекурсивной функции

$f(a_1, a_2, \dots, a_n, y) = (x_1 + x_2 - x_3 - y)$ заданной алгоритмом

- А) $x_1=4, x_2=2, x_3=3$; Б) $x_1=9, x_2=5, x_3=3$;
 В) $x_1=12, x_2=5, x_3=4$; Г) $x_1=15, x_2=9, x_3=7$;
 Д) $x_1=17, x_2=12, x_3=13$; Е) $x_1=19, x_2=9, x_3=8$

Задача 2

Найти y если дано

- А) $f = \{(x_1, x_2, x_3, y)\} = J_{3,1}, x_1 = 5, x_2 = 6, x_3 = 7$;
 Б) $f = \{(x_1, x_2, x_3, y)\} = J_{3,2}, x_1 = 5, x_2 = 6, x_3 = 7$;
 В) $f = \{(x_1, x_2, x_3, y)\} = J_{3,3}, x_1 = 5, x_2 = 6, x_3 = 7$;
 Г) $f = \{(x_1, x_2, x_3, y)\} = C_3, x_1 = 5, x_2 = 6, x_3 = 7$ и $C_3=1$
 Д) $f = \{(x_1, x_2, x_3, y)\} = C_3, x_1 = 5, x_2 = 6, x_3 = 7$ и $C_3=0$
 Е) $f = \{(x, y)\} = I(x), x = 6$

Задача 3. Вычислить **функции усеченного вычитания**

$$f(x, y) = \begin{cases} x - y & \text{при } x > y; \\ 0 & \text{при } x \leq y \end{cases}$$

по схеме примитивной рекурсии

Практическая работа № 3

Задание. Применяя алгоритм Хаффмана произвести кодирование символов в сообщении согласно варианта.

Номер варианта	Сообщение
1	RRRYYYDDSSSSSTTTDDDQQQQSSDDDTSSAAABB
2	FFFECCCCLPPLLKKKHNHKAABBEEERRTTYZZC

Практическая работа № 4

Задача 1

Определить результаты выполнения следующих операций над элементами массива $S = \{1, 2, 3, 4, 5\}$:

- А) $S.\text{Insert}\{6\}$; Б) $S.\text{Insert}\{4\}$; В) $S.\text{remove}\{4\}$; Г) $S.\text{remove}\{6\}$;
 Д) $S.\text{in}\{6\}$; Е) $\text{Size}\{S\}$; Ж) $\text{Maximum}\{S\}$:

Задача 2. Определить результаты выполнения следующих операций над

элементами стека $S = \{1, 2, 3, 8, 17\}$, заданного в форме массива

А) $S.Push\{18\}$; Б) $S.Pop$; В) $S.Peek$; Г) $S[1]$ Д) $S[top[S]]$

Задача 3

В заданном списке выделить головной элемент и хвостовую часть.

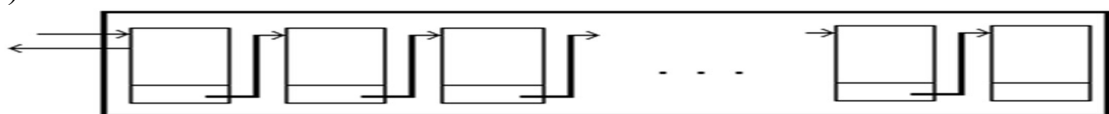
А) $[a, b, c]$ Б) $[5, 6, 7]$ В) $[[x, y], z, k]$ Г) $[A, [B, C, D]]$

Задача 4

Установить соответствие между структурой данных и ее графическим изображением

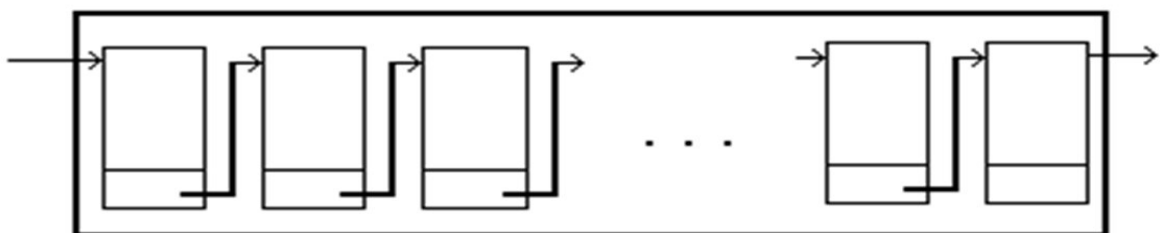
1. Стек 2) Очередь 3) Циклический список 4) Дек

А)

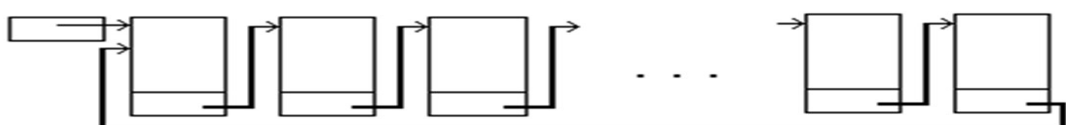


Б)

В)



Г)



Практическая работа № 5

Стек — структура данных, в которую можно добавлять элементы и извлекать их, причём элемент, который был добавлен последним, извлекается первым (**LIFO—Last In First Out**, последний пришёл первый ушёл). Указатель **top** всегда указывает на **последний** добавленный элемент.

Пример интерфейса класса `LinkedList` и структуры `Node`

```
template<class Type>
struct Node
{
    Type info;
    Node<Type>* link;
};
template<class Type>
class LinkedList
```

```
{
public:
linkedStack(); // Конструктор по умолчанию
linkedStack(const linkedStack<Type>
&otherStack);
// Копирующий конструктор
~linkedStack(); // Деструктор
const linkedStack<Type>& operator=(const
linkedStack<Type>& otherStack);
// Перегрузка оператора присваивания
bool isEmpty() const;
// Отвечает на вопрос пуст ли стэк?
void initialize();
// Опустошает стэк. (top = NULL)
void push(const Type& newItem);
// Добавляет newItem в стэк
Type top() const;
// Возвращает данные top элемента стэка
void pop(); // Удаляет top элемент стэка
```

private:

Node<Type>* top; // указатель на последний
добавленный элемент стэка

**void copy(const linkedStack<Type>
&otherStack) ;**

// Функция копирования. Создает копию стэка
otherStack и присваивает её **this** стэку.

Контрольные вопросы

5. Что такое абстрактная структура данных?
6. Основное свойство структуры данных стэк?
7. Напишите алгоритм основных операций со стэком.
8. Какие способы реализации структуры данных стэк вы знаете?
9. Каков будет результат применения операций PUSH($S,4$), PUSH($S,1$), PUSH($S,3$), POP(S), PUSH($S,6$), PUSH($S,4$), PUSH($S,1$), PUSH($S,3$), POP(S), POP(S) к пустому стэку, хранящемуся в массиве $S[1..6]$? К тому же стэку, уже хранящему 2 элемента?

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задание 1

Разработать консольное приложение, которое с помощью абстрактной структуры данных СТЭК располагает элементы данного массива (*вектора*) в обратном порядке (*инвертирует*). Тип данных элементов массива выбрать произвольно. Приложение должно:

1. делать запрос на ввод размера и элементов создаваемого массива;
2. выводить в консоль исходный массив после его создания;
3. делать запросы на:
 - 3.1. инвертирование исходного массива:
✓ в случае «Да» выводить в консоль инвертированный массив;
 - 3.2. изменение исходного массива:
✓ в случае «Да» создавать новый массив и выводить его в консоль;
 - 3.3. выход из приложения
✓ в случае «Да» выходить из приложения;

Задание 2

Разработать консольное приложение, которое с помощью абстрактной структуры данных СТЭК проверяет соответствие открывающих и 76

закрывающих HTML-тэгов во фрагменте HTML кода, введённого с клавиатуры. Приложение должно:

1. делать запрос на ввод HTML кода;
2. выводить в консоль:
 - ✓ в случае соответствия – сообщение об этом;
 - ✓ в случае несоответствия – сообщение об этом и тэг(и), для которого нет пары;
3. делать запрос на выход из приложения.

Задание 3

Разработать консольное приложение, которое с помощью абстрактной структуры данных СТЭК вычисляет арифметическое выражение, записанное в постфиксной форме (*postfix notation*). Выражение может содержать 4 основных действия, скобки и числа. Для распознавания чисел их можно вводить сразу после определённого символа, например, **#**. Приложение должно:

1. делать запрос на ввод выражения в постфиксной форме;
2. выводить в консоль:
 - ✓ введённое выражение и результат вычислений в случае корректного ввода;
 - ✓ сообщение об ошибке и введённое выражение в случае некорректного ввода;
3. делать запрос на выход из приложения.

Практическая работа № 6

Двусвязный линейный список — структура данных, где каждый элемент содержит указатель на **следующий** и **предыдущий** элементы

Поддерживаемые операции

1. вставить элемент перед текущим элементом;
2. вставить элемент после текущего элемента;
3. получить указатель на первый элемент списка.
4. получить указатель на последний элемент списка.
5. получить данные первого элемента списка.
6. получить данные последнего элемента списка.
7. получить указатель на следующий, относительно данного, элемент списка;
8. получить указатель на предыдущий, относительно данного, элемент списка;
9. удалить выбранный элемент из списка;
10. найти элемент с указанной информационной частью и вернуть указатель на него;
11. очистить список;
12. копировать список;
13. вывести в консоль все элементы списка

ДОБАВЛЕНИЕ ЭЛЕМЕНТОВ В ДВУСВЯЗНЫЙ СПИСОК

Графическая иллюстрация:

Важно! Как при добавлении, так и при удалении элементов необходимо:

1. Специально рассматривать случай когда элемент добавляется или удаляется в качестве первого или последнего элемента списка. В этом случае перенаправляются указатели first и last.

2. Проверять, является ли список пустым или нет.

При создании двусвязного списка нужно помнить, что каждый элемент списка имеет 2 (два) указателя, — на предыдущий и на следующий элементы.

Первый элемент списка имеет указатель на предыдущий элемент, равный NULL (nullptr). Последний элемент списка имеет указатель на следующий элемент, равный NULL (nullptr).

Контрольные вопросы

1. Что такое абстрактная структура данных?

2. Основное свойство структуры данных двусвязный линейный список?

3. Напишите алгоритм основных операций с двусвязным линейным списком?

4. Перечислите операции двусвязного списка, которые «недоступны» для стека и очереди?

5. Перечислите операции двусвязного списка, которые также «доступны» для стека? Очереди?

ПРАКТИЧЕСКИЕ ЗАДАНИЯ

Задание 1

Используя абстрактную структуру данных **ДВУСВЯЗНЫЙ ЛИНЕЙНЫЙ СПИСОК**, разработать консольное приложение в соответствии со своим вариантом. Приложение должно обеспечивать диалог с помощью меню и контроль ошибок при вводе.

Вариант 1

Динамическая информация о наличии автобусов в автобусном парке и на маршрутах включают в себя следующие сведения о каждом автобусе:

📍 номер автобуса;

📍 фамилию и инициалы водителя;

first

last

NULL

NULL

📍 номер маршрута.

Программа должна обеспечивать:

📍 начальное формирование данных о всех автобусах в виде списка;

📍 по номеру автобуса:

📖 «переводить» автобус из парка на маршрут – удалять данные об этом автобусе из списка автобусов, находящихся в парке, и добавлять эти данные в список автобусов, находящихся на маршруте;

📖 «переводить» автобус с маршрута в парк – удалять данные об этом автобусе из списка автобусов, находящихся на маршруте, и добавлять эти данные в список автобусов, находящихся в парке;

📍 выдавать по запросу сведения об автобусах, находящихся в парке, или об автобусах, находящихся на маршруте;

🖱 выдавать по запросу сведения о всех автобусах.

Вариант 2

Список содержит текущую информацию о заявках на авиабилеты. Каждая заявка включает в себя:

- 🖱 пункт назначения;
- 🖱 номер рейса;
- 🖱 фамилию и инициалы пассажира;
- 🖱 желаемую дату вылета.

Программа должна обеспечивать:

- 🖱 хранение всех заявок в виде списка;
- 🖱 добавление заявок в список;
- 🖱 удаление заявок из списка;
- 🖱 вывод заявок по заданному номеру рейса и дате вылета;
- 🖱 вывод всех заявок.

Вариант 3

Список содержит текущую информацию о книгах в библиотеке. Сведения о книгах включают в себя:

- 🖱 номер УДК;
- 🖱 фамилию и инициалы автора;
- 🖱 название;
- 🖱 год издания;
- 🖱 количество экземпляров данной книги в библиотеке.

Программа должна обеспечивать:

- 🖱 начальное формирование данных о всех книгах в библиотеке в виде вписка;
- 🖱 возможность получить книгу при вводе номера УДК и в зависимости от состояния списка:
 - 📖 выдавать книгу «на руки» и уменьшать кол-во экземпляров данной книги в библиотеке на единицу;
 - 📖 выдавать сообщение о том, что требуемой книги в библиотеке нет;
 - 📖 требуемая книга находится на руках.
- 🖱 возможность вернуть книгу при вводе номера УДК (кол-во экземпляров книги увеличивается на единицу);
- 🖱 выдавать сведения о наличии книг в библиотеке по запросу.

Вариант 4

Для каждого файла в некотором каталоге содержатся следующие сведения:

- 🖱 имя файла;
- 🖱 дата создания;
- 🖱 количество обращений к файлу.

Программа должна обеспечивать:

- 🖱 начальное формирование каталога файлов;
- 🖱 вывод каталога файлов;
- 🖱 удаление файлов, дата создания которых меньше заданной;
- 🖱 выборку файлов с наибольшим количеством обращений.

Вариант 5

Каждая компонента предметного указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти.

Программа должна обеспечивать:

- 🌐 начальное формирование предметного указателя;
- 🌐 вывод предметного указателя;
- 🌐 вывод номера страниц для заданного слова.

Вариант 6

Каждая компонента текста помощи для программы содержит термин (слово) и текст, содержащий пояснения к этому термину. Количество строк текста, относящихся к одному термину, от одной до пяти.

Программа должна обеспечивать:

- 🌐 начальное формирование текста помощи;
- 🌐 вывод текста помощи;
- 🌐 вывод поясняющего текста для заданного термина.

Вариант 7

Картотека в бюро обмена квартир содержит следующие сведения о каждой квартире:

- 🌐 количество комнат;
- 🌐 этаж;
- 🌐 площадь;
- 🌐 адрес.

Программа должна обеспечивать:

88

- 🌐 начальное формирование картотеки;
- 🌐 ввод заявки на обмен;
- 🌐 поиск в картотеке подходящего варианта:
 - 📖 при равенстве кол-ва комнат и этажа и различии площадей в пределах 10м² выводится соответствующая карточка и удаляется из списка;
 - 📖 в противном случае поступившая заявка включается в список;
- 🌐 вывод всего списка.

Вариант 8

Анкета для опроса населения содержит две группы вопросов. Первая группа содержит сведения о респонденте:

- 🌐 возраст;
- 🌐 пол;
- 🌐 образование (начальное, среднее, высшее).

Вторая группа содержит собственно вопрос анкеты, ответ на который может быть ДА или НЕТ.

Программа должна обеспечивать:

- 🌐 начальный ввод анкет и формирование из них линейного списка;
- 🌐 ответы на следующие вопросы на основе анализа анкет:
 - а). сколько мужчин старше 40 лет, имеющих высшее образование, ответили ДА на вопрос анкеты;

- б). сколько женщин моложе 30 лет, имеющих среднее образование, ответили НЕТ на вопрос анкеты;
- в). сколько мужчин моложе 25 лет, имеющих начальное образование, ответили ДА на вопрос анкеты;
- г). вывод всех анкет и ответов на вопросы в консоль.

Вариант 9

На междугородной телефонной станции картотека абонентов, содержит сведения о телефонах и их владельцах.

Программа должна обеспечивать:

- 🖱 начальное формирование картотеки в виде линейного списка;
- 🖱 вывод всей картотеки в консоль;
- 🖱 ввод номера телефона и вывод времени разговора;
- 🖱 ввод данных абонента и вывод извещения на оплату телефонного разговора.

Практическая работа № 8

Сравнительный анализ алгоритмов сортировки

Для выполнения задания необходимо разделиться на команды по 5-6 человек в каждой.

Темы для проектов

- 1) Быстрая сортировка
- 2) Сортировка посредством простого выбора
- 3) Сортировка квадратичным выбором
- 4) Сортировка выбором из дерева
- 5) Сортировка вставками

Проект должен включать в себя:

- 1) Теоретическое описание выбранного алгоритма сортировки
- 2) Оценку вычислительной сложности данного метода
- 3) Листинг программного кода, реализующий соответствующий алгоритм сортировки
- 4) Анализ результатов работы алгоритма для различных вариантов реализации программного кода (время работы программы, объем используемой памяти и т.д)
- 5) Результаты тестов программной реализации алгоритма для массивов данных разной длины (от 100 до 1000 чисел) и степени отсортированности (почти отсортированная, отсортированная наоборот, случайная выборка)
- 6) Выводы относительно эффективности применения данного алгоритма для наборов данных различной длины и степени отсортированности

Данный проект должен быть оформлен в форме презентации

Практическая работа № 9

Алгоритмы на деревьях поиска

Для выполнения проекта необходимо разделиться на команды по 5-6 человек в каждой.

Темы для проектов

- 1) Алгоритм поиска элемента с заданным ключом
- 2) Алгоритм удаление вершины без потомков
- 3) Алгоритм удаление вершины с двумя потомками
- 4) Алгоритм удаление вершины с одним потомком
- 5) Алгоритм вставки элемента в дерево
- 6) Алгоритм поиска минимального элемента в дереве.

Проект должен включать в себя:

- 1) Теоретическое описание выбранного алгоритма для дерева поиска.
- 2) Оценку вычислительной сложности данного метода.
- 3) Листинг программного кода, реализующий соответствующий алгоритм.
- 4) Анализ результатов работы алгоритма для различных вариантов реализации программного кода (время работы программы, объем используемой памяти и т.д).
- 5) Результаты тестов программной реализации алгоритма для деревьев разной высоты (от 100 до 1000) и степени сбалансированности (сбалансированное и несбалансированное дерево поиска).
- 6) Выводы относительно эффективности применения данного алгоритма для наборов данных различной длины и сбалансированности дерева.

Данный проект должен быть оформлен в форме презентации и загружен в Булгаков не позднее двух недель с