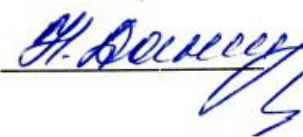


Муниципальное автономное общеобразовательное учреждение города
Новосибирска
«Лицей №22 «Надежда Сибири»
Главный корпус на Советской: г. Новосибирск, ул. Советская, 63, тел. 222-35-15,
e-mail: l_22@edu54.ru
Корпус 99 на Чаплыгина: г. Новосибирск, ул. Чаплыгина, 59, тел. 223-74-15

РАССМОТРЕНО на заседании инженерной кафедры протокол № 1 от 25.08.2025  Кириленко К.А. ФИО руководителя кафедры	СОГЛАСОВАНО Заместитель директора  Н.А. Данилова от 29.08.2025
--	--

РАБОЧАЯ ПРОГРАММА

Информатика. Разработка веб приложений (Фласк)

10 «ИП», 11 «ИП» класса

(уровень среднего общего образования)

Разработчик:

Кириленко Ксения Алексеевна

Рабочая программа по учебному предмету «Информатика. Разработка веб приложений (Флask)» (предметная область «Математика и информатика») (далее соответственно – программа по разработке веб приложений (Флask), Флask) составлена на основе Федеральной рабочей программы по информатике и является авторской, включает пояснительную записку, содержание обучения, планируемые результаты освоения программы по разработке веб приложений (Флask).

Пояснительная записка отражает общие цели и задачи изучения искусственного интеллекта, место в структуре учебного плана, а также подходы к отбору содержания, к определению планируемых результатов.

Содержание обучения раскрывает содержательные линии, которые предлагаются для обязательного изучения в 10 и 11 классе на уровне среднего общего образования.

Планируемые результаты освоения программы по моделированию физических процессов включают личностные, метапредметные результаты за период обучения в 10 - 11 классе на уровне среднего общего образования, а также предметные достижения обучающегося.

1. Пояснительная записка

Общая характеристика учебного предмета "Разработка веб приложений (Флask)"

Разработка веб-приложений представляет собой ключевую дисциплину в современной цифровой экономике, объединяющую дизайн, программирование и инфраструктуру для создания интерактивных систем, работающих через интернет. Изучение основ веб-разработки с использованием микрофреймворка Flask в школе открывает учащимся возможность не просто потреблять, а создавать и «оживлять» веб-контент, развивая глубокое понимание архитектуры веб-приложений и практические навыки программирования, востребованные на рынке IT.

Функциональная значимость предмета для школьников заключается в освоении мощного и элегантного инструмента для создания полнофункциональных веб-сайтов и веб-сервисов, включающих такие фундаментальные концепции, как маршрутизация (routing), обработка HTTP-запросов и ответов, работа с шаблонами (Jinja2), взаимодействие с базами данных (SQLAlchemy, SQLite) и обработка пользовательских данных через формы. Эти знания переводят абстрактные представления о работе интернета в конкретные и осязаемые цифровые продукты, позволяя решать прикладные задачи от создания личного блога и портфолио до разработки API и прототипов коммерческих сервисов.

Знание принципов фреймворка Flask и умение его применять важно для каждого школьника, стремящегося понять, как устроены и создаются современные веб-платформы. Понимание того, как сервер обрабатывает запросы пользователя, динамически генерирует HTML-страницы, хранит и извлекает информацию из базы данных, помогает учащимся развивать логическое, структурное и системное мышление, а также способность к полноценному проектному циклу — от идеи до развертывания (deploy).

Фреймворк Flask, выполняя свои основные функции, позволяет учащимся абстрагироваться от сложной низкоуровневой реализации протоколов и сосредоточиться на концептуальной логике приложения и его архитектуре. Он предоставляет минималистичный, но расширяемый каркас для быстрой разработки, что дает возможность сосредоточиться на содержании, пользовательском опыте, безопасности и качестве кода, что является ключевым этапом в любой профессиональной разработке.

Обучение веб-разработке с помощью Flask направлено на развитие интеллектуальных и технических способностей учащихся, включая умение декомпозировать задачу на отдельные маршруты и функции, проектировать структуру данных, работать с SQL-запросами, разделять логику и представление с помощью шаблонизации и визуализировать результаты в виде интуитивно понятного веб-интерфейса. Это способствует развитию алгоритмического и критического мышления, навыков отладки и тестирования кода, работы с документацией и системами контроля версий (Git), что является crucial для успешного обучения и дальнейшей карьеры в качестве backend- или fullstack-разработчика.

Содержание программы по разработке веб-приложений ориентировано также на развитие функциональной грамотности учащихся, включая умение читать и понимать техническую документацию, использовать сторонние библиотеки и расширения (Flask-Extensions) для построения более сложных систем, оценивать безопасность и производительность своего кода и применять полученные знания для реализации собственных творческих и предпринимательских идей, расширяя свои профессиональные горизонты.

Цели и задачи изучения учебного предмета «Разработка веб приложений (Флask)».

Изучение предмета «Разработка веб-приложений (Flask)» направлено на достижение следующих целей:

- Формирование системного понимания архитектуры веб-приложений клиент-серверного типа как основы современного интернета, связывающего теоретические знания о сетевых протоколах с их практической реализацией средствами программирования на Python.
- Развитие компетенций в области создания, отладки и развертывания полнофункциональных веб-приложений, включающих обработку запросов, шаблонизацию, работу с базами данных и аутентификацию пользователей, с использованием современного микрофреймворка Flask.
- Стимулирование интереса к проектной и предпринимательской деятельности в сфере IT через создание работающих прототипов веб-сервисов, личных блогов, портфолио или API.

Задачи изучения предмета:

- Развивать алгоритмическое и структурное мышление через процесс проектирования моделей данных, маршрутов приложения (routes) и взаимодействия между его компонентами (бэкенд-фронтенд).
- Формировать навыки полноценного жизненного цикла разработки программного продукта: от проектирования UML-диаграмм и написания кода до тестирования, отладки и публикации (деплой) проекта на веб-сервере.
- Воспитывать культуру написания чистого, читаемого и безопасного кода, понимая важность валидации пользовательского ввода и защиты от распространенных уязвимостей (например, SQL-инъекции, XSS).
- Развивать умение самостоятельно учиться: работать с официальной документацией Flask и сторонних библиотек (SQLAlchemy, WTForms), искать решения возникающих проблем в технических блогах, на Stack Overflow и в профессиональных сообществах.

Особенности классов

Рабочая программа по предмету «Информатика. Разработка веб-приложений (Фласк)» для 10-го и 11-го «ИП» классов предназначена для углубленного изучения учащимися информационно-технологического профиля в группах «Искусственный интеллект» и «Наука». На изучение данного модуля отведено 33 часа в 10-м классе и 30 часов в 11-ом классе.

Место предмета в учебном плане лица

Учебный план на изучение «Информатика. Разработка веб-приложений (Фласк)» в 10 и 11 «ИП» классах среднего общего образования отводит 1 учебный час в неделю (всего 33 учебных часа в 10 классе и 30 в 11 классе) за счёт части, формируемой участниками образовательных отношений.

Учебный год	Количество часов	
	10 «ИП»	11 «ИП»
2025/2026	33	30

К тематическому планированию применяется модульный принцип построения образовательной программы, что позволяет выстраивать индивидуальную образовательную парадигму и обеспечивать саморазвитие при индивидуальном темпе работы с учебным материалом, контроль и самоконтроль знаний.

Используемые образовательные технологии, в том числе дистанционные

Обучение искусственному интеллекту может осуществляться с использованием дистанционных образовательных технологий (далее ДОТ), которое предполагает как самостоятельное прохождение учебного материала учеником, так и с помощью сопровождения учителя. При применении ДОТ используются платформы: лицейская

платформа дистанционного обучения Moodle, ФГИС «Моя школа», ГИС «Электронная школа» Новосибирской области.

При реализации рабочей программы могут быть использованы материалы для подготовки к профилям олимпиады КД НТИ и стандартов Всероссийского чемпионатного движения по профессиональному мастерству «Профессионалы».

Информация о промежуточной аттестации

Промежуточная аттестация осуществляется по окончании учебного модуля с целью проверки степени и качества усвоения материала по результатам изучения тематических модулей и проводится в форме аттестационных работ.

Текущий контроль осуществляется с целью проверки степени и качества усвоения материала в ходе его изучения в следующих формах: самостоятельных и проверочных работ.

Текущий контроль и промежуточная аттестация осуществляются в соответствии с «Положением об осуществлении текущего контроля успеваемости и промежуточной аттестации обучающихся, их формах, периодичности и порядке проведения муниципального автономного общеобразовательного учреждения города Новосибирска «Лицей № 22 «Надежда Сибири» (протокол педагогического совета №1 от 29.08.2023).

Итоговая аттестация проводится в соответствии с законодательством РФ.

Промежуточная аттестация по предмету «Разработка веб приложений (Флask)» в 10 «ИП» классе

№ модульн ой	Название модуля	Количество часов модуле	Номер урока ПА	Форма ПА
МР № 1	Разработка веб приложений (Флask)	30	30	Практическая работа

Промежуточная аттестация по предмету «Разработка веб приложений (Флask)» в 11 «ИП» классе

№ модульн ой	Название модуля	Количество часов модуле	Номер урока ПА	Форма ПА
МР № 1	Разработка веб приложений (Флask)	33	33	Практическая работа

2. Содержание учебного предмета

10 класс

Модуль 1 «Разработка веб приложений (Флask)»

Введение во Flask, настройка окружения и первое веб-приложение. Маршрутизация (роутинг) и основы шаблонизатора Jinja2. Наследование шаблонов и подключение статических файлов (CSS, JS). Работа с HTML-формами, методы GET и POST. Обработка и валидация данных из форм на сервере. Введение в базы данных. Основы работы с SQLite. Введение в ORM. Основы SQLAlchemy и создание моделей. Создание и миграция структуры базы данных с помощью Flask-Migrate. CRUD-операции: создание записей в БД (Create). CRUD-операции: чтение записей из БД (Read). CRUD-операции: обновление и удаление записей (Update, Delete). Связи между таблицами (один-ко-многим). Работа с сессиями и cookies для сохранения состояния пользователя.

11 класс

Модуль 1 «Разработка веб приложений (Фласк)»

Введение во Flask, настройка окружения и первое веб-приложение. Маршрутизация (роутинг) и основы шаблонизатора Jinja2. Наследование шаблонов и подключение статических файлов (CSS, JS). Работа с HTML-формами, методы GET и POST. Обработка и валидация данных из форм на сервере. Введение в базы данных. Основы работы с SQLite. Введение в ORM. Основы SQLAlchemy и создание моделей. Создание и миграция структуры базы данных с помощью Flask-Migrate. CRUD-операции: создание записей в БД (Create). CRUD-операции: чтение записей из БД (Read). CRUD-операции: обновление и удаление записей (Update, Delete). Связи между таблицами (один-ко-многим). Работа с сессиями и cookies для сохранения состояния пользователя.

Планируемые образовательные результаты освоения учебного предмета Искусственный интеллект

Личностные результаты

1. сформированность мировоззрения, соответствующего современному уровню развития науки и техники;
2. готовность и способность к образованию, в том числе самообразованию, на протяжении всей жизни; сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
3. навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;
4. эстетическое отношение к миру, включая эстетику научного и технического творчества;

Метапредметные результаты:

1.2.5	Анализировать полученные в ходе решения задачи результаты, критически оценивать их достоверность, прогнозировать изменение в новых условиях
1.2.6	Уметь переносить знания в познавательную и практическую области жизнедеятельности; уметь интегрировать знания из разных предметных областей; осуществлять целенаправленный поиск переноса средств и способов действия в профессиональную среду
1.2.7	Способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания; ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; ставить проблемы и задачи, допускающие альтернативные решения; выдвигать новые идеи, предлагать оригинальные подходы и решения; разрабатывать план решения проблемы с учетом анализа имеющихся материальных и нематериальных ресурсов
1.3	Работа с информацией
1.3.1	Владеть навыками получения информации из источников разных типов, самостоятельно осуществлять поиск, анализ, систематизацию и интерпретацию информации различных видов и форм представления
1.3.2	Создавать тексты в различных форматах с учетом назначения информации и целевой аудитории, выбирая оптимальную форму представления и визуализации
1.3.3	Оценивать достоверность, легитимность информации, ее соответствие правовым и морально-этическим нормам
1.3.4	Использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности
1.3.5	Владеть навыками распознавания и защиты информации, информационной безопасности личности
2	Коммуникативные УУД
2.1	Общение
2.1.1	Осуществлять коммуникации во всех сферах жизни; владеть различными способами общения и взаимодействия
2.1.2	Развернуто и логично излагать свою точку зрения с использованием языковых средств
2.1.3	Аргументированно вести диалог
3	
3.1	Самоорганизация
3.1.1	Самостоятельно осуществлять познавательную деятельность, выявлять проблемы, ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях; давать оценку новым ситуациям
3.1.2	Самостоятельно составлять план решения проблемы с учетом имеющихся ресурсов, собственных возможностей и предпочтений; делать осознанный выбор, аргументировать его, брать ответственность за решение; оценивать приобретенный опыт; способствовать формированию и проявлению широкой эрудиции в разных областях знаний
3.2	Самоконтроль

3.2.1	Давать оценку новым ситуациям, вносить коррективы в деятельность, оценивать соответствие результатов целям
3.2.2	Владеть навыками познавательной рефлексии как осознания совершаемых действий и мыслительных процессов, их результатов и оснований; использовать приемы рефлексии для оценки ситуации, выбора верного решения; уметь оценивать риски и своевременно принимать решения по их снижению
3.3	Эмоциональный интеллект, предполагающий сформированность: саморегулирования, включающего самоконтроль, умение принимать ответственность за свое поведение, способность адаптироваться к эмоциональным изменениям и проявлять гибкость, быть открытым новому; внутренней мотивации, включающей стремление к достижению цели и успеху, оптимизм, инициативность, умение действовать, исходя из своих возможностей

Выпускник научится:

1. самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;
2. продуктивно общаться и взаимодействовать в процессе совместной деятельности, учитывать позиции других участников деятельности, эффективно разрешать конфликты;
3. владеть навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем;
4. использовать средства информационных и коммуникационных технологий в решении когнитивных, коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Выпускник получит возможность научиться:

1. быть готовым и способным к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;
2. быть способным и готовым к самостоятельному поиску методов решения практических задач, применению различных методов познания.

Регулятивные универсальные учебные действия

1. самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута;
2. оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали;
3. ставить и формулировать собственные задачи в образовательной деятельности и жизненных ситуациях;
4. оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели;
5. выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты;
6. организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели;
7. сопоставлять полученный результат деятельности с поставленной заранее целью.

Познавательные универсальные учебные действия

1. искать и находить обобщенные способы решения задач, в том числе, осуществлять развернутый информационный поиск и ставить на его основе новые (учебные и познавательные) задачи;

2. критически оценивать и интерпретировать информацию с разных позиций, распознавать и фиксировать противоречия в информационных источниках;

3. использовать различные модельно-схематические средства для представления существенных связей и отношений, а также противоречий, выявленных в информационных источниках;

4. находить и приводить критические аргументы в отношении действий и суждений другого; спокойно и разумно относиться к критическим замечаниям в отношении собственного суждения, рассматривать их как ресурс собственного развития;

5. выходить за рамки учебного предмета и осуществлять целенаправленный поиск возможностей для широкого переноса средств и способов действия;

6. выстраивать индивидуальную образовательную траекторию, учитывая ограничения со стороны других участников и ресурсные ограничения;

7. менять и удерживать разные позиции в познавательной деятельности.

Коммуникативные универсальные учебные действия

1. осуществлять деловую коммуникацию как со сверстниками, так и со взрослыми (как внутри образовательной организации, так и за ее пределами), подбирать партнеров для деловой коммуникации исходя из соображений результативности взаимодействия, а не личных симпатий;

2. при осуществлении групповой работы быть как руководителем, так и членом команды в разных ролях (генератор идей, критик, исполнитель, выступающий, эксперт и т.д.);

3. координировать и выполнять работу в условиях реального, виртуального и комбинированного взаимодействия;

4. развернуто, логично и точно излагать свою точку зрения с использованием адекватных (устных и письменных) языковых средств;

5. распознавать конфликтогенные ситуации и предотвращать конфликты до их активной фазы, выстраивать деловую и образовательную коммуникацию, избегая личностных оценочных суждений.

Предметные результаты

Выпускник будет демонстрировать:

1) систематизация знаний, относящихся к *математическим объектам информатики*; умение строить математические объекты информатики, в том числе логические формулы;

2) сформированность базовых навыков и умений по соблюдению требований *техники безопасности*, гигиены и ресурсосбережения при работе со средствами информатизации;

3) владение опытом построения и использования *компьютерно-математических моделей*, проведения экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов; сформированность представлений о необходимости *анализа соответствия модели* и моделируемого объекта (процесса);

4) сформированность представлений о способах хранения и простейшей обработке данных; умение пользоваться *базами данных* и справочными системами; владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними;

Выпускник получит возможность продемонстрировать:

1. владение системой базовых знаний, отражающих *вклад информатики* в формирование современной научной картины мира;

2. владение навыками *алгоритмического мышления* и понимание необходимости формального описания алгоритмов.

ПРОВЕРЯЕМЫЕ НА ЕГЭ ПО ИНФОРМАТИКЕ ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

Код проверяемого требования	Проверяемые требования к предметным результатам освоения основной образовательной программы среднего общего образования
<i>1.</i>	<i>Знать (понимать)</i>
1.1	Понимание основных принципов устройства и функционирования современных стационарных и мобильных компьютеров; тенденций развития компьютерных технологий; владение навыками работы с операционными системами и основными видами программного обеспечения для решения учебных задач по выбранной специализации
1.4	Понимание базовых алгоритмов обработки числовой и текстовой информации (запись чисел в позиционной системе счисления, делимость целых чисел; нахождение всех простых чисел в заданном диапазоне; обработка многоразрядных целых чисел; анализ символьных строк и других), алгоритмов поиска и сортировки
1.5	Знание функциональные возможности инструментальных средств среды разработки
1.6	Владение основными сведениями о базах данных, их структуре, средствах создания и работы с ними
1.7	Понимание возможностей и ограничений технологий искусственного интеллекта в различных областях; наличие представлений об использовании информационных технологий в различных профессиональных сферах
<i>2.</i>	<i>Уметь</i>
2.1	Умение использовать компьютерно-математические модели для анализа объектов и процессов: формулировать цель моделирования, выполнять анализ результатов, полученных в ходе моделирования; оценивать адекватность модели моделируемому объекту или процессу; представлять результаты моделирования в наглядном виде
2.2	Умение классифицировать основные задачи анализа данных (прогнозирование, классификация, кластеризация, анализ отклонений); понимать последовательность решения задач анализа данных: сбор первичных данных, очистка и оценка качества данных, выбор и (или) построение модели, преобразование данных, визуализация данных, интерпретация результатов

Код проверяемого требования	Проверяемые требования к предметным результатам освоения основной образовательной программы среднего общего образования
2.9	Умение анализировать алгоритмы с использованием таблиц трассировки; определять без использования компьютера результаты выполнения несложных программ, включающих циклы, ветвления и подпрограммы, при заданных исходных данных
2.10	Умение определять сложность изучаемых в курсе базовых алгоритмов (суммирование элементов массива, сортировка массива, переборные алгоритмы, двоичный поиск) и приводить примеры нескольких алгоритмов разной сложности для решения одной задачи
2.11	Владение универсальным языком программирования высокого уровня (Паскаль, Python, Java, C++, C#), представлениями о базовых типах данных и структурах данных; умение использовать основные управляющие конструкции; умение осуществлять анализ предложенной программы: определять результаты работы программы при заданных исходных данных; определять, при каких исходных данных возможно получение указанных результатов; выявлять данные, которые могут привести к ошибке в работе программы; формулировать предложения по улучшению программного кода
2.12	Умение реализовывать на выбранном для изучения языке программирования высокого уровня (Паскаль, Python, Java, C++, C#) типовые алгоритмы обработки чисел, числовых последовательностей и массивов: представление числа в виде набора простых сомножителей; нахождение максимальной (минимальной) цифры натурального числа, записанного в системе счисления с основанием, не превышающим 10; вычисление обобщённых характеристик элементов массива или числовой последовательности (суммы, произведения среднего арифметического, минимального и максимального элементов, количества элементов, удовлетворяющих заданному условию); сортировку элементов массива; умение использовать в программах данные различных типов с учётом ограничений на диапазон их возможных значений, применять при решении задач структуры данных (списки, словари, стеки, очереди, деревья); применять стандартные и собственные подпрограммы для обработки числовых данных и символьных строк; использовать при разработке программ библиотеки подпрограмм; умение использовать средства отладки программ в среде программирования

ПЕРЕЧЕНЬ ЭЛЕМЕНТОВ СОДЕРЖАНИЯ, ПРОВЕРЯЕМЫХ НА ЕГЭ ПО ИНФОРМАТИКЕ

Код	Проверяемый элемент содержания
3	Алгоритмы и программирование
3.1	Формализация понятия алгоритма. Машина Тьюринга как универсальная модель вычислений
3.2	Оценка сложности вычислений. Время работы и объём используемой памяти, их зависимость от размера исходных данных. Оценка асимптотической сложности алгоритмов. Алгоритмы полиномиальной сложности. Переборные алгоритмы. Примеры различных алгоритмов решения одной задачи, которые имеют различную сложность
3.3	Определение возможных результатов работы простейших алгоритмов управления исполнителями и вычислительных алгоритмов. Определение исходных данных, при которых алгоритм может дать требуемый результат
3.4	Алгоритмы обработки натуральных чисел, записанных в позиционных системах счисления: разбиение записи числа на отдельные цифры, нахождение суммы и произведения цифр, нахождение максимальной (минимальной) цифры. Представление числа в виде набора простых сомножителей. Алгоритм быстрого возведения в степень. Поиск простых чисел в заданном диапазоне с помощью алгоритма «решето Эратосфена»
3.5	Многоразрядные целые числа, задачи длинной арифметики
3.6	Язык программирования (Паскаль, Python, Java, C++, C#). Типы данных: целочисленные, вещественные, символьные, логические. Ветвления. Сложные условия. Циклы с условием. Циклы по переменной. Обработка данных, хранящихся в файлах. Текстовые и двоичные файлы. Файловые переменные (файловые указатели). Чтение из файла. Запись в файл. Разбиение задачи на подзадачи. Подпрограммы (процедуры и функции). Использование стандартной библиотеки языка программирования
3.7	Рекурсия. Рекурсивные процедуры и функции. Использование стека для организации рекурсивных вызовов
3.8	Численные методы. Точное и приближённое решения задачи. Численное решение уравнений с помощью подбора параметра. Численные методы решения уравнений: метод перебора, метод половинного деления. Приближённое вычисление длин кривых. Вычисление площадей фигур с помощью численных методов (метод прямоугольников, метод трапеций). Поиск максимума (минимума) функции одной переменной методом половинного деления

Код	Проверяемый элемент содержания
3.9	Обработка символьных данных. Встроенные функции языка программирования для обработки символьных строк. Алгоритмы обработки символьных строк: подсчёт количества появлений символа в строке, разбиение строки на слова по пробельным символам, поиск подстроки внутри данной строки, замена найденной подстроки на другую строку. Генерация всех слов в некотором алфавите, удовлетворяющих заданным ограничениям. Преобразование числа в символьную строку и обратно
3.12	Словари (ассоциативные массивы, отображения). Хэш-таблицы. Построение алфавитно-частотного словаря для заданного текста
3.13	Стеки. Анализ правильности скобочного выражения. Вычисление арифметического выражения, записанного в постфиксной форме. Очереди. Использование очереди для временного хранения данных
3.15	Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения. Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева
3.17	Понятие об объектно-ориентированном программировании. Объекты и классы. Свойства и методы объектов. Объектно-ориентированный анализ. Разработка программ на основе объектно-ориентированного подхода. Инкапсуляция, наследование, полиморфизм

3. Тематическое планирование

10 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контрол ьные работы	Практич еские работы	
Модуль №1 «Разработка веб-приложений (Флask)» - 33 часа					
1.1	Разработка веб приложений (Флask)	31		3	https://stepik.org/course/97540/promo
1.2	Итоговый проект	2		1	

11 класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательные ресурсы
		Всего	Контроль ные работы	Практич еские работы	

Модуль №1 «Разработка веб-приложений (Флask)» - 30 часов					
1.1	Разработка веб приложений (Флask)	28		3	https://stepik.org/course/97540/promo
1.2	Итоговый проект	2		1	

5. Приложения к программе

ПОУРОЧНОЕ ПЛАНИРОВАНИЕ

10 Класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательн ые ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Разработка веб приложений (Флask) - 33 часа					
1.1	Введение во Flask, настройка окружения и первое веб- приложение.	2			https://stepik.org/course/97540/promo
1.2	Маршрутизация (роутинг) и основы шаблонизатора Jinja2.	2			
1.3	Наследование шаблонов и подключение статических файлов (CSS, JS).	2			
1.4	Работа с HTML- формами, методы GET и POST.	2			
1.5	Обработка и валидация данных из форм на сервере.	2			
1.6	Введение в базы	2			

	данных. Основы работы с SQLite.				
1.7	Введение в ORM. Основы SQLAlchemy и создание моделей.	2			
1.8	Создание и миграция структуры базы данных с помощью Flask-Migrate.	2			
1.9	CRUD-операции: создание записей в БД (Create).	2			
1.10	CRUD-операции: чтение записей из БД (Read).	2			
1.11	CRUD-операции: обновление и удаление записей (Update, Delete).	2			
1.12	Связи между таблицами (один-ко-многим).	2			
1.13	Работа с сессиями и cookies для сохранения состояния пользователя.	2			
1.14	Проект «Блог»: Проектирование и настройка. Реализация вывода постов.	2		1	
1.15	Проект «Блог»: Реализация добавления новых постов через форму.	2		1	
1.16	Проект «Блог»: Реализация редактирования и удаления постов.	2		1	

1.17	Подведение итогов. Защита итоговых проектов.	2			
------	--	---	--	--	--

11 Класс

№ п/п	Наименование разделов и тем программы	Количество часов			Электронные (цифровые) образовательн ые ресурсы
		Всего	Контрольные работы	Практические работы	
Модуль 1. Разработка веб приложений (Фласк) - 33 часа					
1.1	Введение во Flask, настройка окружения и первое веб- приложение.	2			https://stepik.org/course/97540/promo
1.2	Маршрутизация (роутинг) и основы шаблонизатора Jinja2.	2			
1.3	Наследование шаблонов и подключение статических файлов (CSS, JS).	2			
1.4	Работа с HTML- формами, методы GET и POST.	2			
1.5	Обработка и валидация данных из форм на сервере.	2			
1.6	Введение в базы данных. Основы работы с SQLite.	2			
1.7	Введение в ORM. Основы	2			

	SQLAlchemy и создание моделей.				
1.8	Создание и миграция структуры базы данных с помощью Flask-Migrate.	2			
1.9	CRUD-операции: создание записей в БД (Create).	2			
1.10	CRUD-операции: чтение записей из БД (Read).	2			
1.11	CRUD-операции: обновление и удаление записей (Update, Delete).	2			
1.12	Связи между таблицами (один-ко-многим).	2			
1.13	Работа с сессиями и cookies для сохранения состояния пользователя.	2			
1.14	Проект «Блог»: Проектирование и настройка. Реализация вывода постов.	2		1	
1.15	Подведение итогов. Защита итоговых проектов.	3			

КИМ

10 и 11 класс

Модуль 1 «Разработка веб приложений»

Задание 1: Настройка проекта и базовая структура (Настройка окружения, Первое приложение)

1 Создайте новую директорию для проекта my_blog.

2 Создайте и активируйте виртуальное окружение.

bash

Copy Download

```
python -m venv venv
source venv/bin/activate # для Linux/MacOS
# или
venv\Scripts\activate.bat # для Windows
```

3 Установите необходимые зависимости:

bash

Copy Download

```
pip install flask flask-sqlalchemy flask-migrate
```

4 Создайте файл app.py. Импортируйте Flask и создайте экземпляр приложения.

5 Создайте базовый маршрут /, который возвращает строку "Главная страница моего блога!".

6 Запустите приложение и убедитесь, что по адресу <http://127.0.0.1:5000/> все работает.

Задание 2: Шаблоны и статические файлы (Jinja2, Наследование, Статика)

1 Создайте структуру папок:

```
my_blog/
├── app.py
├── templates/
│   ├── base.html
│   └── index.html
├── static/
│   ├── css/
│   │   └── style.css
│   └── js/
│       └── script.js
```

2 Создайте базовый шаблон templates/base.html. Он должен содержать базовый HTML, блоки для контента (`{% block content %}{% endblock %}`) и заголовка. Подключите файл static/css/style.css.

3 Создайте шаблон templates/index.html, который наследует от base.html. В блоке content выведите заголовок "Последние посты".

4 Измените маршрут / в app.py, чтобы он рендерил шаблон index.html с помощью функции render_template.

5 Напишите несколько простых стилей в `static/css/style.css` (например, измените цвет фона `body`). Убедитесь, что стили применяются.

Задание 3: Модели данных и настройка БД (SQLAlchemy, Flask-Migrate)

1 Настройте базу данных. В `app.py` добавьте конфигурацию для SQLite (`SQLALCHEMY_DATABASE_URI`) и инициализируйте экземпляр SQLAlchemy и Migrate.

2 Создайте модель Post с полями:

- `id` (Integer, `primary_key=True`)
- `title` (String(100), `nullable=False`)
- `content` (Text, `nullable=False`)
- `created_at` (DateTime, `default=datetime.utcnow`)

3 В терминале инициализируйте миграции и создайте первую миграцию.

```
bash

flask db init
flask db migrate -m "Initial migration - Post model"
flask db upgrade
```

4 Убедитесь, что файл базы данных `app.db` создан.

Задание 4: Форма для добавления поста (HTML-формы, GET/POST)

1 Создайте новую функцию представления (view function) для отображения формы по адресу `/add_post`.

2 Создайте шаблон `add_post.html` с HTML-формой.

Форма должна использовать метод POST.

3 Добавьте поля: `<input type="text" name="title">` и `<textarea name="content"></textarea>`.

4 Добавьте кнопку отправки формы.

Пока что просто выводите полученные из формы данные с помощью `request.form` в консоль. Не забудьте импортировать `request` из `flask` и указать методы ['GET', 'POST'] для маршрута.

Задание 5: CRUD - Create (Создание поста, Валидация)

1 Модифицируйте маршрут `/add_post`. Если метод запроса POST:

2 Проведите базовую валидацию: проверьте, что оба поля не пустые.

- 3 Создайте новый экземпляр модели Post, передав в него данные из формы.
- 4 Добавьте запись в сессию и сохраните ее в БД.
- 5 Перенаправьте пользователя на главную страницу (/) с помощью `redirect(url_for('index'))`.
- 6 Добавьте в навигационную панель (в `base.html`) ссылку на страницу добавления поста.

Задание 6: CRUD - Read (Чтение постов)

Модифицируйте маршрут / (главная страница).
Добавьте в него запрос к базе данных, который получит ВСЕ записи из таблицы Post (или несколько последних, используя `order_by(Post.created_at.desc())`).
Передайте полученный список постов в шаблон `index.html`.
Измените шаблон `index.html`. Используя цикл Jinja2, выведите все полученные посты: отобразите заголовок, содержимое и дату создания для каждого.

Задание 7: CRUD - Update & Delete (Обновление и удаление)

Создайте новый маршрут `/edit_post/<int:post_id>`.
Для метода GET: Найди пост по его `id` и отобрази форму `edit_post.html`, предзаполненную данными этого поста.
Для метода POST: Найди пост, обнови его данные из формы и сохрани изменения в БД. Перенаправь на главную.
Создайте новый маршрут `/delete_post/<int:post_id>` (обычно для удаления используют метод POST, но для простоты можно и GET).
В этом маршруте: Найди пост по `id`, удали его из БД и перенаправь на главную.
В шаблоне `index.html` рядом с каждым постом добавьте ссылки "Редактировать" (ведущую на `/edit_post/...`) и "Удалить" (ведущую на `/delete_post/...`).

Задание 8 (Повышенной сложности): Аутентификация пользователя (Сессии, Cookies)*

Создайте модель User с полями `id`, `username` и `password`.
Создайте формы для регистрации и входа.
Реализуйте views для регистрации (`/register`) и входа (`/login`).

При успешном входе сохраняйте `user_id` в сессии (`session['user_id'] = user.id`).
Создайте декоратор `@login_required`, который будет проверять, авторизован ли пользователь.
Защитите маршруты `/add_post`, `/edit_post/...`, `/delete_post/...` этим декоратором.
Добавьте в модель Post связь "один-ко-многим" с моделью User (внешний ключ `user_id`). При создании поста автоматически привязывайте его к текущему пользователю.
Измените отображение и функционал: показывайте кнопки редактирования/удаления только для постов, принадлежащих текущему пользователю.