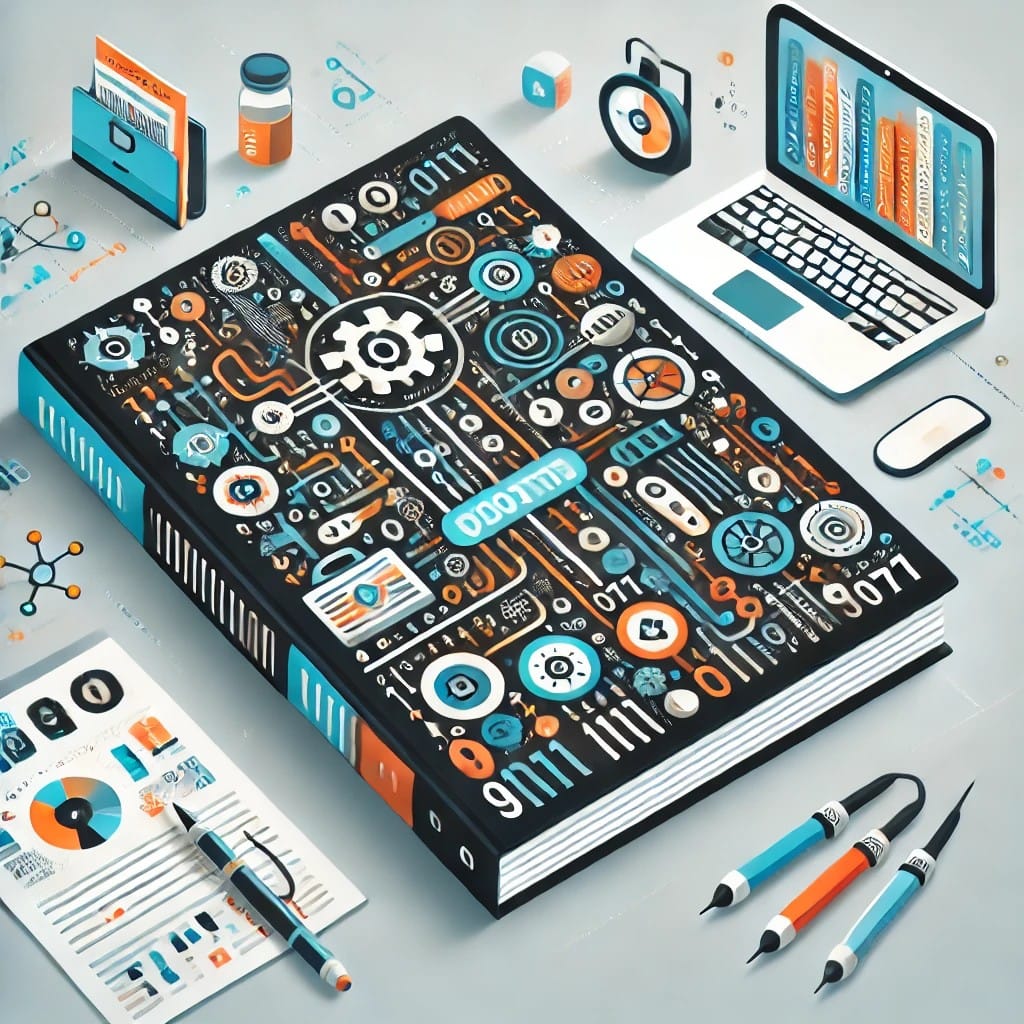
КГУ «Общеобразовательная школа имени Надежды Крупской города Державинск отдела образования по Жаркаинскому району управления образования Акмолинской области"

**"Информатика**

**и выбор профессии"**

Сборник практических заданий

для 9 класса



Учитель информатики Каирова Д.М.

|  |  |  |
| --- | --- | --- |
| **Раздел 1. Работа с информацией** | | |
| № | Тема урока | Цели урока |
| 1 | Свойства информации | 9.2.1.1 определять свойства информации (актуальность, точность, достоверность, ценность); 9.4.1.1 критически оценивать риски, связанные с продолжительным использованием компьютеров |
| **Тема: Свойства информации и их роль в работе аналитика данных**  **Цель задания:**  Понять свойства информации и их применение в профессии аналитика данных.  **Описание:**  Аналитик данных (Data Analyst) — специалист, который собирает, анализирует и интерпретирует данные для поддержки принятия решений. В своей работе он использует свойства информации, такие как достоверность, актуальность, полнота, ценность, точность и понятность. Ваша задача — проанализировать, как аналитик данных использует эти свойства для выполнения своих профессиональных обязанностей.  **Этапы выполнения:**   1. **Изучение свойств информации**:    * **Актуальность**: Информация должна быть своевременной и соответствовать текущим задачам.    * **Достоверность**: Информация должна быть проверенной и надежной.    * **Полнота**: Информация должна быть достаточной для принятия решений.    * **Ценность**: Информация должна приносить пользу или экономическую выгоду.    * **Точность**: Информация должна быть точной и соответствовать реальным данным.    * **Понятность**: Информация должна быть представлена в форме, доступной для понимания. 2. **Задание**: Представьте, что вы — начинающий аналитик данных, которому поручили подготовить отчёт для компании на основе определённого набора данных (например, данные о продажах или посещаемости сайта).   **Задания к выполнению:**   * + **Анализ достоверности информации**: Проверьте, откуда поступили данные, и оцените их надежность (например, источники данных могут быть официальные отчеты или внутренние данные компании).   + **Оценка актуальности данных**: Проверьте, насколько данные актуальны (например, не устарели ли они). Оцените, нужно ли дополнительно собирать новые данные.   + **Проверка полноты**: Убедитесь, что у вас есть все необходимые данные для анализа. Если данных недостаточно, укажите, какие данные следует дополнительно собрать.   + **Ценность данных**: Проанализируйте, как данные могут быть полезны для принятия решений. Приведите примеры, как компания может использовать эту информацию (например, оптимизация продаж, улучшение маркетинговой стратегии).   + **Оценка точности**: Проверьте точность данных (например, верны ли расчеты, нет ли ошибок при вводе данных).   + **Понятность отчёта**: Представьте отчёт так, чтобы он был понятен даже для людей, не обладающих глубокими знаниями в области анализа данных (графики, диаграммы, понятные объяснения).  1. **Презентация результатов**: Подготовьте краткий отчет или презентацию, в которой:    * Объясните каждое свойство информации.    * Покажите, как эти свойства применяются при работе с данными.    * Приведите примеры ошибок, которые могут возникнуть, если игнорировать те или иные свойства информации (например, устаревшие или недостоверные данные могут привести к неверным бизнес-решениям).   **Оценка:**   * Понимание и правильное использование свойств информации. * Примеры того, как аналитик данных применяет эти свойства в своей работе. * Качество представления отчета: ясность, полнота и структурированность.   Таким образом, учащиеся не только изучат теорию о свойствах информации, но и поймут её прикладное значение через профессию аналитика данных, что сделает урок более интересным и практичным. | | |
| 2 | Совместная работа с документами | 9.1.3.1 осуществлять совместную работу с документами с использованием облачных технологий |
| **Тема: Совместная работа с документами и её роль в работе менеджера проектов**  **Цель задания:**  Изучить способы совместной работы с документами и понять, как это используется в профессии менеджера проектов для эффективного управления командой и проектом.  **Описание:**  Менеджер проекта отвечает за планирование, организацию и контроль выполнения задач в рамках проекта. Одной из ключевых задач менеджера является работа с документами, особенно когда проектная команда состоит из нескольких человек. Для успешного управления проектом менеджер должен уметь организовать эффективную совместную работу с документами.  Ваша задача — смоделировать ситуацию, в которой вы, как менеджер проекта, организуете работу команды, используя инструменты для совместной работы с документами.  **Этапы выполнения:**   1. **Изучение инструментов для совместной работы с документами**:    * Исследуйте популярные инструменты для совместного редактирования документов (например, Google Docs, Microsoft Word Online, Trello, Asana или другие).    * Определите, как эти инструменты позволяют нескольким людям одновременно редактировать документы, оставлять комментарии, отслеживать изменения и работать над общими задачами. 2. **Задание**: Представьте, что вы — менеджер проекта, который организует совместную работу над проектом по созданию нового продукта для компании (например, мобильного приложения или веб-сайта). В команде несколько специалистов: дизайнер, программист, маркетолог и тестировщик.   **Задания к выполнению:**   * + Создайте шаблон **плана проекта** в Google Docs или Microsoft Word Online, где каждый член команды сможет добавлять свои идеи и предложения в реальном времени.   + Разработайте таблицу в документе, где каждый участник будет записывать свои задачи и сроки выполнения.   + Организуйте обсуждение документа с помощью функции комментариев: попросите каждого участника команды оставить свои комментарии к плану, предложить улучшения или уточнения.   + Назначьте отдельные разделы документа для каждого члена команды, где они будут добавлять информацию, связанную с их работой (например, дизайнер добавит идеи для дизайна, программист — описание функционала, маркетолог — стратегию продвижения).   + Используйте функцию отслеживания изменений, чтобы видеть, кто и когда вносил правки, и при необходимости возвращаться к предыдущим версиям документа.  1. **Презентация результатов**:    * Подготовьте итоговый отчет, в котором:      + Объясните, как происходила совместная работа с документом, какие инструменты использовались.      + Покажите примеры комментариев и правок, которые сделали члены команды.      + Опишите, как менеджер проекта контролировал процесс и координировал действия команды, используя совместные документы. 2. **Дополнительное задание (по желанию)**:    * Внедрите интеграцию между разными инструментами (например, Google Docs и Trello), чтобы отслеживать задачи и прогресс команды в режиме реального времени.   **Результат:**  Это задание позволяет учащимся не только изучить инструменты для совместной работы с документами, но и применить знания на практике, моделируя реальные рабочие ситуации, с которыми сталкиваются менеджеры проектов в своей повседневной деятельности. | | |
| 3 | Сетевой этикет | 9.4.2.1 рассуждать о последствиях нарушения этических и правовых норм в сети |
| **Тема: Сетевой этикет и его роль в работе менеджера по работе с клиентами(Customer Support Manager).**  **Цель задания:**  Понять основы сетевого этикета и их применение в профессиональной деятельности, связанной с взаимодействием с клиентами через интернет, на примере профессии менеджера по работе с клиентами.  **Описание:**  Менеджер по работе с клиентами (Customer Support Manager) взаимодействует с клиентами через электронную почту, мессенджеры, социальные сети и другие онлайн-платформы. В его обязанности входит грамотное, вежливое и профессиональное общение, соблюдение норм сетевого этикета. Ваша задача — проанализировать ключевые принципы сетевого этикета и применить их на практике, моделируя взаимодействие с клиентами.  **Этапы выполнения:**   1. **Изучение правил сетевого этикета**:    * Ознакомьтесь с основными принципами сетевого этикета:      + **Вежливость и уважение**: обращение к собеседнику с уважением, использование приветствий и благодарностей.      + **Правильное оформление сообщений**: избегание капслока (он воспринимается как крик), соблюдение грамматических и пунктуационных норм.      + **Оперативность ответа**: своевременные ответы на вопросы и запросы клиентов.      + **Краткость и ясность**: сообщение должно быть четким, понятным и по существу.      + **Конфиденциальность**: защита персональных данных и уважение к частной информации клиента. 2. **Задание**: Представьте, что вы — менеджер по работе с клиентами, и ваша задача — вести профессиональную переписку с клиентами, соблюдая нормы сетевого этикета.   **Задания к выполнению**:   * + **Написать вежливый ответ на запрос клиента**: Клиент обращается к вам с вопросом о продукте или услуге. Ответьте на его запрос, следуя правилам сетевого этикета (приветствие, использование вежливых фраз, грамотная структура текста).   Пример задачи: Клиент написал, что у него возникли проблемы с входом в личный кабинет на сайте. Ваше задание — ответить ему, предлагая решение и соблюдая сетевой этикет.   * + **Решение конфликтной ситуации**: Клиент недоволен качеством продукта и написал жалобу. Напишите ответ, в котором вы профессионально и вежливо реагируете на претензию, предлагая возможные пути решения проблемы.   Пример задачи: Клиент недоволен тем, что его заказ задержан. Вам нужно ответить, объяснив причину задержки и предложив варианты компенсации или решения проблемы.   * + **Создать шаблон письма** для общения с клиентами: Составьте шаблон письма, который можно будет использовать для стандартных ответов клиентам (например, по поводу доставки товаров, уточнения информации или запроса обратной связи).  1. **Презентация результатов**:    * Подготовьте несколько примеров переписки (имитацию общения с клиентами), в которых продемонстрируете правила сетевого этикета. Каждое сообщение должно содержать приветствие, основной текст и заключение с благодарностью.    * Объясните, какие правила сетевого этикета вы использовали, и как они помогают поддерживать профессиональные и вежливые отношения с клиентами. 2. **Дополнительное задание (по желанию)**:    * Проведите небольшой анализ: сравните примеры "хорошей" и "плохой" переписки с точки зрения сетевого этикета. Объясните, какие ошибки могут возникнуть при игнорировании правил и как это может повлиять на отношения с клиентами и репутацию компании.   **Оценка:**   * Понимание основных принципов сетевого этикета. * Умение применять правила сетевого общения в профессиональных ситуациях. * Качество примеров переписки: вежливость, грамотность, соблюдение правил оформления. * Способность решать конфликтные ситуации через онлайн-общение.   **Результат:**  Учащиеся научатся применять правила сетевого этикета на практике, моделируя профессиональные ситуации, с которыми сталкиваются менеджеры по работе с клиентами. Это задание поможет развить навыки вежливого и грамотного онлайн-общения, что важно в любой профессии, связанной с работой через интернет. | | |
| **Раздел 2. Выбираем компьютер** | | |
|  | Конфигурация компьютера | 9.1.1.1 выбирать конфигурацию компьютера в зависимости от его назначения |
| Задание по информатике на тему "Конфигурация компьютера" можно легко связать с профессией **системного администратора** или **ИТ-специалиста** по сборке и настройке компьютеров. Это задание поможет учащимся понять, какие компоненты компьютера важны для различных профессиональных задач.  **Задание: Конфигурация компьютера для системного администратора**  **Цель задания:**  Разработать конфигурацию компьютера для системного администратора, учитывая специфику его работы.  **Описание:**  Системный администратор отвечает за управление, поддержку и обслуживание компьютерных сетей и серверов. Для выполнения этих задач ему требуется компьютер с определёнными характеристиками. Ваша задача — разработать оптимальную конфигурацию рабочего компьютера для системного администратора, учитывая его профессиональные потребности.  **Этапы выполнения:**   1. **Изучение работы системного администратора**:    * Изучите, какие задачи выполняет системный администратор (управление серверами, работа с сетевым оборудованием, настройка программного обеспечения, мониторинг безопасности и др.).    * Определите, какие ресурсы (оперативная память, процессор, диск и т.д.) наиболее важны для выполнения этих задач. 2. **Подбор компонентов компьютера**:    * **Процессор**: Определите, какой процессор лучше подходит для многозадачной работы и запуска виртуальных машин.    * **Оперативная память (ОЗУ)**: Рассчитайте необходимый объем памяти для работы с несколькими виртуальными машинами и специализированными программами.    * **Жесткий диск (SSD/HDD)**: Оцените необходимость быстрого доступа к данным и сохранности информации.    * **Видеокарта**: Решите, нужна ли мощная видеокарта или достаточно встроенной графики.    * **Сетевое оборудование**: Подберите компоненты для быстрого сетевого соединения и удалённой работы. 3. **Составление итоговой конфигурации**:    * Соберите список выбранных компонентов, указав их основные характеристики и стоимость.    * Объясните, почему каждый компонент выбран, и как он будет использоваться в работе системного администратора. 4. **Презентация**:    * Подготовьте презентацию, в которой покажете выбранную конфигурацию и обоснование каждого компонента.    * Опишите, как эта конфигурация улучшит рабочий процесс системного администратора.   **Оценка:**   * Полнота и обоснованность выбора конфигурации. * Понимание специфики работы системного администратора. * Умение анализировать технические характеристики и их соответствие профессиональным задачам.   Таким образом, учащиеся не только изучат конфигурацию компьютера, но и поймут, как правильно подбирать оборудование под определенные профессиональные задачи. | | |
|  | Выбор программного обеспечения | 9.1.2.1 выбирать программное обеспечение в зависимости от потребности пользователя |
| **Тема: Выбор программного обеспечения в зависимости от потребности пользователя**  **Цель задания:**  Научиться выбирать программное обеспечение, исходя из потребностей пользователя, используя знания о программных решениях, и понять, как это умение применяется в работе IT-консультанта.  **Описание:**  **IT-консультант** — это профессионал, который помогает организациям и частным пользователям подбирать наиболее подходящее программное обеспечение для решения различных задач, таких как автоматизация процессов, защита данных, улучшение продуктивности и т.д. IT-консультант должен уметь анализировать потребности пользователя и предлагать соответствующие программы.  Ваша задача — выбрать оптимальное программное обеспечение для конкретных пользователей, исходя из их потребностей.  **Этапы выполнения:**   1. **Изучение различных типов программного обеспечения**:    * **Офисные приложения** (например, Microsoft Office, Google Workspace).    * **Графические редакторы** (Adobe Photoshop, GIMP).    * **Программы для работы с видео** (Adobe Premiere, DaVinci Resolve).    * **Антивирусы и программы для защиты данных** (Kaspersky, Avast).    * **Программное обеспечение для управления проектами** (Trello, Asana). 2. **Задание**: Представьте, что вы — IT-консультант, и вам нужно помочь нескольким пользователям выбрать программное обеспечение, которое решит их задачи.   **Задачи к выполнению**:   * + **Пользователь 1**: Фрилансер-дизайнер, работающий с графикой. Ему требуется программа для редактирования изображений.     - Какие программы вы предложите ему?     - Обоснуйте выбор с точки зрения функциональности и цены.   + **Пользователь 2**: Руководитель малого бизнеса, который нуждается в инструменте для управления проектами и отслеживания задач в команде.     - Какие программные решения вы предложите для управления проектами?     - Объясните, как выбранные программы помогут пользователю организовать работу команды.   + **Пользователь 3**: Студент, которому нужно писать эссе, делать презентации и работать с электронными таблицами.     - Какие офисные программы лучше всего подойдут для его задач?     - Объясните, почему вы выбрали это программное обеспечение.   + **Пользователь 4**: Обычный пользователь, который беспокоится о безопасности своего компьютера и данных.     - Какие антивирусные программы и средства защиты данных вы порекомендуете?     - Объясните выбор с точки зрения надежности и простоты использования.  1. **Презентация результатов**:    * Подготовьте документ или презентацию, в которой покажете выбранное программное обеспечение для каждого пользователя.    * Обоснуйте свой выбор, объясняя, как программы решают конкретные задачи пользователя и чем они отличаются от других вариантов на рынке.    * Укажите ключевые критерии выбора программного обеспечения: цена, функциональность, простота использования, поддержка и безопасность. 2. **Дополнительное задание (по желанию)**:    * Проанализируйте преимущества и недостатки платного и бесплатного программного обеспечения для каждого пользователя. Объясните, в каких случаях бесплатные решения могут быть лучше платных.   **Оценка:**   * Умение правильно выбрать программное обеспечение в зависимости от потребности пользователя. * Понимание ключевых характеристик программного обеспечения. * Способность аргументировать выбор и сравнивать разные программы. * Качество презентации с ясными и структурированными обоснованиями.   **Результат:**  Учащиеся научатся выбирать программное обеспечение в зависимости от конкретных потребностей пользователей, что является ключевым навыком для любой профессии в сфере IT. Задание поможет им понять, как эти знания применяются в реальной жизни на примере профессии IT-консультанта. | | |
|  | «Расчет стоимости компьютера» | 9.1.1.1 выбирать конфигурацию компьютера в зависимости от его назначения; 9.1.2.1 выбирать программное обеспечение в зависимости от потребности пользователя; 9.3.1.1 разрабатывать и исследовать модели процессов (физических, биологических, экономических) в электронных таблицах |
| **IT-специалиста по сборке компьютеров.**  **Цели урока:**   * **9.1.1.1** Выбирать конфигурацию компьютера в зависимости от его назначения. * **9.1.2.1** Выбирать программное обеспечение в зависимости от потребности пользователя. * **9.3.1.1** Разрабатывать и исследовать модели процессов (физических, биологических, экономических) в электронных таблицах.   **Задание: Расчет стоимости компьютера для разных профессий**  **Описание:**  Представьте, что вы — **системный администратор** или **IT-специалист по сборке компьютеров**, и ваша задача — разработать конфигурацию компьютеров для разных типов пользователей, а также рассчитать стоимость каждой конфигурации с учетом необходимого программного обеспечения.  **Этапы выполнения:**  **1. Выбор конфигурации компьютера в зависимости от назначения (9.1.1.1)**   * **Задача**: Вам необходимо разработать три разные конфигурации компьютера для разных профессий. Например:   + **Компьютер для геймера**: Высокопроизводительный компьютер с мощной видеокартой, процессором, оперативной памятью и большим объёмом SSD.   + **Компьютер для дизайнера**: Мощный компьютер с качественным дисплеем, видеокартой и большим объемом оперативной памяти для работы с графикой и видео.   + **Компьютер для офисного работника**: Базовая конфигурация для работы с офисными программами и электронной почтой. * **Задание**:   + Выберите комплектующие для каждого типа компьютера (процессор, видеокарта, оперативная память, диск, блок питания и т.д.).   + Рассчитайте стоимость каждой конфигурации, используя данные с интернет-магазинов или типовые цены.   + Представьте собранную конфигурацию для каждого пользователя и объясните, почему вы выбрали именно такие компоненты для каждой профессии.   **2. Выбор программного обеспечения в зависимости от потребностей пользователя (9.1.2.1)**   * **Задача**: После того как конфигурации компьютеров выбраны, необходимо подобрать программное обеспечение для каждой профессии:   + **Геймер**: Платформы для игр (Steam), программы для записи и стриминга видео.   + **Дизайнер**: Графические редакторы (Adobe Photoshop, CorelDRAW), программы для видеомонтажа (Premiere Pro).   + **Офисный работник**: Офисные программы (Microsoft Office, Google Workspace), антивирус. * **Задание**:   + Для каждого типа компьютера подберите необходимое программное обеспечение, объясняя, почему оно нужно для данного пользователя.   + Рассчитайте стоимость программного обеспечения (если оно платное) и добавьте в общую стоимость компьютера.   **3. Разработка модели в электронных таблицах (9.3.1.1)**   * **Задача**: Составьте модель расчета стоимости компьютера в электронных таблицах (например, в Google Sheets или Microsoft Excel).   + Создайте таблицу, где для каждой конфигурации компьютера будет указана стоимость всех компонентов (процессор, видеокарта, память и т.д.).   + Добавьте стоимость программного обеспечения.   + Рассчитайте общую стоимость каждой конфигурации. * **Задание**:   + Создайте таблицу, включающую стоимость каждого компонента и ПО для каждой конфигурации.   + Постройте график, который покажет зависимость стоимости компьютера от его назначения (например, офисный компьютер дешевле, чем игровой).   + Используйте функции электронной таблицы для автоматического расчета общей стоимости конфигураций и для анализа зависимости стоимости от изменений в комплектации.   **Пример структуры задания:**  **Таблица для расчета стоимости компонентов и ПО:**   | **Компоненты / ПО** | **Офисный ПК** | **ПК для геймера** | **ПК для дизайнера** | | --- | --- | --- | --- | | Процессор | 5 000 | 20 000 | 15 000 | | Видеокарта | 3 000 | 50 000 | 30 000 | | Оперативная память | 2 000 | 10 000 | 8 000 | | SSD / HDD | 4 000 | 10 000 | 7 000 | | Монитор | 6 000 | 15 000 | 20 000 | | Программное обеспечение | 10 000 | 5 000 | 15 000 | | **Общая стоимость** | **30 000** | **110 000** | **95 000** |   **Презентация результатов:**   * Создайте презентацию или отчет, где представьте все три конфигурации компьютеров, включая выбор компонентов и программного обеспечения. * Объясните выбор конфигурации и ПО для каждого пользователя. * Покажите модель расчета стоимости в таблицах и продемонстрируйте графики, построенные на основе данных.   **Оценка:**   * Умение выбирать конфигурацию компьютера в зависимости от его назначения. * Правильный выбор программного обеспечения для различных пользователей. * Способность создавать экономические модели в электронных таблицах. * Логичное и аргументированное объяснение выбора конфигураций и ПО.   **Результат:**  Учащиеся научатся выбирать конфигурацию компьютера для различных задач и пользователей, выбирать соответствующее программное обеспечение, а также разрабатывать экономические модели в электронных таблицах. Это задание позволяет развить навыки работы с реальными потребностями пользователей и сделать процесс обучения более практико-ориентированным.  **Менеджер по продажам компьютерной техники**  **Цели урока:**   * **9.1.1.1** Выбирать конфигурацию компьютера в зависимости от его назначения. * **9.1.2.1** Выбирать программное обеспечение в зависимости от потребности пользователя. * **9.3.1.1** Разрабатывать и исследовать модели процессов (физических, биологических, экономических) в электронных таблицах.   **Описание:**  Представьте, что вы — **менеджер по продажам компьютерной техники** в магазине. Ваша задача — помочь клиенту выбрать компьютер в зависимости от его потребностей и рассчитать полную стоимость, включая стоимость программного обеспечения.  **Этапы выполнения:**  **1. Выбор конфигурации компьютера в зависимости от назначения (9.1.1.1)**   * **Задача**: Вам необходимо выбрать три разные конфигурации компьютера для разных типов клиентов. Например:   + **Клиент 1: Студент для учебы и базовой работы** (офисные программы, просмотр видео, работа с интернетом).   + **Клиент 2: Геймер** (высокопроизводительный компьютер для игр с мощной видеокартой и процессором).   + **Клиент 3: Графический дизайнер** (компьютер с мощными ресурсами для работы с графикой и видео). * **Задание**:   Выберите компоненты для каждой конфигурации компьютера (процессор, видеокарта, оперативная память, диск и т.д.).  Найдите цены на эти компоненты в онлайн-магазинах.  Рассчитайте стоимость каждой конфигурации.  Объясните, почему вы выбрали именно такие компоненты для каждого клиента, исходя из их целей использования компьютера.  **2. Выбор программного обеспечения в зависимости от потребности пользователя (9.1.2.1)**   * **Задача**: Подобрать программное обеспечение для каждого клиента в зависимости от его нужд.   **Студент**: Офисные программы (Microsoft Office или LibreOffice), браузеры, антивирусное ПО.  **Геймер**: Платформы для игр (Steam), программы для записи видео или стриминга.  **Графический дизайнер**: Графические редакторы (Adobe Photoshop, CorelDRAW), программы для видеомонтажа.   * **Задание**:   + Подберите подходящее программное обеспечение для каждой конфигурации.   + Рассчитайте стоимость программного обеспечения (если оно платное).   + Добавьте стоимость программного обеспечения в общий расчет стоимости компьютера.   **3. Разработка и исследование модели в электронных таблицах (9.3.1.1)**  **Задача**: Разработать модель расчета стоимости компьютера в электронных таблицах (например, Microsoft Excel или Google Sheets).  Создайте таблицу, которая содержит список всех компонентов и программного обеспечения для каждой конфигурации, с указанием цен.  Используйте формулы для расчета общей стоимости каждой конфигурации.  Постройте диаграмму, которая покажет зависимость стоимости от типа компьютера (например, офисный ПК будет дешевле, чем игровой или дизайнерский ПК).  **Задание**:  Создайте таблицу, где укажете стоимость всех компонентов и программного обеспечения для каждой конфигурации.  Постройте график, показывающий, как варьируется стоимость компьютеров в зависимости от их назначения.  Используйте функции электронных таблиц для автоматического расчета общей стоимости и сравнения разных конфигураций.  **Пример структуры таблицы:**   | **Компоненты / ПО** | **Компьютер для студента** | **Компьютер для геймера** | **Компьютер для дизайнера** | | --- | --- | --- | --- | | Процессор | 10 000 | 20 000 | 18 000 | | Видеокарта | 5 000 | 40 000 | 35 000 | | Оперативная память | 4 000 | 12 000 | 10 000 | | SSD / HDD | 3 000 | 10 000 | 8 000 | | Программное обеспечение | 6 000 | 4 000 | 15 000 | | **Общая стоимость** | **28 000** | **86 000** | **86 000** |   **Презентация результатов:**   * Подготовьте документ или презентацию с тремя конфигурациями компьютеров, включая расчет стоимости компонентов и программного обеспечения для каждого клиента. * Объясните ваш выбор компонентов и программного обеспечения, исходя из потребностей каждого клиента. * Покажите таблицу с расчетами и график, демонстрирующий разницу в стоимости для разных типов компьютеров.   **Дополнительное задание (по желанию):**   * Проведите анализ различных вариантов для каждого клиента: сравните два варианта конфигурации для каждого типа пользователя (например, для геймера с бюджетной и топовой видеокартой) и оцените, как это повлияет на общую стоимость.   **Оценка:**   * Умение выбирать конфигурацию компьютера в зависимости от его назначения. * Способность подбирать программное обеспечение в зависимости от потребностей пользователя. * Умение разрабатывать экономические модели и проводить расчеты в электронных таблицах. * Логичное и аргументированное объяснение выбора конфигураций и программного обеспечения.   **Результат:**  Учащиеся научатся выбирать конфигурации компьютеров и программное обеспечение для разных типов пользователей, создавать экономические модели в электронных таблицах и визуализировать данные. Это задание позволяет развивать навыки, необходимые для работы менеджером по продажам компьютерной техники, что делает процесс обучения более практическим и полезным. | | |
| **Раздел 3. Базы данных** | | |
|  | Базы данных | 9.2.2.1 объяснять термины «базы данных, запись, поле» |
| Задание 1  **Разработка базы данных в электронных таблицах (Google Sheets или Microsoft Excel)**   * **Задача**: Создать модель базы данных в электронных таблицах. В одной таблице создайте поля и записи, как указано выше.   + Используйте функции сортировки, фильтрации и поиска для управления данными. Например, отфильтруйте клиентов по дате последнего взаимодействия.   + Рассчитайте количество клиентов, которые взаимодействовали с компанией в определенный период (например, за последние 6 месяцев).   **Пример структуры таблицы:**   | **Имя клиента** | **Номер телефона** | **Электронная почта** | **Дата последнего взаимодействия** | **Примечания** | | --- | --- | --- | --- | --- | | Иван Иванов | +7 999 999 9999 | ivanov@example.com | 01.09.2023 | Интересовался новинками | | Ольга Петрова | +7 888 888 8888 | petrova@example.com | 25.08.2023 | Ждет скидок | | Сергей Ковалев | +7 777 777 7777 | kovalev@example.com | 10.05.2023 | Интересовался поддержкой | | Анна Смирнова | +7 666 666 6666 | smirnova@example.com | 15.12.2022 | Запрашивала презентацию |   **4. Презентация и выводы:**   * **Презентация базы данных**: Покажите структуру вашей базы данных, объясните, какие поля были созданы и для чего. * **Анализ данных**: Продемонстрируйте, как можно использовать фильтры и сортировку для анализа информации в базе данных. Например, найдите всех клиентов, с которыми не было контакта более чем полгода. * **Роль базы данных в работе менеджера**: Объясните, как менеджер по работе с клиентами использует базу данных для поддержания связей с клиентами, улучшения сервиса и увеличения продаж.   **Дополнительное задание (по желанию):**   * Добавьте в базу данных дополнительные поля, такие как «Предпочтительные продукты» или «Дата последнего заказа». Используйте эту информацию для создания более детальных фильтров (например, клиентов, которые интересуются определенной продукцией, но не совершали покупку долгое время).   **Оценка:**   * Понимание основных терминов «база данных», «запись», «поле». * Умение создавать простые базы данных и заполнять их информацией. * Способность анализировать данные в базе с помощью фильтров и сортировки. * Логичное и структурированное объяснение, как база данных помогает в профессиональной деятельности.   **Результат:**  Учащиеся научатся понимать основные понятия баз данных и их применение в реальной профессии менеджера по работе с клиентами. Они получат практические навыки создания и использования баз данных, а также увидят, как анализ данных помогает в профессиональной деятельности.  **Задание 2**  **Банковский кассир-операционист**  **Цели урока:**   * **9.1.1.1** Выбирать конфигурацию компьютера в зависимости от его назначения. * **9.1.2.1** Выбирать программное обеспечение в зависимости от потребности пользователя. * **9.3.1.1** Разрабатывать и исследовать модели процессов (физических, биологических, экономических) в электронных таблицах.   **Описание:**  Профессия **банковского кассира-операциониста** предполагает активную работу с базами данных для ведения учета клиентов, выполнения денежных операций и отслеживания транзакций. Ваша задача — создать и настроить простую базу данных для работы с клиентами банка, выбрать подходящее программное обеспечение, а также разработать модель для анализа данных в электронных таблицах.  **Этапы выполнения:**  **1. Выбор конфигурации компьютера для работы с базами данных (9.1.1.1)**   * **Задача**: Выберите конфигурацию компьютера, который будет использоваться банковским кассиром для работы с базами данных. Учтите, что компьютер должен быть достаточно мощным для работы с финансовыми программами и базами данных, но не требует высокопроизводительных компонентов, как у геймерских ПК. * **Задание**:   1. Определите основные компоненты компьютера:      + **Процессор**: Достаточный для быстрого выполнения операций с базами данных.      + **Оперативная память (ОЗУ)**: Для обработки больших массивов данных.      + **Накопитель (SSD или HDD)**: Для хранения финансовых данных и отчетов.      + **Монитор**: Для удобной работы с таблицами и отчетами.   2. Рассчитайте примерную стоимость конфигурации, исходя из задач банковского кассира.   **2. Выбор программного обеспечения для работы с базами данных (9.1.2.1)**   * **Задача**: Подберите программное обеспечение, которое будет использоваться банковским кассиром для ведения баз данных клиентов и выполнения финансовых операций:   + **СУБД (система управления базами данных)**: Например, Microsoft Access, MySQL или Oracle.   + **Офисные программы**: Для работы с отчетами (Microsoft Excel, Google Sheets).   + **Программы для обеспечения безопасности**: Антивирус и программы шифрования для защиты клиентских данных. * **Задание**:   + Определите, какие программы необходимы для работы с базами данных.   + Рассчитайте стоимость программного обеспечения (если оно платное).   + Объясните, почему это программное обеспечение подходит для работы кассира-операциониста.   **3. Разработка базы данных и анализ в электронных таблицах (9.3.1.1)**   * **Задача**: Создайте упрощенную модель базы данных для хранения информации о клиентах банка. Включите следующие поля:   + **Имя клиента**.   + **Номер счета**.   + **Баланс счета**.   + **Дата последней транзакции**.   + **Тип счета** (например, дебетовый, кредитный). * **Задание**:   + **Создание базы данных**:     - Воспользуйтесь электронной таблицей (например, Google Sheets или Microsoft Excel) для создания базы данных.     - Введите несколько записей для разных клиентов с вымышленными данными.   + **Анализ данных**:     - Используя функции электронной таблицы, рассчитайте средний баланс по всем клиентам.     - Найдите клиентов с балансом ниже определенного значения (например, ниже 10 000 руб.).     - Создайте фильтры для сортировки данных по дате последней транзакции или типу счета.   + **Визуализация данных**:     - Постройте график или диаграмму, которая покажет распределение клиентов по типам счетов или изменения балансов со временем.   **Пример структуры базы данных:**   | **Имя клиента** | **Номер счета** | **Баланс счета** | **Дата последней транзакции** | **Тип счета** | | --- | --- | --- | --- | --- | | Иван Иванов | 1234567890 | 50 000 руб | 01.09.2023 | Дебетовый | | Ольга Петрова | 9876543210 | 8 000 руб | 15.08.2023 | Кредитный | | Сергей Ковалев | 1111222233 | 25 000 руб | 10.07.2023 | Дебетовый | | Анна Смирнова | 2222333344 | 120 000 руб | 01.09.2023 | Депозитный |   **4. Презентация результатов:**   * **Документ или презентация**: Представьте базу данных клиентов, объясните, почему именно такие поля были выбраны, и как кассир-операционист использует подобные данные в работе. * **Таблица с расчетом данных**: Покажите, как рассчитать средний баланс и отсортировать клиентов с минимальными остатками на счетах. * **Диаграмма**: Постройте диаграмму, показывающую распределение клиентов по типам счетов или иные экономические показатели.   **Дополнительное задание (по желанию):**   * **Анализ транзакций**: Добавьте в базу данных информацию о транзакциях клиентов (сумма, дата, тип транзакции) и проведите анализ того, какие клиенты чаще всего совершают переводы.   **Оценка:**   * Понимание основ баз данных и их применения в реальной жизни. * Способность выбирать программное обеспечение в зависимости от задач. * Умение работать с электронными таблицами, анализировать и визуализировать данные. * Логичное и структурированное объяснение выбора конфигурации компьютера и ПО для кассира.   **Результат:**  Учащиеся научатся создавать и использовать базу данных для работы с клиентами, исследовать экономические процессы с помощью электронных таблиц, а также выбирать конфигурацию компьютера и программное обеспечение для выполнения задач банковского кассира-операциониста. Это задание помогает применить теоретические знания на практике, учитывая требования реальной профессии.  **Задание 3**  **Практические задания по информатике для профессии архивариуса**  **Введение:**  Архивариус — специалист, который занимается организацией, хранением и управлением архивными документами. В современном мире архивные системы переходят в цифровой формат, и создание базы данных играет важную роль в эффективной организации архива. Ваша задача — научиться создавать и управлять базами данных, которые используются архивариусом для систематизации, поиска и анализа архивных материалов.  **Практические задания:**  **Задание 1: Создание структуры базы данных для архива**  **Цель:**  Научиться создавать структуру базы данных для хранения и поиска архивных документов.  **Описание:**  Представьте, что вы архивариус и вам нужно создать базу данных для хранения информации об архивных документах. В базе данных должны быть поля, которые позволяют легко находить и сортировать документы.  **Этапы выполнения:**   1. Определите поля для базы данных:    * **Название документа**.    * **Дата создания документа**.    * **Автор документа**.    * **Тип документа** (например, письмо, отчет, договор).    * **Место хранения** (архивное отделение или ячейка).    * **Краткое описание содержания документа**.    * **Дата добавления в архив**. 2. Создайте таблицу в программе для работы с электронными таблицами (например, Microsoft Excel, Google Sheets или LibreOffice Calc). 3. Заполните таблицу примерными данными для 10 архивных документов.   **Пример структуры базы данных:**   | **Название документа** | **Дата создания** | **Автор** | **Тип документа** | **Место хранения** | **Описание** | **Дата добавления** | | --- | --- | --- | --- | --- | --- | --- | | Отчет за 2023 год | 01.01.2023 | Иванов И.И. | Отчет | Отделение 1 | Финансовый отчет | 15.01.2023 | | Письмо Петру I | 10.02.1705 | Петр I | Письмо | Отделение 2 | Письмо по дипломатии | 12.12.2020 | | Договор аренды | 15.05.2015 | ООО "Рентал" | Договор | Отделение 3 | Договор аренды офиса | 20.05.2015 |   **Задание 2: Фильтрация и сортировка данных в базе данных**  **Цель:**  Научиться использовать функции сортировки и фильтрации для работы с базой данных.  **Описание:**  Ваша задача — использовать функции фильтрации и сортировки для организации архивных документов. Например, вам нужно найти все документы, созданные в 2023 году, или отсортировать документы по типу.  **Этапы выполнения:**   1. **Фильтрация**:    * Используйте функцию фильтрации, чтобы отобразить только документы, созданные в 2023 году.    * Найдите все документы, относящиеся к типу "Письмо". 2. **Сортировка**:    * Отсортируйте все записи по дате добавления в архив в порядке убывания.    * Отсортируйте документы по авторам в алфавитном порядке.   **Пример:**  После сортировки по дате создания документов ваши данные могут выглядеть так:   | **Название документа** | **Дата создания** | **Автор** | **Тип документа** | **Место хранения** | **Описание** | **Дата добавления** | | --- | --- | --- | --- | --- | --- | --- | | Письмо Петру I | 10.02.1705 | Петр I | Письмо | Отделение 2 | Письмо по дипломатии | 12.12.2020 | | Договор аренды | 15.05.2015 | ООО "Рентал" | Договор | Отделение 3 | Договор аренды офиса | 20.05.2015 | | Отчет за 2023 год | 01.01.2023 | Иванов И.И. | Отчет | Отделение 1 | Финансовый отчет | 15.01.2023 |   **Задание 3: Поиск и анализ данных в базе данных**  **Цель:**  Научиться эффективно искать информацию в базе данных и анализировать ее.  **Описание:**  В архиве часто нужно искать документы по разным параметрам — по дате, по автору или по содержанию. Ваша задача — разработать и применить критерии поиска в базе данных.  **Этапы выполнения:**   1. **Поиск по дате**:    * Найдите все документы, которые были добавлены в архив в 2023 году.    * Найдите документы, созданные в конкретный период (например, с 2015 по 2023 год). 2. **Поиск по ключевым словам**:    * Используйте функцию поиска, чтобы найти документы, содержащие определенное слово в описании (например, "отчет" или "договор"). 3. **Анализ**:    * Подсчитайте количество документов, относящихся к разным типам (письма, отчеты, договоры и т.д.).    * Рассчитайте средний возраст документов в архиве на основе их даты создания.   **Задание 4: Разработка модели для анализа архивных данных**  **Цель:**  Научиться разрабатывать модели для анализа данных с использованием электронных таблиц.  **Описание:**  Используя электронные таблицы, вы можете создавать модели для анализа архивных данных. Например, можно создать график, показывающий количество документов, добавленных в архив в разные годы, или диаграмму, отображающую распределение типов документов.  **Этапы выполнения:**   1. **Создание диаграммы**:    * Постройте диаграмму, которая покажет количество документов, добавленных в архив по годам. 2. **Анализ типов документов**:    * Создайте круговую диаграмму, которая отобразит распределение документов по типам (например, сколько писем, отчетов и договоров хранится в архиве). 3. **Рассчитайте общую численность документов** по каждому типу и создайте таблицу для наглядности.   **Пример диаграммы:**   * **Количество документов по годам**:   **Оценка:**   * Умение создавать структуры баз данных для архивных целей. * Способность работать с фильтрацией, сортировкой и поиском данных. * Навыки анализа архивных данных и создания отчетов. * Логичное и структурированное использование электронных таблиц для работы с данными.   **Результат:**  Учащиеся получат практические навыки создания баз данных, работы с архивными документами, а также научатся анализировать и визуализировать данные. Эти задания позволяют применить теоретические знания в реальной профессиональной деятельности архивариуса. | | |
|  | Создание базы данных в электронных таблицах | 9.2.2.2 создавать базу данных в электронной таблице |
| **Профессия HR-специалист (специалиста по кадрам)**  **Цель урока:**   * **9.2.2.2** Создавать базу данных в электронной таблице.   **Описание:**  Профессия **HR-специалиста** (специалиста по управлению персоналом) включает в себя работу с большими объемами данных о сотрудниках компании. Для этого HR-специалист использует базы данных для хранения и управления информацией о работниках: данные о стаже, должностях, зарплатах, датах приемов на работу и других аспектах. Ваша задача — создать базу данных сотрудников в электронной таблице, научиться управлять этой информацией и анализировать её.  **Задание 1: Создание структуры базы данных сотрудников**  **Цель:**  Научиться создавать структуру базы данных для хранения информации о сотрудниках компании.  **Описание:**  Вам нужно создать базу данных для хранения информации о сотрудниках компании, которая будет включать основные поля, необходимые для HR-специалиста.  **Этапы выполнения:**   1. Определите поля, которые будут использоваться для хранения данных о сотрудниках:    * **Имя и фамилия**.    * **Дата приема на работу**.    * **Должность**.    * **Отдел**.    * **Зарплата**.    * **Дата последнего повышения**.    * **Контактная информация**. 2. Создайте таблицу в электронных таблицах (например, Google Sheets, Microsoft Excel или LibreOffice Calc). 3. Заполните таблицу данными для 10 вымышленных сотрудников.   **Пример структуры базы данных:**   | **Имя и Фамилия** | **Дата приема на работу** | **Должность** | **Отдел** | **Зарплата** | **Дата последнего повышения** | **Контактная информация** | | --- | --- | --- | --- | --- | --- | --- | | Иван Иванов | 01.02.2020 | Менеджер | Продажи | 50 000 | 01.03.2022 | ivanov@example.com | | Ольга Петрова | 15.08.2018 | Аналитик | Финансы | 70 000 | 01.06.2021 | petrova@example.com | | Сергей Ковалев | 10.01.2019 | Инженер | IT | 80 000 | 15.05.2023 | kovalev@example.com |   **Задание 2: Фильтрация и сортировка данных в базе данных**  **Цель:**  Научиться использовать функции сортировки и фильтрации для работы с базой данных сотрудников.  **Описание:**  Используя функции фильтрации и сортировки, вам нужно организовать базу данных для быстрого поиска нужной информации. Например, найти сотрудников, получивших повышение за последний год, или отсортировать их по зарплате.  **Этапы выполнения:**   1. **Фильтрация**:    * Отфильтруйте сотрудников, у которых зарплата выше 60 000 рублей.    * Найдите сотрудников, которые были приняты на работу в 2020 году. 2. **Сортировка**:    * Отсортируйте сотрудников по дате приема на работу в порядке возрастания.    * Отсортируйте сотрудников по отделам или по зарплате в порядке убывания.   **Пример после сортировки по дате приема на работу:**   | **Имя и Фамилия** | **Дата приема на работу** | **Должность** | **Отдел** | **Зарплата** | **Дата последнего повышения** | **Контактная информация** | | --- | --- | --- | --- | --- | --- | --- | | Ольга Петрова | 15.08.2018 | Аналитик | Финансы | 70 000 | 01.06.2021 | petrova@example.com | | Сергей Ковалев | 10.01.2019 | Инженер | IT | 80 000 | 15.05.2023 | kovalev@example.com | | Иван Иванов | 01.02.2020 | Менеджер | Продажи | 50 000 | 01.03.2022 | ivanov@example.com |   **Задание 3: Поиск и анализ данных в базе данных**  **Цель:**  Научиться искать данные в базе и анализировать их для решения различных HR-задач.  **Описание:**  Вам нужно использовать функции поиска и анализа, чтобы ответить на различные вопросы, возникающие у HR-специалиста.  **Этапы выполнения:**   1. **Поиск по дате**:    * Найдите всех сотрудников, которые были приняты на работу за последние 2 года. 2. **Поиск по ключевым словам**:    * Найдите сотрудников, работающих в отделе "Продажи". 3. **Анализ данных**:    * Подсчитайте среднюю зарплату сотрудников компании.    * Подсчитайте количество сотрудников в каждом отделе.   **Пример анализа:**   * Средняя зарплата в компании: 66 666 руб. * Количество сотрудников в отделе "IT": 3.   **Задание 4: Визуализация данных**  **Цель:**  Научиться создавать диаграммы для наглядного представления данных.  **Описание:**  Используйте данные вашей базы для создания графиков и диаграмм, чтобы визуализировать распределение сотрудников по отделам или их зарплату.  **Этапы выполнения:**   1. **Создание диаграммы**:    * Постройте столбчатую диаграмму, которая покажет количество сотрудников в каждом отделе. 2. **Анализ зарплат**:    * Постройте круговую диаграмму, показывающую распределение зарплат по сотрудникам.   **Пример диаграммы:**   * **Количество сотрудников в отделах**:   **Оценка:**   * Умение создавать базы данных в электронных таблицах. * Способность использовать функции сортировки и фильтрации для поиска данных. * Навыки анализа данных и создания визуализаций. * Логичное и структурированное использование электронных таблиц для работы с информацией.   **Результат:**  Учащиеся научатся создавать и управлять базами данных, проводить анализ информации и использовать инструменты визуализации данных в электронных таблицах. Эти навыки напрямую применимы в профессиональной деятельности HR-специалиста, а также в других профессиях, связанных с управлением данными. | | |
|  | Методы поиска информации | 9.2.2.3 осуществлять поиск, сортировку и фильтрацию данных |
| **Профессия журналиста**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  Журналист — это профессионал, который собирает, анализирует и представляет информацию для широкой аудитории. Один из ключевых навыков журналиста — это умение находить, сортировать и фильтровать информацию, чтобы предоставить наиболее актуальные и достоверные данные. В этом задании вы научитесь методам поиска, сортировки и фильтрации данных, основываясь на примере журналистской работы в Казахстане.  **Задание 1: Поиск информации по казахстанским источникам**  **Цель:**  Научиться использовать различные методы поиска информации в Интернете и других источниках.  **Описание:**  Представьте, что вы журналист в Казахстане, который собирает информацию для статьи на тему «Экономическое развитие регионов Казахстана». Ваша задача — найти актуальную информацию с помощью различных методов поиска.  **Этапы выполнения:**   1. **Интернет-поиск**:    * Используйте поисковые системы (например, Google) для поиска статистики и отчетов по экономическому развитию регионов Казахстана.    * Пример поискового запроса: «экономическое развитие регионов Казахстана 2023». 2. **Официальные источники**:    * Найдите информацию на официальных сайтах, таких как:      + **Комитет по статистике Министерства национальной экономики РК** (<https://stat.gov.kz>).      + **Министерство экономики Казахстана** (<https://economy.gov.kz>). 3. **Поиск в новостных порталах**:    * Используйте казахстанские новостные сайты, такие как **Kazinform** или **Tengrinews**, чтобы найти последние новости и аналитические статьи по экономике регионов. 4. **Поиск в социальных сетях**:    * Используйте социальные сети (например, Twitter, Facebook) для поиска мнений экспертов и дискуссий по теме.   **Задание 2: Сортировка и фильтрация данных по регионам Казахстана**  **Цель:**  Научиться сортировать и фильтровать данные для получения актуальной информации.  **Описание:**  Вы собрали данные о развитии экономики различных регионов Казахстана, и теперь ваша задача — провести сортировку и фильтрацию данных, чтобы анализировать их.  **Этапы выполнения:**   1. **Создание таблицы**:    * Соберите данные по экономическим показателям для 5 регионов Казахстана (например, ВВП региона, уровень безработицы, инвестиции в регион). 2. **Пример структуры таблицы**:  | **Регион** | **ВВП (млрд тг)** | **Уровень безработицы (%)** | **Инвестиции в регион (млрд тг)** | | --- | --- | --- | --- | | Алматы | 12 000 | 5.5 | 1500 | | Нур-Султан | 10 000 | 4.8 | 1200 | | Карагандинская | 8 500 | 6.2 | 800 | | Атырауская | 15 000 | 3.5 | 2000 | | Туркестанская | 7 500 | 7.0 | 600 |  1. **Фильтрация**:    * Используя фильтры в электронных таблицах, отобразите только регионы с уровнем безработицы ниже 6%. 2. **Сортировка**:    * Отсортируйте регионы по объему инвестиций в порядке убывания.    * Отсортируйте регионы по ВВП в порядке возрастания.   **Пример после сортировки по инвестициям:**   | **Регион** | **ВВП (млрд тг)** | **Уровень безработицы (%)** | **Инвестиции в регион (млрд тг)** | | --- | --- | --- | --- | | Атырауская | 15 000 | 3.5 | 2000 | | Алматы | 12 000 | 5.5 | 1500 | | Нур-Султан | 10 000 | 4.8 | 1200 | | Карагандинская | 8 500 | 6.2 | 800 | | Туркестанская | 7 500 | 7.0 | 600 |   **Задание 3: Поиск и анализ данных по городам Казахстана**  **Цель:**  Научиться анализировать данные, найденные через поиск и фильтрацию, для написания журналистского материала.  **Описание:**  Теперь ваша задача — провести более глубокий анализ собранных данных и сделать выводы о развитии экономики в разных регионах Казахстана.  **Этапы выполнения:**   1. **Поиск по ключевым словам**:    * Найдите все регионы с ВВП выше 10 000 млрд тенге. 2. **Анализ данных**:    * Определите, в каких регионах наибольшие инвестиции и самый низкий уровень безработицы. Сформулируйте выводы: как это связано с экономическим ростом региона? 3. **Визуализация данных**:    * Постройте столбчатую диаграмму, показывающую объем ВВП в каждом регионе.    * Постройте график, показывающий уровень безработицы в разных регионах.   **Пример диаграммы:**   * **ВВП регионов Казахстана**:   **Оценка:**   * Умение эффективно искать информацию из разных источников (официальные сайты, новостные порталы, социальные сети). * Способность сортировать и фильтровать данные по заданным критериям. * Навыки анализа и визуализации данных для более наглядного представления информации. * Логичное и структурированное использование данных для решения журналистских задач.   **Результат:**  Учащиеся научатся использовать методы поиска, сортировки и фильтрации данных, что является важным навыком для журналистов и других профессионалов, работающих с большими объемами информации. Они смогут не только находить нужные данные, но и анализировать их для подготовки материалов, связанных с актуальными событиями в Казахстане.  **Сотрудник полиции**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  В работе сотрудника полиции важную роль играет умение эффективно искать, сортировать и фильтровать информацию. Это может включать анализ данных для расследований, поиск информации по базам данных и сортировку данных для отчетов. Ваша задача — научиться использовать методы поиска и сортировки данных, применяя их в контексте работы полиции.  **Задание 1: Поиск информации по базе данных правонарушений**  **Цель:**  Научиться использовать методы поиска информации в базе данных, подобной той, что используют полицейские для расследований.  **Описание:**  Представьте, что вы сотрудник полиции и вам нужно найти информацию по правонарушениям в базе данных для анализа и отчетности. В базе данных содержатся данные о нарушениях закона за последние несколько лет, включая дату, место, тип правонарушения и данные правонарушителя.  **Этапы выполнения:**   1. **Создание таблицы**:    * Создайте таблицу с данными о правонарушениях, которая включает следующие поля:      + **Дата правонарушения**.      + **Место (город/район)**.      + **Тип правонарушения** (например, кража, мошенничество, хулиганство).      + **Возраст правонарушителя**. 2. **Пример структуры таблицы**:  |  |  |  |  | | --- | --- | --- | --- | | **Дата** | **Место** | **Тип правонарушения** | **Возраст правонарушителя** | | 01.02.2023 | Алматы | Кража | 23 | | 15.05.2023 | Нур-Султан | Мошенничество | 35 | | 20.03.2023 | Шымкент | Хулиганство | 17 | | 12.04.2023 | Караганда | Кража | 28 |   **Задание 2: Сортировка и фильтрация данных по правонарушениям**  **Цель:**  Научиться сортировать и фильтровать данные для быстрого поиска нужной информации.  **Описание:**  После создания базы данных с информацией о правонарушениях вам нужно научиться сортировать и фильтровать данные, чтобы находить информацию по конкретным запросам, например, по типам преступлений или возрасту правонарушителей.  **Этапы выполнения:**   1. **Фильтрация**:    * Используйте функцию фильтрации, чтобы найти все случаи краж, совершенных за последний год.    * Отфильтруйте данные по возрасту правонарушителей, чтобы отобразить только тех, кто младше 25 лет. 2. **Сортировка**:    * Отсортируйте правонарушения по дате, начиная с самых свежих событий.    * Отсортируйте данные по типу правонарушения, сгруппировав все кражи, мошенничества и хулиганства вместе.   **Пример после сортировки по дате правонарушений:**   |  |  |  |  | | --- | --- | --- | --- | | **Дата** | **Место** | **Тип правонарушения** | **Возраст правонарушителя** | | 15.05.2023 | Нур-Султан | Мошенничество | 35 | | 12.04.2023 | Караганда | Кража | 28 | | 20.03.2023 | Шымкент | Хулиганство | 17 | | 01.02.2023 | Алматы | Кража | 23 |   **Задание 3: Поиск и анализ данных по правонарушениям**  **Цель:**  Научиться эффективно искать и анализировать информацию для выявления закономерностей в правонарушениях.  **Описание:**  Вы должны использовать данные, чтобы анализировать распространенность различных типов правонарушений в разных городах и выявлять закономерности, например, где чаще совершаются кражи или в каком возрасте чаще всего происходят нарушения закона.  **Этапы выполнения:**   1. **Поиск по типу правонарушения**:    * Найдите все случаи мошенничества за последний год. 2. **Анализ данных**:    * Проанализируйте данные, чтобы выявить, в каком городе чаще всего совершаются кражи.    * Подсчитайте, в каком возрасте правонарушители чаще всего совершают преступления. 3. **Визуализация данных**:    * Постройте график, показывающий количество правонарушений по типам (кражи, мошенничество, хулиганство).    * Постройте диаграмму, показывающую распределение правонарушений по возрасту.   **Пример диаграммы:**   * **Количество правонарушений по типам**:   **Оценка:**   * Умение осуществлять поиск информации в базах данных. * Способность сортировать и фильтровать данные по заданным критериям. * Навыки анализа данных и построения визуализаций для более наглядного представления информации. * Логичное и структурированное использование данных для решения задач полицейского.   **Результат:**  Учащиеся научатся использовать методы поиска, сортировки и фильтрации данных, что является важным навыком в работе полиции. Они смогут эффективно находить нужные данные, анализировать их и использовать для создания отчетов и визуализаций, что поможет в расследованиях и принятии решений. | | |
|  | Сортировка и фильтрация данных | 9.2.2.3 осуществлять поиск, сортировку и фильтрацию данных |
| **Профессия аналитик данных**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  Аналитик данных — это специалист, который обрабатывает и анализирует большие массивы данных для выявления закономерностей, трендов и аномалий, что помогает компаниям принимать правильные решения. В этом задании вы научитесь использовать методы сортировки и фильтрации данных, применяя их в реальных задачах аналитика данных.  **Задание 1: Сортировка данных по продажам**  **Цель:**  Научиться сортировать данные в таблице для анализа продаж различных товаров.  **Описание:**  Представьте, что вы аналитик данных в компании, которая продает различные товары. Вам нужно проанализировать данные по продажам, чтобы понять, какие товары пользуются наибольшим спросом, а какие продаются хуже всего.  **Этапы выполнения:**   1. **Создание таблицы**:    * Введите данные о продажах за последний месяц, включая:      + **Название товара**.      + **Количество проданных единиц**.      + **Цена за единицу**.      + **Общая выручка** (количество проданных единиц \* цена за единицу). 2. **Пример структуры таблицы**:  | **Название товара** | **Количество проданных единиц** | **Цена за единицу (тг)** | **Общая выручка (тг)** | | --- | --- | --- | --- | | Смартфон | 100 | 150 000 | 15 000 000 | | Планшет | 50 | 100 000 | 5 000 000 | | Ноутбук | 30 | 200 000 | 6 000 000 | | Наушники | 200 | 10 000 | 2 000 000 | | Телевизор | 20 | 300 000 | 6 000 000 |  1. **Сортировка данных**:    * Отсортируйте товары по количеству проданных единиц в порядке убывания, чтобы определить самые популярные товары.    * Отсортируйте товары по общей выручке, чтобы определить, какой товар принес компании наибольшую прибыль.   **Пример после сортировки по количеству проданных единиц:**   | **Название товара** | **Количество проданных единиц** | **Цена за единицу (тг)** | **Общая выручка (тг)** | | --- | --- | --- | --- | | Наушники | 200 | 10 000 | 2 000 000 | | Смартфон | 100 | 150 000 | 15 000 000 | | Планшет | 50 | 100 000 | 5 000 000 | | Ноутбук | 30 | 200 000 | 6 000 000 | | Телевизор | 20 | 300 000 | 6 000 000 |   **Задание 2: Фильтрация данных по выручке**  **Цель:**  Научиться фильтровать данные для анализа продаж товаров, которые приносят наибольшую прибыль.  **Описание:**  Теперь ваша задача — отфильтровать товары, которые принесли компании наибольшую прибыль, и сосредоточиться только на тех товарах, которые генерируют выручку выше 5 000 000 тенге.  **Этапы выполнения:**   1. **Фильтрация данных**:    * Примените фильтр, чтобы отобразить только те товары, общая выручка которых превышает 5 000 000 тенге. 2. **Анализ результатов**:    * Проанализируйте отфильтрованные данные: сколько товаров принесло компании наибольшую прибыль, и какие из них следует продвигать более активно.   **Пример после фильтрации по общей выручке:**   | **Название товара** | **Количество проданных единиц** | **Цена за единицу (тг)** | **Общая выручка (тг)** | | --- | --- | --- | --- | | Смартфон | 100 | 150 000 | 15 000 000 | | Ноутбук | 30 | 200 000 | 6 000 000 | | Телевизор | 20 | 300 000 | 6 000 000 |   **Задание 3: Поиск трендов в данных**  **Цель:**  Научиться находить тренды и закономерности в данных для прогнозирования продаж.  **Описание:**  Теперь вам нужно найти, какие товары чаще всего продаются в зависимости от их цены и количества проданных единиц. Это поможет вам прогнозировать, какие товары могут быть востребованы в будущем.  **Этапы выполнения:**   1. **Поиск по ключевым словам**:    * Найдите все товары, цена которых выше 100 000 тенге. Это поможет сосредоточиться на более дорогих товарах. 2. **Анализ трендов**:    * Сравните количество проданных товаров и их цену: есть ли закономерность, что более дорогие товары продаются хуже, чем более дешевые? 3. **Визуализация данных**:    * Постройте график, показывающий зависимость количества проданных единиц от цены за единицу.   **Пример графика:**   * **Зависимость количества проданных товаров от цены**:   **Оценка:**   * Умение сортировать данные для анализа продаж. * Способность фильтровать данные по заданным критериям для выявления ключевых товаров. * Навыки поиска закономерностей и трендов в данных. * Логичное и структурированное использование методов сортировки и фильтрации для анализа данных.   **Результат:**  Учащиеся научатся сортировать и фильтровать данные, анализировать продажи товаров и находить закономерности в данных, что является важным навыком в работе аналитика данных. Эти знания помогут применять методы сортировки и фильтрации для анализа данных в реальных рабочих ситуациях.  **Задание 2**  **Профессия врач**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  В профессии врача сортировка и фильтрация данных являются важными навыками для работы с медицинскими картами пациентов, результатами анализов, а также статистическими данными по здоровью населения. В этом задании учащиеся научатся использовать методы сортировки и фильтрации данных, которые необходимы врачам для анализа информации о пациентах.  **Задание 1: Сортировка данных по пациентам**  **Цель:**  Научиться сортировать данные пациентов для эффективного ведения медицинских карт и планирования лечения.  **Описание:**  Представьте, что вы врач в поликлинике и вам нужно работать с базой данных пациентов. Вам требуется проанализировать данные для лучшего понимания состояния здоровья пациентов и их текущих назначений.  **Этапы выполнения:**   1. **Создание таблицы**:    * Введите данные о пациентах, включая:      + **Имя пациента**.      + **Возраст**.      + **Диагноз**.      + **Дата последнего визита**.      + **Назначенное лечение**. 2. **Пример структуры таблицы**:  | **Имя пациента** | **Возраст** | **Диагноз** | **Дата последнего визита** | **Назначенное лечение** | | --- | --- | --- | --- | --- | | Иван Иванов | 45 | Гипертония | 01.09.2023 | Лекарства | | Ольга Петрова | 32 | Астма | 15.08.2023 | Ингаляторы | | Сергей Ковалев | 60 | Сахарный диабет | 20.09.2023 | Диета, лекарства | | Анна Смирнова | 28 | Простуда | 05.09.2023 | Симптоматическое лечение |  1. **Сортировка данных**:    * Отсортируйте пациентов по дате последнего визита, чтобы узнать, кто из них требует повторного осмотра.    * Отсортируйте пациентов по возрасту, чтобы выделить возрастные группы, требующие особого внимания (например, пожилые люди с хроническими заболеваниями).   **Пример после сортировки по дате последнего визита:**   | **Имя пациента** | **Возраст** | **Диагноз** | **Дата последнего визита** | **Назначенное лечение** | | --- | --- | --- | --- | --- | | Сергей Ковалев | 60 | Сахарный диабет | 20.09.2023 | Диета, лекарства | | Анна Смирнова | 28 | Простуда | 05.09.2023 | Симптоматическое лечение | | Иван Иванов | 45 | Гипертония | 01.09.2023 | Лекарства | | Ольга Петрова | 32 | Астма | 15.08.2023 | Ингаляторы |   **Задание 2: Фильтрация данных по заболеваниям**  **Цель:**  Научиться фильтровать данные для анализа информации по пациентам с определенными заболеваниями.  **Описание:**  Вам необходимо фильтровать данные о пациентах, чтобы сосредоточиться на определенных группах, например, всех пациентах с хроническими заболеваниями или тех, кто нуждается в регулярном медицинском наблюдении.  **Этапы выполнения:**   1. **Фильтрация данных**:    * Используйте функцию фильтрации, чтобы отобразить только пациентов с хроническими заболеваниями (например, гипертония, диабет).    * Отфильтруйте пациентов младше 40 лет, чтобы проанализировать состояние здоровья более молодой части пациентов. 2. **Анализ результатов**:    * Проведите анализ: сколько пациентов имеют хронические заболевания и нуждаются в регулярном наблюдении?    * Проанализируйте, сколько пациентов младше 40 лет обращались за помощью с серьезными заболеваниями.   **Пример после фильтрации по хроническим заболеваниям:**   | **Имя пациента** | **Возраст** | **Диагноз** | **Дата последнего визита** | **Назначенное лечение** | | --- | --- | --- | --- | --- | | Иван Иванов | 45 | Гипертония | 01.09.2023 | Лекарства | | Сергей Ковалев | 60 | Сахарный диабет | 20.09.2023 | Диета, лекарства |   **Задание 3: Поиск и анализ данных по пациентам**  **Цель:**  Научиться использовать методы поиска и анализа данных для лучшего понимания медицинских показателей пациентов.  **Описание:**  Теперь ваша задача — проанализировать данные пациентов, чтобы понять, какие группы пациентов требуют более внимательного наблюдения и планировать их дальнейшее лечение.  **Этапы выполнения:**   1. **Поиск по ключевым словам**:    * Найдите всех пациентов, у которых диагностированы хронические заболевания. 2. **Анализ данных**:    * Подсчитайте средний возраст пациентов с хроническими заболеваниями.    * Определите, сколько пациентов посещали врача за последние 2 месяца. 3. **Визуализация данных**:    * Постройте столбчатую диаграмму, которая покажет количество пациентов с разными диагнозами.    * Постройте график, показывающий, как изменяется частота визитов пациентов в зависимости от их возраста.   **Пример графика:**   * **Количество пациентов с разными диагнозами**:   **Оценка:**   * Умение осуществлять сортировку данных для анализа состояния здоровья пациентов. * Способность фильтровать данные для выделения групп пациентов с определенными заболеваниями. * Навыки поиска закономерностей и трендов в данных пациентов. * Логичное и структурированное использование методов сортировки и фильтрации для анализа медицинских данных.   **Результат:**  Учащиеся научатся сортировать и фильтровать медицинские данные, что является важным навыком для врачей при работе с пациентами. Эти навыки помогут анализировать состояние здоровья пациентов, находить нужную информацию и принимать обоснованные решения на основе данных, что сделает медицинскую практику более эффективной | | |
|  | Работа с базой данных | 9.2.2.2 создавать базу данных в электронной таблице; 9.2.2.3 осуществлять поиск, сортировку и фильтрацию данных |
| **Профессия педагог**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  В профессии педагога сортировка и фильтрация данных — важные навыки для работы с информацией о учениках, учебными достижениями, успеваемостью и посещаемостью. В этом задании учащиеся научатся работать с данными, используя методы сортировки и фильтрации, применяя их в контексте работы учителя в школе.  **Задание 1: Сортировка данных по успеваемости учеников**  **Цель:**  Научиться сортировать данные о результатах учащихся для анализа успеваемости в классе.  **Описание:**  Представьте, что вы учитель в школе Казахстана и вам нужно проанализировать результаты контрольной работы учеников. Ваша задача — отсортировать данные по успеваемости и выявить сильные и слабые стороны класса.  **Этапы выполнения:**   1. **Создание таблицы**:    * Введите данные о результатах контрольной работы учеников, включая:      + **Имя ученика**.      + **Оценка за контрольную работу** (по 5-балльной системе).      + **Посещаемость** (в процентах).      + **Предмет** (например, математика, казахский язык, информатика). 2. **Пример структуры таблицы**:  | **Имя ученика** | **Оценка за контрольную работу** | **Посещаемость (%)** | **Предмет** | | --- | --- | --- | --- | | Айсулу Нурлан | 4 | 95% | Математика | | Данияр Сатпаев | 5 | 100% | Информатика | | Алия Ахметова | 3 | 80% | Казахский язык | | Нуржан Сейтов | 2 | 60% | Математика |  1. **Сортировка данных**:    * Отсортируйте учеников по оценкам в порядке убывания, чтобы увидеть лучших и слабейших учеников.    * Отсортируйте данные по посещаемости, чтобы увидеть, как посещаемость влияет на успеваемость.   **Пример после сортировки по оценкам:**   | **Имя ученика** | **Оценка за контрольную работу** | **Посещаемость (%)** | **Предмет** | | --- | --- | --- | --- | | Данияр Сатпаев | 5 | 100% | Информатика | | Айсулу Нурлан | 4 | 95% | Математика | | Алия Ахметова | 3 | 80% | Казахский язык | | Нуржан Сейтов | 2 | 60% | Математика |   **Задание 2: Фильтрация данных по предметам**  **Цель:**  Научиться фильтровать данные для анализа успеваемости по разным предметам.  **Описание:**  Теперь вам нужно сосредоточиться на результатах контрольных работ по отдельным предметам, чтобы определить, по каким предметам ученики справляются лучше, а где им требуется дополнительная помощь.  **Этапы выполнения:**   1. **Фильтрация данных**:    * Используйте функцию фильтрации, чтобы отобразить только учеников, которые писали контрольную работу по математике.    * Отфильтруйте учеников, у которых оценка ниже 3 баллов, чтобы выделить тех, кому требуется дополнительная поддержка. 2. **Анализ результатов**:    * Проанализируйте данные: сколько учеников получили оценку выше 4 баллов по каждому предмету?    * Определите, на каких предметах нужно провести дополнительную работу с учениками.   **Пример после фильтрации по предмету "Математика":**   | **Имя ученика** | **Оценка за контрольную работу** | **Посещаемость (%)** | **Предмет** | | --- | --- | --- | --- | | Айсулу Нурлан | 4 | 95% | Математика | | Нуржан Сейтов | 2 | 60% | Математика |   **Задание 3: Поиск и анализ данных по успеваемости**  **Цель:**  Научиться находить и анализировать данные для выявления трендов в успеваемости учеников.  **Описание:**  Теперь вам нужно провести анализ данных по успеваемости учеников в зависимости от их посещаемости, чтобы понять, как посещаемость влияет на их результаты.  **Этапы выполнения:**   1. **Поиск по ключевым словам**:    * Найдите всех учеников, у которых посещаемость ниже 80%, чтобы понять, как она влияет на успеваемость. 2. **Анализ данных**:    * Подсчитайте среднюю оценку учеников с посещаемостью выше 90%.    * Определите, есть ли зависимость между посещаемостью и успеваемостью учеников. 3. **Визуализация данных**:    * Постройте диаграмму, показывающую, как посещаемость влияет на успеваемость учеников.    * Постройте график, показывающий распределение оценок по разным предметам.   **Пример диаграммы:**   * **Зависимость посещаемости и успеваемости учеников**:   **Оценка:**   * Умение сортировать данные для анализа успеваемости учеников. * Способность фильтровать данные для выделения групп учеников по предметам и оценкам. * Навыки поиска и анализа закономерностей в данных, связанных с посещаемостью и успеваемостью. * Логичное и структурированное использование методов сортировки и фильтрации для решения педагогических задач.   **Результат:**  Учащиеся научатся сортировать и фильтровать данные, что является важным навыком для педагога при анализе успеваемости учеников. Эти задания помогут им развить навыки поиска данных, анализа успеваемости и планирования работы с классом, что сделает учебный процесс более эффективным.  **Задание 2**  **Профессия строитель, прораб**  **Цель урока:**   * **9.2.2.3** Осуществлять поиск, сортировку и фильтрацию данных.   **Описание:**  Профессия строителя требует умения работать с данными о строительных материалах, затратах на строительство, графиках выполнения работ и других важных аспектах. В этом задании учащиеся научатся сортировать и фильтровать данные, используя примеры из строительной сферы Казахстана, такие как анализ затрат на строительные проекты или отслеживание использования материалов.  **Задание 1: Сортировка данных по строительным материалам**  **Цель:**  Научиться сортировать данные о строительных материалах для анализа затрат и планирования закупок.  **Описание:**  Представьте, что вы работаете в строительной компании в Казахстане и вам нужно проанализировать данные о материалах, использованных на различных строительных объектах. Ваша задача — отсортировать эти данные для эффективного управления ресурсами.  **Этапы выполнения:**   1. **Создание таблицы**:    * Введите данные о строительных материалах, включая:      + **Название материала** (например, бетон, кирпич, арматура).      + **Количество использованных единиц** (например, кубометры или штуки).      + **Стоимость за единицу**.      + **Общая стоимость материала** (рассчитывается как количество \* стоимость за единицу).      + **Строительный объект** (например, жилой дом в Алматы или торговый центр в Нур-Султане). 2. **Пример структуры таблицы**:  | **Название материала** | **Количество (единиц)** | **Стоимость за единицу (тг)** | **Общая стоимость (тг)** | **Строительный объект** | | --- | --- | --- | --- | --- | | Бетон | 100 | 25 000 | 2 500 000 | Жилой дом в Алматы | | Кирпич | 5000 | 120 | 600 000 | Торговый центр в Нур-Султане | | Арматура | 200 | 15 000 | 3 000 000 | Жилой дом в Алматы | | Дерево | 300 | 10 000 | 3 000 000 | Школа в Караганде |  1. **Сортировка данных**:    * Отсортируйте материалы по общей стоимости в порядке убывания, чтобы увидеть, какие материалы требуют наибольших затрат.    * Отсортируйте материалы по строительным объектам, чтобы лучше понять, какие материалы используются на каждом объекте.   **Пример после сортировки по общей стоимости:**   | **Название материала** | **Количество (единиц)** | **Стоимость за единицу (тг)** | **Общая стоимость (тг)** | **Строительный объект** | | --- | --- | --- | --- | --- | | Арматура | 200 | 15 000 | 3 000 000 | Жилой дом в Алматы | | Дерево | 300 | 10 000 | 3 000 000 | Школа в Караганде | | Бетон | 100 | 25 000 | 2 500 000 | Жилой дом в Алматы | | Кирпич | 5000 | 120 | 600 000 | Торговый центр в Нур-Султане |   **Задание 2: Фильтрация данных по строительным объектам**  **Цель:**  Научиться фильтровать данные для анализа использования материалов на разных объектах.  **Описание:**  Теперь вам нужно сосредоточиться на анализе данных по строительным объектам, чтобы увидеть, какие материалы использовались на каждом объекте и каковы их затраты.  **Этапы выполнения:**   1. **Фильтрация данных**:    * Используйте функцию фильтрации, чтобы отобразить только те материалы, которые использовались на строительстве жилого дома в Алматы.    * Отфильтруйте материалы, стоимость которых превышает 2 000 000 тенге, чтобы увидеть самые дорогие материалы. 2. **Анализ результатов**:    * Проанализируйте данные: какие материалы используются чаще всего и на каких объектах?    * Какие материалы являются самыми дорогими и на каких объектах их используют?   **Пример после фильтрации по строительному объекту «Жилой дом в Алматы»:**   | **Название материала** | **Количество (единиц)** | **Стоимость за единицу (тг)** | **Общая стоимость (тг)** | **Строительный объект** | | --- | --- | --- | --- | --- | | Бетон | 100 | 25 000 | 2 500 000 | Жилой дом в Алматы | | Арматура | 200 | 15 000 | 3 000 000 | Жилой дом в Алматы |   **Задание 3: Поиск и анализ данных по затратам на материалы**  **Цель:**  Научиться находить и анализировать данные для выявления затрат на материалы по разным объектам и составления бюджета.  **Описание:**  Теперь вам нужно проанализировать затраты на строительные материалы, чтобы составить бюджет и понять, какие материалы и объекты требуют наибольших затрат.  **Этапы выполнения:**   1. **Поиск по ключевым словам**:    * Найдите все материалы, затраты на которые превышают 1 000 000 тенге, чтобы выявить наиболее дорогие элементы. 2. **Анализ данных**:    * Подсчитайте общие затраты на каждый объект (суммируя стоимость всех материалов, использованных на объекте).    * Определите, какой объект требует наибольших затрат на материалы. 3. **Визуализация данных**:    * Постройте диаграмму, показывающую распределение затрат на материалы по строительным объектам.    * Постройте график, показывающий зависимость затрат от количества использованных материалов.   **Пример диаграммы:**   * **Затраты на материалы по строительным объектам**:   **Оценка:**   * Умение сортировать данные для анализа затрат на строительные материалы. * Способность фильтровать данные для выделения материалов по объектам и затратам. * Навыки поиска закономерностей и анализа данных для составления бюджета строительства. * Логичное и структурированное использование методов сортировки и фильтрации данных для планирования строительных проектов.   **Результат:**  Учащиеся научатся сортировать и фильтровать данные, что является важным навыком для строителя при планировании и управлении строительными проектами. Эти задания помогут им развить навыки анализа затрат на материалы, распределения ресурсов и планирования бюджета, что делает процесс управления строительными проектами более эффективным. | | |
| **Раздел 4. Программирование алгоритмов на языке программирования Python (пайтон)** | | |
|  | Одномерный массив | 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов |
| **Профессия финансовый аналитик**  **Цели урока:**   * **9.3.3.1** Создавать программы на языке программирования Python с использованием одномерных массивов.   **Описание:**  Финансовый аналитик — это специалист, который анализирует экономические данные и тренды для прогнозирования и принятия решений. В своей работе аналитик часто использует массивы данных для расчетов и анализа. В этом задании вы научитесь создавать программы на языке Python для обработки одномерных массивов, используя реальные примеры экономических данных Казахстана.  **Задание 1: Анализ цен на нефть в Казахстане**  **Цель:**  Научиться работать с одномерными массивами в Python для анализа экономических данных.  **Описание:**  Представьте, что вы финансовый аналитик в Казахстане, который анализирует цены на нефть за последние 12 месяцев. Вам нужно написать программу на Python, которая будет рассчитывать среднюю цену, наибольшую и наименьшую цены, а также выведет все данные на экран.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о ценах на нефть за последние 12 месяцев в одномерный массив (в тенге за баррель). 2. **Пример массива**:   python  Копировать код  prices = [28000, 29000, 29500, 30000, 31000, 30500, 29500, 29000, 28500, 29500, 30000, 30500]   1. **Задание**:    * Напишите программу на Python, которая выполнит следующие задачи:      + Выведет все цены на нефть.      + Рассчитает и выведет среднюю цену за год.      + Найдет и выведет наибольшую и наименьшую цену. 2. **Пример кода на Python**:   python  Копировать код  prices = [28000, 29000, 29500, 30000, 31000, 30500, 29500, 29000, 28500, 29500, 30000, 30500]  # Вывод всех цен  print("Цены на нефть за 12 месяцев:", prices)  # Расчет средней цены  avg\_price = sum(prices) / len(prices)  print("Средняя цена на нефть за год:", avg\_price)  # Поиск наибольшей и наименьшей цены  max\_price = max(prices)  min\_price = min(prices)  print("Наибольшая цена на нефть:", max\_price)  print("Наименьшая цена на нефть:", min\_price)   1. **Результат выполнения программы**:   less  Копировать код  Цены на нефть за 12 месяцев: [28000, 29000, 29500, 30000, 31000, 30500, 29500, 29000, 28500, 29500, 30000, 30500]  Средняя цена на нефть за год: 29625.0  Наибольшая цена на нефть: 31000  Наименьшая цена на нефть: 28000  **Задание 2: Прогнозирование доходов компании на основе цен на нефть**  **Цель:**  Научиться применять одномерные массивы для прогнозирования доходов компании на основе экономических данных.  **Описание:**  Теперь ваша задача — рассчитать предполагаемые доходы компании, которая производит 10 000 баррелей нефти в месяц. Вы используете массив цен на нефть, чтобы вычислить доход за каждый месяц и общий доход за год.  **Этапы выполнения:**   1. **Создание массива доходов**:    * Используйте массив цен на нефть и объем производства (10 000 баррелей) для расчета доходов за каждый месяц. 2. **Задание**:    * Напишите программу на Python, которая:      + Рассчитает доход компании за каждый месяц.      + Найдет общий доход за год.      + Выведет доход за каждый месяц и общий доход на экран. 3. **Пример кода на Python**:   python  Копировать код  prices = [28000, 29000, 29500, 30000, 31000, 30500, 29500, 29000, 28500, 29500, 30000, 30500]  production = 10000 # Объем производства (баррелей в месяц)  # Расчет доходов за каждый месяц  revenues = [price \* production for price in prices]  # Вывод доходов за каждый месяц  print("Доход компании за каждый месяц:", revenues)  # Расчет общего дохода за год  total\_revenue = sum(revenues)  print("Общий доход компании за год:", total\_revenue)   1. **Результат выполнения программы**:   less  Копировать код  Доход компании за каждый месяц: [280000000, 290000000, 295000000, 300000000, 310000000, 305000000, 295000000, 290000000, 285000000, 295000000, 300000000, 305000000]  Общий доход компании за год: 3655000000  **Задание 3: Анализ затрат на строительство объектов в Казахстане**  **Цель:**  Научиться применять одномерные массивы для анализа затрат на строительство.  **Описание:**  Представьте, что вы финансист в строительной компании Казахстана и анализируете затраты на строительство различных объектов. Ваша задача — написать программу на Python, которая поможет проанализировать затраты на строительство нескольких объектов.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о затратах на строительство 5 объектов (в миллионах тенге). 2. **Пример массива**:   python  Копировать код  costs = [500, 750, 900, 650, 800]   1. **Задание**:    * Напишите программу, которая:      + Выведет затраты на каждый объект.      + Найдет объект с наибольшими и наименьшими затратами.      + Рассчитает средние затраты на строительство. 2. **Пример кода на Python**:   python  Копировать код  costs = [500, 750, 900, 650, 800]  # Вывод затрат на каждый объект  print("Затраты на строительство объектов:", costs)  # Поиск наибольших и наименьших затрат  max\_cost = max(costs)  min\_cost = min(costs)  print("Наибольшие затраты:", max\_cost)  print("Наименьшие затраты:", min\_cost)  # Расчет средних затрат  avg\_cost = sum(costs) / len(costs)  print("Средние затраты на строительство:", avg\_cost)   1. **Результат выполнения программы**:   less  Копировать код  Затраты на строительство объектов: [500, 750, 900, 650, 800]  Наибольшие затраты: 900  Наименьшие затраты: 500  Средние затраты на строительство: 720.0  **Оценка:**   * Умение создавать и работать с одномерными массивами в Python. * Способность использовать массивы для анализа реальных данных. * Навыки разработки программ для решения задач, связанных с экономикой и финансами.   **Результат:**  Учащиеся научатся создавать программы на языке Python, используя одномерные массивы для анализа реальных данных, связанных с финансами и строительством в Казахстане. Эти навыки могут быть полезны в таких профессиях, как финансовый аналитик, экономист или строительный менеджер. | | |
|  | Поиск элемента с заданными свойствами | 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов |
| **Профессия менеджера по продажам автомобилей**  **Цели урока:**   * **9.3.3.1** Создавать программы на языке программирования Python с использованием одномерных массивов.   **Описание:**  Менеджер по продажам автомобилей должен уметь анализировать и сравнивать автомобили по их характеристикам: цене, расходу топлива, мощности двигателя и другим параметрам, чтобы предложить клиентам наиболее подходящий вариант. В этом задании учащиеся научатся создавать программы на языке Python для поиска элементов (автомобилей) с заданными свойствами в одномерных массивах.  **Задание 1: Поиск автомобиля по цене**  **Цель:**  Научиться работать с одномерными массивами в Python для поиска элементов с определенными свойствами.  **Описание:**  Представьте, что вы работаете менеджером по продажам автомобилей в Казахстане. У вас есть данные о нескольких автомобилях, и вам нужно найти автомобиль, который соответствует запросам клиента по цене.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о ценах на автомобили (в миллионах тенге) в одномерный массив. 2. **Пример массива**:   python  Копировать код  prices = [7.5, 5.0, 6.2, 8.1, 4.8, 9.0]   1. **Задание**:    * Напишите программу на Python, которая:      + Запрашивает у пользователя максимальную цену, которую он готов потратить на автомобиль.      + Выводит все автомобили, которые стоят меньше или равно этой цене. 2. **Пример кода на Python**:   python  Копировать код  prices = [7.5, 5.0, 6.2, 8.1, 4.8, 9.0]  # Запрос у пользователя максимальной цены  max\_price = float(input("Введите максимальную цену на автомобиль (в млн тг): "))  # Поиск автомобилей по цене  affordable\_cars = [price for price in prices if price <= max\_price]  # Вывод результатов  if affordable\_cars:  print("Автомобили, доступные по вашей цене:", affordable\_cars)  else:  print("Нет автомобилей, соответствующих вашему бюджету.")   1. **Результат выполнения программы**:   less  Копировать код  Введите максимальную цену на автомобиль (в млн тг): 6.0  Автомобили, доступные по вашей цене: [5.0, 4.8]  **Задание 2: Поиск автомобиля по расходу топлива**  **Цель:**  Научиться искать элементы массива с определенными характеристиками с использованием языка Python.  **Описание:**  Теперь клиент интересуется автомобилем с низким расходом топлива. Ваша задача — найти автомобиль с расходом топлива не более заданного значения.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о расходе топлива автомобилей (в литрах на 100 км) в одномерный массив. 2. **Пример массива**:   python  Копировать код  fuel\_consumption = [7.8, 6.5, 5.9, 8.2, 7.0, 6.8]   1. **Задание**:    * Напишите программу, которая:      + Запрашивает у пользователя максимальный расход топлива.      + Выводит автомобили, которые соответствуют этому критерию (расход не больше указанного значения). 2. **Пример кода на Python**:   python  Копировать код  fuel\_consumption = [7.8, 6.5, 5.9, 8.2, 7.0, 6.8]  # Запрос у пользователя максимального расхода топлива  max\_fuel = float(input("Введите максимальный расход топлива (л/100 км): "))  # Поиск автомобилей с подходящим расходом топлива  efficient\_cars = [fuel for fuel in fuel\_consumption if fuel <= max\_fuel]  # Вывод результатов  if efficient\_cars:  print("Автомобили с расходом топлива ниже или равным указанному:", efficient\_cars)  else:  print("Нет автомобилей с таким расходом топлива.")   1. **Результат выполнения программы**:   less  Копировать код  Введите максимальный расход топлива (л/100 км): 7.0  Автомобили с расходом топлива ниже или равным указанному: [6.5, 5.9, 7.0, 6.8]  **Задание 3: Поиск автомобиля по мощности двигателя**  **Цель:**  Научиться искать элементы массива с определенными свойствами с использованием одномерных массивов на Python.  **Описание:**  Теперь ваш клиент хочет купить автомобиль с мощным двигателем. Вам нужно помочь ему найти автомобили, которые имеют мощность двигателя больше заданного значения.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о мощности двигателя автомобилей (в лошадиных силах) в одномерный массив. 2. **Пример массива**:   python  Копировать код  horsepower = [150, 180, 200, 220, 170, 160]   1. **Задание**:    * Напишите программу на Python, которая:      + Запрашивает у пользователя минимальную мощность двигателя.      + Выводит все автомобили, мощность двигателя которых превышает указанное значение. 2. **Пример кода на Python**:   python  Копировать код  horsepower = [150, 180, 200, 220, 170, 160]  # Запрос у пользователя минимальной мощности двигателя  min\_hp = int(input("Введите минимальную мощность двигателя (л.с.): "))  # Поиск автомобилей с мощностью двигателя выше указанной  powerful\_cars = [hp for hp in horsepower if hp > min\_hp]  # Вывод результатов  if powerful\_cars:  print("Автомобили с мощностью двигателя выше указанной:", powerful\_cars)  else:  print("Нет автомобилей с такой мощностью двигателя.")   1. **Результат выполнения программы**:   less  Копировать код  Введите минимальную мощность двигателя (л.с.): 170  Автомобили с мощностью двигателя выше указанной: [180, 200, 220]  **Оценка:**   * Умение создавать программы на языке Python с использованием одномерных массивов. * Способность применять методы поиска для нахождения элементов с заданными свойствами. * Навыки разработки программ для решения задач, связанных с анализом данных в профессиональной деятельности.   **Результат:**  Учащиеся научатся создавать программы на языке Python с использованием одномерных массивов для поиска элементов с определенными свойствами, что является важным навыком для различных профессий, включая менеджера по продажам автомобилей, аналитика данных и других специалистов, работающих с массивами данных. | | |
|  | Перестановка элементов | 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов |
| **Профессия логиста**  **Цель урока:**   * **9.3.3.1** Создавать программы на языке программирования Python с использованием одномерных массивов.   **Описание:**  В профессии логиста важно оптимизировать порядок доставки товаров или ресурсов, что может включать перестановку элементов (грузов, маршрутов) для улучшения эффективности работы. В этом задании учащиеся научатся создавать программы на языке Python для перестановки элементов одномерных массивов, применяя это к задачам логистики в Казахстане.  **Задание 1: Перестановка маршрутов доставки**  **Цель:**  Научиться работать с одномерными массивами в Python для перестановки элементов, моделируя изменение порядка доставки грузов.  **Описание:**  Представьте, что вы работаете логистом в крупной транспортной компании в Казахстане. Вам нужно оптимизировать маршруты доставки товаров между городами. Для этого вам необходимо переставить элементы в массиве, чтобы улучшить порядок выполнения доставок.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о городах, через которые проходит доставка грузов (например, Алматы, Нур-Султан, Караганда). 2. **Пример массива**:   python  Копировать код  cities = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]   1. **Задание**:    * Напишите программу на Python, которая:      + Выведет исходный порядок маршрутов.      + Переставит элементы массива так, чтобы доставка начиналась с Шымкента.      + Выведет новый порядок маршрутов после перестановки. 2. **Пример кода на Python**:   python  Копировать код  cities = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]  # Вывод исходного порядка маршрутов  print("Исходный порядок маршрутов:", cities)  # Перестановка: перенос Шымкента на первое место  cities.remove("Шымкент")  cities.insert(0, "Шымкент")  # Вывод нового порядка маршрутов  print("Новый порядок маршрутов:", cities)   1. **Результат выполнения программы**:   less  Копировать код  Исходный порядок маршрутов: ['Алматы', 'Нур-Султан', 'Караганда', 'Шымкент', 'Атырау']  Новый порядок маршрутов: ['Шымкент', 'Алматы', 'Нур-Султан', 'Караганда', 'Атырау']  **Задание 2: Перестановка приоритета грузов**  **Цель:**  Научиться изменять порядок элементов в одномерном массиве для оптимизации доставки товаров по приоритету.  **Описание:**  Теперь ваша задача — изменить порядок доставки грузов по их приоритету. Например, товары с высоким приоритетом (например, скоропортящиеся продукты) должны доставляться первыми.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о грузах и их приоритете (например, 1 — высокий приоритет, 3 — низкий). 2. **Пример массива**:   python  Копировать код  cargos = ["Мясо", "Одежда", "Техника", "Фрукты", "Химия"]  priorities = [1, 3, 2, 1, 3] # 1 — высокий приоритет, 3 — низкий приоритет   1. **Задание**:    * Напишите программу, которая:      + Переставит грузы в порядке их приоритета (грузы с приоритетом 1 должны идти первыми).      + Выведет список грузов в порядке приоритета. 2. **Пример кода на Python**:   python  Копировать код  cargos = ["Мясо", "Одежда", "Техника", "Фрукты", "Химия"]  priorities = [1, 3, 2, 1, 3]  # Объединение данных о грузах и приоритетах в список кортежей  cargo\_priority = list(zip(priorities, cargos))  # Сортировка по приоритету  cargo\_priority.sort()  # Вывод грузов по приоритету  sorted\_cargos = [cargo for priority, cargo in cargo\_priority]  print("Грузы в порядке приоритета:", sorted\_cargos)   1. **Результат выполнения программы**:   less  Копировать код  Грузы в порядке приоритета: ['Мясо', 'Фрукты', 'Техника', 'Одежда', 'Химия']  **Задание 3: Перестановка маршрутов по расстоянию**  **Цель:**  Научиться изменять порядок элементов массива в зависимости от их характеристик (например, расстояние между городами).  **Описание:**  Представьте, что ваша задача — оптимизировать доставку по расстоянию между городами. Чем ближе города друг к другу, тем раньше они должны быть в маршруте.  **Этапы выполнения:**   1. **Создание массива**:    * Введите данные о городах и расстояниях между ними (например, в километрах). 2. **Пример массива**:   python  Копировать код  cities = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]  distances = [1200, 1000, 800, 1500, 1800] # Расстояние в км от начальной точки   1. **Задание**:    * Напишите программу, которая:      + Переставит города в порядке увеличения расстояния от начальной точки.      + Выведет города в оптимальном порядке для доставки. 2. **Пример кода на Python**:   python  Копировать код  cities = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]  distances = [1200, 1000, 800, 1500, 1800]  # Объединение данных о городах и расстояниях в список кортежей  city\_distance = list(zip(distances, cities))  # Сортировка по расстоянию  city\_distance.sort()  # Вывод городов по возрастанию расстояния  sorted\_cities = [city for distance, city in city\_distance]  print("Города в порядке расстояния для доставки:", sorted\_cities)   1. **Результат выполнения программы**:   less  Копировать код  Города в порядке расстояния для доставки: ['Караганда', 'Нур-Султан', 'Алматы', 'Шымкент', 'Атырау']  **Оценка:**   * Умение работать с одномерными массивами и переставлять элементы в Python. * Способность применять методы сортировки и перестановки для решения реальных задач, связанных с логистикой. * Навыки создания программ, оптимизирующих маршруты доставки и управление приоритетами грузов.   **Результат:**  Учащиеся научатся создавать программы на Python, используя одномерные массивы для перестановки элементов, что является важным навыком для логистов и других специалистов, работающих с оптимизацией процессов. Эти навыки могут быть применимы в управлении маршрутами доставки, приоритетами задач и другими аспектами работы. | | |
|  | Двумерный массив | 9.3.3.2 создавать программы на языке программирования Python (пайтон) с использованием двумерных массивов |
| **Профессия экономист.**  **Цель урока:**   * **9.3.3.2** Создавать программы на языке программирования Python с использованием двумерных массивов.   **Описание:**  Экономист в Казахстане может использовать двумерные массивы для анализа различных финансовых данных: таблицы доходов и расходов, данные о ВВП по регионам или секторам экономики, сравнение бюджетов. В этом задании учащиеся научатся создавать программы на Python с использованием двумерных массивов для решения экономических задач.  **Задание 1: Анализ бюджета регионов Казахстана**  **Цель:**  Научиться работать с двумерными массивами в Python для хранения и анализа финансовых данных.  **Описание:**  Представьте, что вы экономист, который анализирует бюджет регионов Казахстана. У вас есть данные о доходах и расходах регионов за определённый год, и ваша задача — вычислить, какие регионы имеют бюджетный профицит (доходы больше расходов), а какие — дефицит (расходы превышают доходы).  **Этапы выполнения:**   1. **Создание двумерного массива**:    * Введите данные о доходах и расходах пяти регионов Казахстана в двумерный массив. 2. **Пример двумерного массива**:    * Каждая строка массива содержит доходы и расходы одного региона:   python  Копировать код  regions = [  [5000, 4500], # Алматы: доходы 5000 млн тг, расходы 4500 млн тг  [6000, 7000], # Нур-Султан: доходы 6000 млн тг, расходы 7000 млн тг  [4500, 4000], # Караганда: доходы 4500 млн тг, расходы 4000 млн тг  [3000, 3200], # Шымкент: доходы 3000 млн тг, расходы 3200 млн тг  [7000, 6500] # Атырау: доходы 7000 млн тг, расходы 6500 млн тг  ]   1. **Задание**:    * Напишите программу на Python, которая:      + Проходит по каждому региону и определяет, есть ли у региона профицит или дефицит бюджета.      + Выводит сообщение о том, у какого региона профицит, а у какого дефицит. 2. **Пример кода на Python**:   python  Копировать код  regions = [  [5000, 4500], # Алматы  [6000, 7000], # Нур-Султан  [4500, 4000], # Караганда  [3000, 3200], # Шымкент  [7000, 6500] # Атырау  ]  region\_names = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]  for i in range(len(regions)):  income = regions[i][0] # Доходы  expense = regions[i][1] # Расходы  if income > expense:  print(f"{region\_names[i]} имеет профицит бюджета.")  else:  print(f"{region\_names[i]} имеет дефицит бюджета.")   1. **Результат выполнения программы**:   Копировать код  Алматы имеет профицит бюджета.  Нур-Султан имеет дефицит бюджета.  Караганда имеет профицит бюджета.  Шымкент имеет дефицит бюджета.  Атырау имеет профицит бюджета.  **Задание 2: Сравнение экономических показателей регионов**  **Цель:**  Научиться использовать двумерные массивы для хранения и анализа нескольких экономических показателей.  **Описание:**  Теперь вам нужно сравнить экономические показатели разных регионов Казахстана, включая данные о ВВП, уровне безработицы и инвестициях. Ваша задача — рассчитать средние показатели по каждому региону.  **Этапы выполнения:**   1. **Создание двумерного массива**:    * Введите данные о ВВП (в миллиардах тенге), уровне безработицы (в процентах) и инвестициях (в миллиардах тенге) для нескольких регионов в двумерный массив. 2. **Пример двумерного массива**:   regions\_data = [  [12, 5.2, 1.5], # Алматы: ВВП 12 млрд тг, безработица 5.2%, инвестиции 1.5 млрд тг  [10, 6.0, 2.0], # Нур-Султан: ВВП 10 млрд тг, безработица 6.0%, инвестиции 2.0 млрд тг  [8, 7.1, 1.8], # Караганда: ВВП 8 млрд тг, безработица 7.1%, инвестиции 1.8 млрд тг  [6, 8.0, 1.3], # Шымкент: ВВП 6 млрд тг, безработица 8.0%, инвестиции 1.3 млрд тг  [14, 4.5, 2.5] # Атырау: ВВП 14 млрд тг, безработица 4.5%, инвестиции 2.5 млрд тг  ]   1. **Задание**:    * Напишите программу, которая:      + Рассчитает среднее значение ВВП, безработицы и инвестиций для всех регионов.      + Выведет результаты на экран. 2. **Пример кода на Python**:   python  Копировать код  regions\_data = [  [12, 5.2, 1.5], # Алматы  [10, 6.0, 2.0], # Нур-Султан  [8, 7.1, 1.8], # Караганда  [6, 8.0, 1.3], # Шымкент  [14, 4.5, 2.5] # Атырау  ]  total\_vvp = 0  total\_unemployment = 0  total\_investment = 0  for region in regions\_data:  total\_vvp += region[0]  total\_unemployment += region[1]  total\_investment += region[2]  avg\_vvp = total\_vvp / len(regions\_data)  avg\_unemployment = total\_unemployment / len(regions\_data)  avg\_investment = total\_investment / len(regions\_data)  print(f"Средний ВВП: {avg\_vvp} млрд тг")  print(f"Средний уровень безработицы: {avg\_unemployment}%")  print(f"Средний объем инвестиций: {avg\_investment} млрд тг")   1. **Результат выполнения программы**:   Копировать код  Средний ВВП: 10.0 млрд тг  Средний уровень безработицы: 6.16%  Средний объем инвестиций: 1.82 млрд тг  **Задание 3: Анализ прибыли компании по регионам**  **Цель:**  Научиться использовать двумерные массивы для хранения и анализа данных о прибыли компании в разных регионах.  **Описание:**  Представьте, что вы анализируете прибыль крупной компании в различных регионах Казахстана за последние три квартала. Ваша задача — рассчитать общую прибыль компании по каждому региону.  **Этапы выполнения:**   1. **Создание двумерного массива**:    * Введите данные о прибыли компании (в миллионах тенге) в каждом регионе за три квартала. 2. **Пример двумерного массива**:   profit = [  [500, 600, 550], # Алматы: прибыль за 1, 2, 3 квартал  [400, 450, 500], # Нур-Султан  [300, 350, 320], # Караганда  [200, 250, 280], # Шымкент  [700, 750, 800] # Атырау  ]   1. **Задание**:    * Напишите программу, которая:      + Рассчитает общую прибыль за три квартала по каждому региону.      + Выведет результаты на экран. 2. **Пример кода на Python**:   profit = [  [500, 600, 550], # Алматы  [400, 450, 500], # Нур-Султан  [300, 350, 320], # Караганда  [200, 250, 280], # Шымкент  [700, 750, 800] # Атырау  ]  region\_names = ["Алматы", "Нур-Султан", "Караганда", "Шымкент", "Атырау"]  for i in range(len(profit)):  total\_profit = sum(profit[i])  print(f"Общая прибыль в регионе {region\_names[i]} за три квартала: {total\_profit} млн тг")   1. **Результат выполнения программы**:   yaml  Копировать код  Общая прибыль в регионе Алматы за три квартала: 1650 млн тг  Общая прибыль в регионе Нур-Султан за три квартала: 1350 млн тг  Общая прибыль в регионе Караганда за три квартала: 970 млн тг  Общая прибыль в регионе Шымкент за три квартала: 730 млн тг  Общая прибыль в регионе Атырау за три квартала: 2250 млн тг  **Оценка:**   * Умение работать с двумерными массивами на языке программирования Python. * Способность применять двумерные массивы для хранения и анализа данных. * Навыки создания программ для решения задач, связанных с экономикой и анализом данных в различных сферах.   **Результат:**  Учащиеся научатся создавать программы на Python с использованием двумерных массивов для решения задач, связанных с анализом экономических данных и финансовой информации. Эти навыки могут быть полезны для профессий, требующих обработки и анализа данных, таких как экономист, бухгалтер или аналитик. | | |
|  | Сортировка | 9.3.2.1 применять алгоритмы сортировки; 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов; 9.3.3.2 создавать программы на языке программирования Python (пайтон)с использованием двумерных массивов |
| **Профессии аналитика данных в области сельского хозяйства.**  **Цели урока:**   * **9.3.2.1** Применять алгоритмы сортировки. * **9.3.3.1** Создавать программы на языке программирования Python с использованием одномерных массивов. * **9.3.3.2** Создавать программы на языке программирования Python с использованием двумерных массивов.   **Описание:**  Аналитик данных в сфере сельского хозяйства анализирует информацию о производстве, урожайности и расходах на сельскохозяйственную продукцию в разных регионах Казахстана. Он может использовать алгоритмы сортировки для упорядочивания данных, что помогает принимать более обоснованные решения по распределению ресурсов и прогнозированию урожая.  В этом задании учащиеся будут учиться создавать программы на Python, используя одномерные и двумерные массивы, а также применять алгоритмы сортировки для решения реальных задач в области сельского хозяйства.  **Задание 1: Сортировка урожайности по регионам (одномерный массив)**  **Цель:**  Научиться применять алгоритмы сортировки к одномерным массивам для анализа данных об урожайности по регионам Казахстана.  **Описание:**  Представьте, что вы аналитик данных в области сельского хозяйства. У вас есть данные об урожайности (в тоннах) по регионам Казахстана за последний год. Ваша задача — отсортировать данные по урожайности, чтобы определить регионы с наибольшей и наименьшей производительностью.  **Этапы выполнения:**   1. **Создание одномерного массива**:    * Введите данные об урожайности (в тоннах) в одномерный массив. Каждый элемент массива соответствует региону Казахстана. 2. **Пример массива**:   yields = [3200, 4500, 2800, 5000, 3900] # Урожайность в тоннах  regions = ["Алматы", "Караганда", "Шымкент", "Нур-Султан", "Атырау"]   1. **Задание**:    * Напишите программу на Python, которая:      + Применит алгоритм сортировки к массиву урожайности.      + Отсортирует данные по возрастанию урожайности и выведет результаты. 2. **Пример кода на Python**:   yields = [3200, 4500, 2800, 5000, 3900]  regions = ["Алматы", "Караганда", "Шымкент", "Нур-Султан", "Атырау"]  # Объединение данных о регионах и урожайности  data = list(zip(yields, regions))  # Сортировка данных по урожайности  data.sort()  # Вывод отсортированных данных  for yield\_tonnes, region in data:  print(f"Регион: {region}, Урожайность: {yield\_tonnes} тонн")   1. **Результат выполнения программы**:   Регион: Шымкент, Урожайность: 2800 тонн  Регион: Алматы, Урожайность: 3200 тонн  Регион: Атырау, Урожайность: 3900 тонн  Регион: Караганда, Урожайность: 4500 тонн  Регион: Нур-Султан, Урожайность: 5000 тонн  **Задание 2: Сортировка расходов на сельскохозяйственные проекты (двумерный массив)**  **Цель:**  Научиться применять алгоритмы сортировки к двумерным массивам для анализа расходов на сельскохозяйственные проекты.  **Описание:**  Теперь вам нужно проанализировать расходы на сельскохозяйственные проекты в разных регионах Казахстана. Каждый проект имеет определённую стоимость. Ваша задача — отсортировать данные о проектах по стоимости, чтобы определить самые дорогие и самые дешевые проекты.  **Этапы выполнения:**   1. **Создание двумерного массива**:    * Введите данные о сельскохозяйственных проектах в двумерный массив. Каждая строка содержит название проекта, регион и стоимость проекта (в миллионах тенге). 2. **Пример массива**:   projects = [  ["Проект А", "Алматы", 500],  ["Проект B", "Нур-Султан", 300],  ["Проект C", "Караганда", 700],  ["Проект D", "Шымкент", 200],  ["Проект E", "Атырау", 400]  ]   1. **Задание**:    * Напишите программу, которая:      + Применит алгоритм сортировки к двумерному массиву, сортируя данные по стоимости проекта.      + Выведет отсортированные данные с указанием проекта, региона и стоимости. 2. **Пример кода на Python**:   projects = [  ["Проект А", "Алматы", 500],  ["Проект B", "Нур-Султан", 300],  ["Проект C", "Караганда", 700],  ["Проект D", "Шымкент", 200],  ["Проект E", "Атырау", 400]  ]  # Сортировка проектов по стоимости  projects.sort(key=lambda x: x[2])  # Вывод отсортированных данных  for project in projects:  print(f"Проект: {project[0]}, Регион: {project[1]}, Стоимость: {project[2]} млн тг")   1. **Результат выполнения программы**:   Проект: Проект D, Регион: Шымкент, Стоимость: 200 млн тг  Проект: Проект B, Регион: Нур-Султан, Стоимость: 300 млн тг  Проект: Проект E, Регион: Атырау, Стоимость: 400 млн тг  Проект: Проект А, Регион: Алматы, Стоимость: 500 млн тг  Проект: Проект C, Регион: Караганда, Стоимость: 700 млн тг  **Задание 3: Применение различных алгоритмов сортировки**  **Цель:**  Научиться применять различные алгоритмы сортировки для решения реальных задач.  **Описание:**  Теперь ваша задача — применить несколько алгоритмов сортировки для анализа данных. Вы будете использовать **сортировку пузырьком** и **быструю сортировку** для обработки данных об урожайности и стоимости проектов.  **Этапы выполнения:**   1. **Пример данных**:    * Используйте данные об урожайности или стоимости проектов из предыдущих заданий. 2. **Задание**:    * Реализуйте сортировку пузырьком для одномерного массива.    * Примените встроенную функцию Python для быстрой сортировки двумерного массива. 3. **Пример кода на Python для сортировки пузырьком**:   yields = [3200, 4500, 2800, 5000, 3900]  # Сортировка пузырьком  for i in range(len(yields)):  for j in range(0, len(yields) - i - 1):  if yields[j] > yields[j + 1]:  yields[j], yields[j + 1] = yields[j + 1], yields[j]  # Вывод отсортированных данных  print("Отсортированная урожайность:", yields)   1. **Результат выполнения программы**:   Отсортированная урожайность: [2800, 3200, 3900, 4500, 5000]  **Оценка:**   * Умение применять различные алгоритмы сортировки к одномерным и двумерным массивам. * Способность использовать сортировку для анализа данных, связанных с экономикой и сельским хозяйством. * Навыки разработки программ на Python для решения задач сортировки.   **Результат:**  Учащиеся научатся применять алгоритмы сортировки для анализа данных, используя одномерные и двумерные массивы на языке Python. Эти навыки могут быть полезны для профессий, связанных с анализом данных, такими как экономист, аналитик или специалист по сельскому хозяйству. | | |
|  | Удаление и вставка элемента | 9.3.3.1 создавать программы на языке программирования Python (пайтон) с использованием одномерных массивов; 9.3.3.2 создавать программы на языке программирования Python (пайтон) с использованием двумерных массивов |
| **Профессия менеджер по управлению складом.**  **Цели урока:**   * **9.3.3.1** Создавать программы на языке программирования Python с использованием одномерных массивов. * **9.3.3.2** Создавать программы на языке программирования Python с использованием двумерных массивов.   **Описание:**  Менеджер по управлению складом в Казахстане часто работает с массивами данных о запасах товаров. В процессе работы ему необходимо добавлять новые товары на склад, удалять или перемещать товары, а также обновлять данные. В этом задании учащиеся научатся работать с удалением и вставкой элементов в одномерные и двумерные массивы, применяя эти навыки к задачам управления складом.  **Задание 1: Удаление и вставка товара в списке (одномерный массив)**  **Цель:**  Научиться работать с одномерными массивами, выполняя операции вставки и удаления элементов для моделирования управления запасами на складе.  **Описание:**  Представьте, что вы менеджер по управлению складом. У вас есть список товаров, хранящихся на складе, и вам нужно добавить новый товар, а затем удалить один из существующих товаров.  **Этапы выполнения:**   1. **Создание одномерного массива**:    * Введите данные о товарах на складе в одномерный массив (например, продукты питания или строительные материалы). 2. **Пример массива**:   products = ["Сахар", "Мука", "Рис", "Чай", "Масло"]   1. **Задание**:    * Напишите программу на Python, которая:      + Вставляет новый товар в определённую позицию в массиве.      + Удаляет товар с указанной позиции. 2. **Пример кода на Python**:   products = ["Сахар", "Мука", "Рис", "Чай", "Масло"]  # Вставка нового товара  new\_product = "Гречка"  position = 2 # Вставляем на 3-ю позицию (индексация начинается с 0)  products.insert(position, new\_product)  # Вывод списка товаров после вставки  print("Список товаров после вставки:", products)  # Удаление товара  remove\_position = 1 # Удаляем товар с 2-й позиции  removed\_product = products.pop(remove\_position)  # Вывод списка товаров после удаления  print("Список товаров после удаления:", products)  print(f"Удалённый товар: {removed\_product}")   1. **Результат выполнения программы**:   Список товаров после вставки: ['Сахар', 'Мука', 'Гречка', 'Рис', 'Чай', 'Масло']  Список товаров после удаления: ['Сахар', 'Гречка', 'Рис', 'Чай', 'Масло']  Удалённый товар: Мука  **Задание 2: Обновление списка товаров на складе (двумерный массив)**  **Цель:**  Научиться работать с двумерными массивами для вставки и удаления элементов, используя пример управления данными о товарах и их количестве на складе.  **Описание:**  Теперь вы должны управлять складскими запасами и вести учет товаров и их количества. Вам нужно обновить двумерный массив, добавив новый товар с его количеством и удалив один из товаров, который больше не нужен.  **Этапы выполнения:**   1. **Создание двумерного массива**:    * Введите данные о товарах и их количестве в двумерный массив, где каждая строка представляет товар и его количество. 2. **Пример массива**:   stock = [  ["Сахар", 100],  ["Мука", 50],  ["Рис", 80],  ["Чай", 40],  ["Масло", 60]  ]   1. **Задание**:    * Напишите программу на Python, которая:      + Вставляет новый товар с его количеством в список.      + Удаляет товар с указанной позиции. 2. **Пример кода на Python**:   stock = [  ["Сахар", 100],  ["Мука", 50],  ["Рис", 80],  ["Чай", 40],  ["Масло", 60]  ]  # Вставка нового товара с его количеством  new\_product = ["Гречка", 70]  position = 2 # Вставляем на 3-ю позицию  stock.insert(position, new\_product)  # Вывод списка товаров после вставки  print("Склад после вставки товара:")  for item in stock:  print(item)  # Удаление товара  remove\_position = 1 # Удаляем товар с 2-й позиции  removed\_product = stock.pop(remove\_position)  # Вывод списка товаров после удаления  print("\nСклад после удаления товара:")  for item in stock:  print(item)  print(f"\nУдалённый товар: {removed\_product}")   1. **Результат выполнения программы**:   Склад после вставки товара:  ['Сахар', 100]  ['Мука', 50]  ['Гречка', 70]  ['Рис', 80]  ['Чай', 40]  ['Масло', 60]  Склад после удаления товара:  ['Сахар', 100]  ['Гречка', 70]  ['Рис', 80]  ['Чай', 40]  ['Масло', 60]  Удалённый товар: ['Мука', 50]  **Задание 3: Управление запасами товаров на складе**  **Цель:**  Научиться использовать одномерные и двумерные массивы для управления запасами товаров на складе.  **Описание:**  Представьте, что вам нужно не только вставлять и удалять товары, но и обновлять их количество на складе. Вы будете работать с массивами данных и выполнять операции по добавлению, удалению и обновлению товаров и их количества.  **Этапы выполнения:**   1. **Пример данных**:    * Используйте массив из предыдущих заданий. 2. **Задание**:    * Напишите программу, которая:      + Добавляет новый товар и его количество.      + Удаляет один из товаров.      + Обновляет количество для существующего товара. 3. **Пример кода на Python**:   stock = [  ["Сахар", 100],  ["Мука", 50],  ["Рис", 80],  ["Чай", 40],  ["Масло", 60]  ]  # Вставка нового товара  new\_product = ["Гречка", 70]  stock.append(new\_product)  # Удаление товара  removed\_product = stock.pop(1) # Удаляем "Мука"  # Обновление количества товара "Рис"  for item in stock:  if item[0] == "Рис":  item[1] = 100 # Обновляем количество до 100  # Вывод результатов  print("Склад после всех операций:")  for item in stock:  print(item)  print(f"\nУдалённый товар: {removed\_product}")   1. **Результат выполнения программы**:   Склад после всех операций:  ['Сахар', 100]  ['Рис', 100]  ['Чай', 40]  ['Масло', 60]  ['Гречка', 70]  Удалённый товар: ['Мука', 50]  **Оценка:**   * Умение вставлять и удалять элементы в одномерных и двумерных массивах на Python. * Способность управлять данными о товарах и их количестве. * Навыки разработки программ для решения задач, связанных с управлением запасами на складе.   **Результат:**  Учащиеся научатся создавать программы на Python, используя одномерные и двумерные массивы для управления данными, связанными с добавлением и удалением товаров. Эти навыки применимы к профессиям, требующим работы с данными, такими как менеджер по управлению складом, логист или аналитик. | | |
| **Раздел 5. Создание 2D игры на языке** | | |
|  | Библиотека PyGame (пайгейм) | 9.3.3.3 подключать библитотеку PyGame (пайгейм); 9.3.3.4 использовать готовые модули библиотеки PyGame (пайгейм) для создания окна для игры |
| **Профессия разработчик видеоигр.**  **Цели урока:**   * **9.3.3.3** Подключать библиотеку PyGame. * **9.3.3.4** Использовать готовые модули библиотеки PyGame для создания окна для игры.   **Описание:**  В Казахстане сфера разработки видеоигр постепенно набирает популярность, и такие навыки, как работа с библиотекой PyGame, могут стать основой для создания игр. Разработчики видеоигр используют различные библиотеки для создания визуальных интерфейсов, а также для добавления анимаций и взаимодействия с пользователем.  В этом задании учащиеся научатся подключать библиотеку PyGame и использовать её для создания простого окна для игры, имитируя задачи разработчиков игр.  **Задание 1: Подключение библиотеки PyGame**  **Цель:**  Научиться подключать библиотеку PyGame в программе на Python.  **Описание:**  Представьте, что вы начинающий разработчик игр, который собирается создать свою первую игру. Первым шагом будет подключение библиотеки PyGame и создание окна для игры.  **Этапы выполнения:**   1. **Подключение PyGame**:    * Убедитесь, что библиотека PyGame установлена. Для этого в командной строке выполните команду:   pip install pygame   1. **Подключение библиотеки в коде**:    * Напишите программу, которая подключает библиотеку PyGame и инициализирует её. 2. **Пример кода на Python**:   import pygame  # Инициализация библиотеки PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((600, 400))  pygame.display.set\_caption("Моя первая игра")  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * При запуске программы откроется окно размером 600x400 пикселей с заголовком "Моя первая игра". Окно можно закрыть, нажав на крестик.   **Задание 2: Создание окна для игры**  **Цель:**  Научиться использовать модули библиотеки PyGame для создания окна и графики в программе.  **Описание:**  Теперь ваша задача — добавить в окно простую графику и элементы интерфейса, такие как цвет фона и примитивные фигуры (квадраты, круги и линии). Это основа для создания интерактивных игр.  **Этапы выполнения:**   1. **Создание окна**:    * Используйте PyGame для создания окна с фоном и простыми фигурами. 2. **Добавление графики**:    * Нарисуйте на экране квадрат, круг или линию. 3. **Пример кода на Python**:   import pygame  # Инициализация библиотеки PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((600, 400))  pygame.display.set\_caption("Моя первая игра")  # Цвета (R, G, B)  white = (255, 255, 255)  black = (0, 0, 0)  red = (255, 0, 0)  blue = (0, 0, 255)  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Заливка фона белым цветом  window.fill(white)  # Рисование квадрата (координаты и размеры)  pygame.draw.rect(window, red, (200, 150, 100, 50))  # Рисование круга (координаты центра и радиус)  pygame.draw.circle(window, blue, (300, 200), 40)  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * На экране появится окно с белым фоном, красным квадратом и синим кругом.   **Задание 3: Добавление движущихся объектов**  **Цель:**  Научиться создавать анимацию объектов в окне игры с использованием PyGame.  **Описание:**  Теперь вы должны добавить интерактивность в вашу игру, заставив объекты двигаться. Например, вы можете заставить квадрат двигаться вправо, а круг — вниз, что создаст простую анимацию.  **Этапы выполнения:**   1. **Движение объектов**:    * Добавьте логику, которая изменяет координаты квадрата и круга в зависимости от времени. 2. **Пример кода на Python**:   import pygame  # Инициализация библиотеки PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((600, 400))  pygame.display.set\_caption("Анимация объектов")  # Цвета (R, G, B)  white = (255, 255, 255)  red = (255, 0, 0)  blue = (0, 0, 255)  # Координаты объектов  square\_x = 200  square\_y = 150  circle\_x = 300  circle\_y = 200  # Скорость движения объектов  square\_speed = 3  circle\_speed = 2  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Изменение координат объектов  square\_x += square\_speed  circle\_y += circle\_speed  # Заливка фона белым цветом  window.fill(white)  # Рисование движущегося квадрата  pygame.draw.rect(window, red, (square\_x, square\_y, 100, 50))  # Рисование движущегося круга  pygame.draw.circle(window, blue, (circle\_x, circle\_y), 40)  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * В окне игры вы увидите движущийся квадрат, который перемещается вправо, и круг, который движется вниз.   **Оценка:**   * Умение подключать библиотеку PyGame в Python. * Способность использовать модули библиотеки PyGame для создания графического окна. * Навыки разработки простых игр с использованием анимации объектов.   **Результат:**  Учащиеся научатся работать с библиотекой PyGame для создания игр. Они смогут создавать графическое окно, рисовать объекты и управлять их движением, что является важным навыком для начинающих разработчиков игр. Эти навыки могут быть полезны в профессии разработчика видеоигр и при создании интерактивных приложений. | | |
|  | Задний фон и персонажи игры | 9.3.3.5 создавать задний фон игры  9.3.3.6 загружать готовые персонажи для игры |
| **Профессия разработчика видеоигр**  **Цели урока:**   * **9.3.3.5** Создавать задний фон игры. * **9.3.3.6** Загружать готовые персонажи для игры.   **Описание:**  В Казахстане игровая индустрия развивается, и профессия **разработчика видеоигр** становится всё более востребованной. Разработчики игр создают интерактивные среды, которые могут включать элементы казахстанской культуры, природы и истории. В этом задании учащиеся научатся создавать задний фон для игры и загружать персонажей, используя библиотеку PyGame, что является одним из ключевых навыков для разработки игр.  **Задание 1: Создание заднего фона для игры**  **Цель:**  Научиться создавать задний фон для игры с использованием библиотеки PyGame.  **Описание:**  Представьте, что вы разрабатываете игру, действие которой происходит на фоне казахстанского пейзажа (например, горы Тянь-Шаня или степи). Ваша задача — создать задний фон для этой игры, используя готовое изображение.  **Этапы выполнения:**   1. **Подготовка изображения**:    * Найдите или создайте изображение с пейзажем Казахстана, например, горы Тянь-Шаня, степи или известные места Казахстана. Изображение должно быть в формате .jpg или .png. 2. **Загрузка изображения фона**:    * Загрузите изображение в PyGame, чтобы оно отображалось в окне игры. 3. **Пример кода на Python для создания фона**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Игра с фоном")  # Загрузка изображения заднего фона  background = pygame.image.load("background\_kazakhstan.jpg")  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Отображение фона  window.blit(background, (0, 0))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * В окне игры будет отображаться выбранное изображение заднего фона с пейзажем Казахстана.   **Задание 2: Загрузка и отображение персонажа**  **Цель:**  Научиться загружать и отображать готового персонажа для игры с использованием PyGame.  **Описание:**  Теперь вам нужно добавить персонажа в вашу игру. Представьте, что это казахский батыр или персонаж национальной культуры, который будет двигаться по игровому полю. Вы загрузите изображение персонажа и разместите его в окне игры.  **Этапы выполнения:**   1. **Подготовка изображения персонажа**:    * Найдите или создайте изображение персонажа, например, казахского батыра или традиционного героя. Формат изображения должен быть .png с прозрачным фоном. 2. **Загрузка изображения персонажа**:    * Загрузите изображение персонажа и отобразите его на фоне игры. 3. **Пример кода на Python для добавления персонажа**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Игра с персонажем")  # Загрузка изображений заднего фона и персонажа  background = pygame.image.load("background\_kazakhstan.jpg")  character = pygame.image.load("batyr.png")  # Координаты персонажа  character\_x = 100  character\_y = 300  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Отображение фона  window.blit(background, (0, 0))  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * В окне игры будет отображаться задний фон и персонаж, например, казахский батыр, размещенный в выбранной вами точке экрана.   **Задание 3: Перемещение персонажа по игровому полю**  **Цель:**  Научиться добавлять взаимодействие с персонажем в игре, позволяя ему перемещаться по полю.  **Описание:**  Теперь ваша задача — добавить возможность перемещения персонажа по полю, используя клавиши стрелок на клавиатуре. Это важная часть игровой механики, которая позволяет игроку управлять персонажем.  **Этапы выполнения:**   1. **Добавление логики для перемещения**:    * Напишите код, который позволит перемещать персонажа по экрану с помощью клавиш стрелок. 2. **Пример кода на Python для перемещения персонажа**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Игра с движущимся персонажем")  # Загрузка изображений заднего фона и персонажа  background = pygame.image.load("background\_kazakhstan.jpg")  character = pygame.image.load("batyr.png")  # Координаты и скорость персонажа  character\_x = 100  character\_y = 300  speed = 5  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа в зависимости от нажатых клавиш  if keys[pygame.K\_LEFT]:  character\_x -= speed  if keys[pygame.K\_RIGHT]:  character\_x += speed  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Отображение фона  window.blit(background, (0, 0))  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Персонаж (например, казахский батыр) будет двигаться по игровому полю, когда вы нажимаете клавиши стрелок на клавиатуре.   **Оценка:**   * Умение создавать и использовать задний фон в играх. * Способность загружать и отображать персонажей. * Навыки добавления интерактивности, позволяя игроку управлять персонажем.   **Результат:**  Учащиеся научатся создавать игры с использованием PyGame, создавая задний фон и добавляя готовых персонажей. Эти навыки могут быть полезны для профессии разработчика видеоигр, особенно при создании игр с культурными элементами Казахстана. | | |
|  | Анимирование персонажей | 9.3.3.7 программировать движение персонажа  9.3.3.8 управлять персонажами с клавиатуры |
| **Профессия разработчик видеоигр.**  **Цели урока:**   * **9.3.3.7** Программировать движение персонажа. * **9.3.3.8** Управлять персонажами с клавиатуры.   **Описание:**  В Казахстане профессия **разработчика видеоигр** становится всё более востребованной благодаря росту интереса к игровой индустрии и развитию технологий. Одним из ключевых навыков разработчика является умение анимировать персонажей и управлять их движениями с помощью клавиатуры. В этом задании учащиеся научатся программировать движение персонажа и добавлять анимацию с использованием библиотеки PyGame, создавая основу для динамических игр.  **Задание 1: Программирование движения персонажа**  **Цель:**  Научиться программировать движение персонажа с использованием библиотеки PyGame.  **Описание:**  Представьте, что вы разрабатываете игру, в которой игрок управляет казахским батыром (героем) на игровом поле. Вам нужно запрограммировать движение персонажа так, чтобы он мог перемещаться по экрану при нажатии клавиш стрелок.  **Этапы выполнения:**   1. **Создание окна и загрузка персонажа**:    * Создайте окно игры с фоном и персонажем (например, изображением казахского батыра).    * Персонаж должен двигаться при нажатии клавиш стрелок. 2. **Пример кода на Python для движения персонажа**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Движение персонажа")  # Загрузка изображения персонажа  character = pygame.image.load("batyr.png")  # Координаты и скорость персонажа  character\_x = 100  character\_y = 300  speed = 5  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа в зависимости от нажатых клавиш  if keys[pygame.K\_LEFT]:  character\_x -= speed  if keys[pygame.K\_RIGHT]:  character\_x += speed  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Отображение фона (заполнение черным цветом)  window.fill((0, 0, 0))  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Персонаж будет двигаться влево, вправо, вверх и вниз в зависимости от нажатых клавиш.   **Задание 2: Анимация персонажа при движении**  **Цель:**  Научиться добавлять анимацию персонажа при его движении с помощью PyGame.  **Описание:**  Теперь вам нужно добавить анимацию, чтобы персонаж изменял положение или внешний вид при движении. Например, когда батыр движется вправо, он должен повернуться вправо, а при движении влево — повернуться влево.  **Этапы выполнения:**   1. **Загрузка анимаций**:    * Создайте или найдите несколько изображений для персонажа (например, для движения влево и вправо).    * Загрузите эти изображения в PyGame и отображайте правильное изображение в зависимости от направления движения. 2. **Пример кода на Python для анимации персонажа**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Анимация персонажа")  # Загрузка изображений персонажа для разных направлений  character\_right = pygame.image.load("batyr\_right.png")  character\_left = pygame.image.load("batyr\_left.png")  character = character\_right # Начальное положение — направо  # Координаты и скорость персонажа  character\_x = 100  character\_y = 300  speed = 5  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа и выбор анимации  if keys[pygame.K\_LEFT]:  character\_x -= speed  character = character\_left # Поворачиваем налево  if keys[pygame.K\_RIGHT]:  character\_x += speed  character = character\_right # Поворачиваем направо  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Отображение фона (черный цвет)  window.fill((0, 0, 0))  # Отображение анимированного персонажа  window.blit(character, (character\_x, character\_y))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Персонаж будет двигаться влево и вправо, меняя своё изображение в зависимости от направления.   **Задание 3: Усложненная анимация — движение персонажа с разными состояниями**  **Цель:**  Научиться добавлять анимации для разных состояний персонажа (например, ходьба, бег).  **Описание:**  Теперь ваша задача — добавить более сложные анимации, например, чтобы персонаж использовал разные изображения для ходьбы и бега в зависимости от скорости его движения.  **Этапы выполнения:**   1. **Загрузка изображений для анимации**:    * Найдите или создайте несколько изображений для персонажа в разных состояниях (например, ходьба и бег). 2. **Пример кода на Python для управления состояниями персонажа**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Анимация персонажа с разными состояниями")  # Загрузка изображений персонажа  walk\_right = pygame.image.load("batyr\_walk\_right.png")  run\_right = pygame.image.load("batyr\_run\_right.png")  walk\_left = pygame.image.load("batyr\_walk\_left.png")  run\_left = pygame.image.load("batyr\_run\_left.png")  character = walk\_right # Начальное изображение  # Координаты и скорость персонажа  character\_x = 100  character\_y = 300  speed = 5  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа и выбор анимации в зависимости от состояния (ходьба или бег)  if keys[pygame.K\_LEFT] and keys[pygame.K\_LSHIFT]: # Бег влево  character\_x -= speed \* 2  character = run\_left  elif keys[pygame.K\_LEFT]:  character\_x -= speed  character = walk\_left  if keys[pygame.K\_RIGHT] and keys[pygame.K\_LSHIFT]: # Бег вправо  character\_x += speed \* 2  character = run\_right  elif keys[pygame.K\_RIGHT]:  character\_x += speed  character = walk\_right  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Отображение фона  window.fill((0, 0, 0))  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Персонаж будет менять своё состояние (ходьба или бег) в зависимости от скорости движения (с нажатой клавишей Shift персонаж будет бежать).   **Оценка:**   * Умение программировать движение персонажа с использованием клавиатуры. * Способность добавлять анимацию для персонажа в зависимости от его действий. * Навыки работы с PyGame для создания интерактивных игр.   **Результат:**  Учащиеся научатся программировать анимацию персонажа и управлять им с клавиатуры, что является важным навыком в разработке игр. Эти знания могут быть применены в профессии разработчика видеоигр, а также при создании интерактивных приложений, связанных с казахстанской культурой или историей. | | |
|  | Программирование условий | 9.3.3.9 разрабатывать игру по готовому сценарию 9.3.3.10 реализовать алгоритм подсчета результатов игры |
| **Профессия разработчика игр**  **Цели урока:**   * **9.3.3.9** Разрабатывать игру по готовому сценарию. * **9.3.3.10** Реализовать алгоритм подсчета результатов игры.   **Описание:**  Разработка игр — это процесс, который требует не только навыков программирования, но и умения реализовывать логику и условия игры. Например, разработчик игр в Казахстане может создавать игры, которые отражают культуру и традиции страны. В этом задании учащиеся научатся программировать условия игры и реализовывать алгоритмы подсчета результатов с использованием библиотеки PyGame.  **Задание 1: Разработка простой игры с условиями**  **Цель:**  Научиться разрабатывать простую игру с использованием программирования условий.  **Описание:**  Представьте, что вы разрабатываете игру, в которой персонаж (например, казахский батыр) собирает монеты или другие ценные предметы. Ваша задача — создать игру, где персонаж перемещается по экрану и собирает предметы, которые исчезают после сбора.  **Этапы выполнения:**   1. **Создание персонажа и предметов**:    * Персонаж должен перемещаться с помощью клавиш стрелок, а предметы должны располагаться на игровом поле.    * Когда персонаж касается предмета, он исчезает, и очки увеличиваются. 2. **Пример кода на Python для создания игры**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Игра: Собери предметы")  # Цвета  black = (0, 0, 0)  white = (255, 255, 255)  red = (255, 0, 0)  # Загрузка персонажа  character = pygame.image.load("batyr.png")  character\_x = 100  character\_y = 300  speed = 5  # Позиция предметов  item\_x = 400  item\_y = 300  item\_collected = False  # Счет игры  score = 0  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа  if keys[pygame.K\_LEFT]:  character\_x -= speed  if keys[pygame.K\_RIGHT]:  character\_x += speed  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Проверка на сбор предмета  if not item\_collected and abs(character\_x - item\_x) < 50 and abs(character\_y - item\_y) < 50:  item\_collected = True  score += 1 # Увеличение счета  # Заливка фона  window.fill(white)  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Отображение предмета, если он не собран  if not item\_collected:  pygame.draw.circle(window, red, (item\_x, item\_y), 20)  # Отображение счета  font = pygame.font.Font(None, 36)  text = font.render(f"Счет: {score}", True, black)  window.blit(text, (10, 10))  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Персонаж будет двигаться по экрану, собирать предметы (например, монеты), и каждый раз, когда предмет будет собран, счет игры увеличится.   **Задание 2: Подсчет результатов игры**  **Цель:**  Научиться реализовывать алгоритм подсчета результатов в игре.  **Описание:**  Теперь вам нужно добавить алгоритм подсчета очков, который будет определять, сколько предметов игрок собрал. Также добавьте условие окончания игры, когда игрок соберет все предметы.  **Этапы выполнения:**   1. **Добавление нескольких предметов**:    * Добавьте несколько предметов на игровое поле и организуйте их сбор персонажем. 2. **Пример кода на Python для подсчета очков**:   import pygame  # Инициализация PyGame  pygame.init()  # Настройка окна игры  window = pygame.display.set\_mode((800, 600))  pygame.display.set\_caption("Игра: Собери все предметы")  # Цвета  black = (0, 0, 0)  white = (255, 255, 255)  red = (255, 0, 0)  # Загрузка персонажа  character = pygame.image.load("batyr.png")  character\_x = 100  character\_y = 300  speed = 5  # Позиция нескольких предметов  items = [(400, 300), (600, 200), (200, 150)]  items\_collected = [False, False, False]  # Счет игры  score = 0  # Основной цикл игры  running = True  while running:  for event in pygame.event.get():  if event.type == pygame.QUIT:  running = False  # Получение состояния клавиш  keys = pygame.key.get\_pressed()  # Движение персонажа  if keys[pygame.K\_LEFT]:  character\_x -= speed  if keys[pygame.K\_RIGHT]:  character\_x += speed  if keys[pygame.K\_UP]:  character\_y -= speed  if keys[pygame.K\_DOWN]:  character\_y += speed  # Проверка на сбор предметов  for i in range(len(items)):  if not items\_collected[i]:  item\_x, item\_y = items[i]  if abs(character\_x - item\_x) < 50 and abs(character\_y - item\_y) < 50:  items\_collected[i] = True  score += 1 # Увеличение счета  # Заливка фона  window.fill(white)  # Отображение персонажа  window.blit(character, (character\_x, character\_y))  # Отображение предметов, если они не собраны  for i in range(len(items)):  if not items\_collected[i]:  item\_x, item\_y = items[i]  pygame.draw.circle(window, red, (item\_x, item\_y), 20)  # Отображение счета  font = pygame.font.Font(None, 36)  text = font.render(f"Счет: {score}", True, black)  window.blit(text, (10, 10))  # Проверка условия окончания игры  if score == len(items):  text = font.render("Все предметы собраны! Игра окончена.", True, black)  window.blit(text, (200, 300))  pygame.display.update()  pygame.time.wait(2000)  running = False  # Обновление экрана  pygame.display.update()  # Завершение работы PyGame  pygame.quit()   1. **Результат выполнения программы**:    * Игра будет продолжаться до тех пор, пока игрок не соберет все предметы на поле. После сбора всех предметов на экране отобразится сообщение "Игра окончена".   **Оценка:**   * Умение разрабатывать игру по готовому сценарию. * Способность реализовывать алгоритм подсчета результатов в игре. * Навыки программирования условий и взаимодействия персонажа с объектами.   **Результат:**  Учащиеся научатся программировать условия игры, такие как взаимодействие персонажа с предметами и сбор очков, а также реализовывать алгоритмы подсчета результатов. Эти навыки применимы к разработке игр и интерактивных приложений, что может быть полезно для профессии разработчика игр и мультимедийных проектов в Казахстане. | | |