Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное
образовательное учреждение высшего образования
«Казанский (Приволжский) федеральный университет»
Институт физики

УТВЕРЖДАЮ

Проректор
по образовательной деятельности
_____ Е.А. Турилова
(подпись)
«____» _____ 2024 г.

Дополнительная профессиональная программа
повышения квалификации
«Использование инструментов ИИ при генерации и анализе кода на Питоне»

Утверждена Учебно-методической комиссией Института физики КФУ (протокол №1 от «10» сентября 2024 г.)

Председатель комиссии _____ О.В. Недопекин
(подпись)

Директор Института физики _____ М.Р. Гафуров
(подпись)

Казань – 2024

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Цель реализации программы

Целью реализации программы повышения квалификации является совершенствование и получение новых компетенций в области информационных технологий, необходимых для профессиональной деятельности.

Слушатель программы получит возможность овладеть следующими основными компетенциями:

- Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
- Владение навыками использования операционных систем, сетевых технологий, современных языков программирования, технологий обработки данных, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных;
- Способность разрабатывать и адаптировать прикладное программное обеспечение, выполнять интеграцию с существующими ИС заказчика;
- Способность проводить тестирование компонентов программного обеспечения ИС и оценивать качество программного обеспечения (надежность, производительность, безопасность, удобство использования);

1.2. Планируемые результаты обучения

Обучающийся, освоивший дисциплину:

Должен знать:

- основные принципы и концепции алгоритмов и процедурного подхода в программировании;
- различные фазы в разработке и поддержке программного обеспечения;
- процесс преобразования кода в исполняемые файлы;
- типы данных, встроенные в язык Python;
- ключевые элементы синтаксиса Python;
- методы организации и структурирования данных;
- основные аспекты, касающиеся работы с функциями;
- основы промт-инжиниринга для генерации и анализа кода с использованием ИИ;
- основные методы анализа и оптимизации кода с помощью инструментов ИИ;
- принципы машинного обучения и его применение для анализа кода.

Должен уметь:

- эффективно структурировать код и разделять задачи на более мелкие подзадачи;
- правильно использовать языковые операторы и команды для достижения целей программирования;
- разрабатывать функции с учетом оптимальной передачи параметров и использования ресурсов;
- решать практические задачи с использованием языковых возможностей для создания базовых консольных приложений;
- использовать OpenAI API для генерации кода на Python;

- применять статический и динамический анализ кода для выявления и исправления ошибок;
- оптимизировать код с использованием инструментов ИИ;
- интегрировать ИИ-инструменты в рабочий процесс разработки и CI/CD pipeline.

Должен владеть навыками:

- создания эффективных консольных программ с использованием языка Python, сочетающих лучшие практики и современные методологии программирования;
- разработки и оптимизации промтов для генерации кода;
- анализа и тестирования кода с использованием библиотек Python;
- применения машинного обучения для анализа и улучшения качества кода.

Должен демонстрировать способность и готовность:

- применять информационные технологии в задачах автоматизации прикладных процессов, включая сетевые приложения, базы данных, системное и прикладное программное обеспечение;
- проводить тестирование компонентов ИС по заданным сценариям, оценивать качество и надежность компонентов ИС;
- разрабатывать и интегрировать прикладное программное обеспечение, компоненты информационной системы;
- создавать прототипы информационных систем;
- использовать ИИ-инструменты для автоматического анализа, генерации и оптимизации кода.

1.3. **Требования к уровню подготовки поступающего на обучение:** наличие высшего образования.

1.4. **Программа разработана на основе:** профессионального стандарта «Программист», утвержденного приказом Министерства труда и социальной защиты РФ от 18.11.2013 №679н.

1.5. **Форма обучения** – очная.

Продолжительность обучения всего: 144 часа, из них аудиторных занятий – 72 часа, самостоятельная работа обучающихся – 72 часа.

По завершении обучения проводится итоговая аттестация. При успешном прохождении итоговой аттестации слушателям выдается удостоверение о повышении квалификации установленного КФУ образца.

2. СОДЕРЖАНИЕ ПРОГРАММЫ

2.1. Учебный план

Наименование раздела, темы	Трудоемкость, час	Аудиторные занятия			Самостоятельная работа, час.
		Всего, час.	в том числе		
			лекции	практич. занятия, семинары	
1	2	3	4	5	6
Тема 1. Введение в инструменты ИИ для программирования	8	4	2	2	4
Тема 2. Использование библиотеки OpenAI для генерации кода	14	6	4	2	8
Тема 3. Основы промт-инжиниринга	14	6	4	2	8
Тема 4. Анализ кода с помощью инструментов ИИ	16	8	4	4	8
Тема 5. Оптимизация кода с использованием ИИ	16	8	4	4	8
Тема 6. Обнаружение ошибок и багов с помощью ИИ	16	8	4	4	8
Тема 7. Применение машинного обучения для анализа кода на уязвимости	24	12	6	6	12
Тема 8. Интеграция ИИ-инструментов в рабочий процесс разработки	32	16	8	8	16
Всего	140	68	36	32	72
Итоговая аттестация	4	4	0	4	0
Итого часов	**144**	**72**	**36**	**36**	**72**

2.2. Календарный учебный график

Период обучения (дни, недели)[1)]	Наименование раздела
1-7 дни	Тема 1. Введение в инструменты ИИ для программирования Тема 2. Использование библиотеки OpenAI для генерации кода Тема 3. Основы промт-инжиниринга
8-13 дни	Тема 4. Анализ кода с помощью инструментов ИИ Тема 5. Оптимизация кода с использованием ИИ
14-19 дни	Тема 6. Обнаружение ошибок и багов с помощью ИИ Тема 7. Применение машинного обучения для анализа кода на уязвимости Тема 8. Интеграция ИИ-инструментов в рабочий процесс разработки Итоговая аттестация

1)Даты обучения будут определены в расписании занятий при наборе группы на обучение

2.3. Рабочие программы разделов

Наименование темы	Содержание лекций (количество часов)	Наименование практических занятий или семинаров (количество часов)	Виды СРС (количество часов)
1	2	3	4
Тема 1. Введение в инструменты ИИ для программирования (4 ч.)	Основы искусственного интеллекта и машинного обучения Обзор инструментов ИИ для генерации и анализа кода Введение в Python как основной язык для ИИ-приложений (2 ч.)	Установка Python и необходимых библиотек Настройка среды разработки (IDE) Подключение к OpenAI API и выполнение первых запросов (2 ч.)	Работа с литературой. Подготовка к практическим занятиям (4 часа).
Тема 2. Использование библиотеки OpenAI для генерации кода (6 ч.)	Установка и настройка OpenAI API Основные возможности и методы OpenAI API Генерация кода на Python с использованием моделей GPT (4 ч.)	Генерация простых программ на Python Создание более сложных программных модулей с помощью OpenAI Оценка качества сгенерированного кода (2 ч.)	Работа с литературой. Подготовка к практическим занятиям (8 часов).
Тема 3. Основы промт-инжиниринга (6 ч.)	Введение в промт-инжиниринг Разработка и оптимизация промтов для генерации кода Примеры успешного использования промтов в различных задачах (4 ч.)	Создание эффективных промтов для генерации кода Оптимизация промтов для улучшения результатов Практические примеры и задачи на промт-инжиниринг (2 ч.)	Работа с литературой. Подготовка к практическим занятиям (8 часов).
Тема 4. Анализ кода с помощью инструментов ИИ (8 ч.)	Введение в анализ кода и его важность Использование библиотек для статического анализа кода (например, Pylint, Flake8) Применение ИИ для динамического анализа кода (4 ч.)	Установка и использование Pylint и Flake8 Анализ кода и выявление ошибок Автоматизация статического анализа в процессе разработки (4 ч.)	Работа с литературой. Подготовка к практическим занятиям (8 часов).
Тема 5. Оптимизация кода с	Основные методы оптимизации кода Использование ИИ для автоматической оптимизации	Использование unittest и pytest для тестирования кода	Работа с литературой. Подготовка к практическим

использование м ИИ (8 ч.)	Примеры и кейсы по улучшению производительности кода (4 ч.)	Применение динамического анализа для выявления проблем Интеграция инструментов тестирования с ИИ-моделями (4 ч.)	занятиям (8 часов).
Тема 6. Обнаружение ошибок и багов с помощью ИИ (8 ч.)	Введение в отладку программного обеспечения Применение ИИ для автоматического обнаружения ошибок Интеграция ИИ-инструментов с системами CI/CD (4 ч.)	Анализ производительности кода Применение ИИ для предложений по оптимизации Рефакторинг и улучшение кода (4 ч.)	Работа с литературой. Подготовка к практическим занятиям (8 часа).
Тема 7. Применение машинного обучения для анализа кода на уязвимости (12 ч.)	Введение в машинное обучение и его применение в анализе кода Построение моделей для предсказания качества и уязвимостей кода Использование библиотек scikit-learn и TensorFlow для анализа кода (6 ч.)	Построение и обучение моделей на основе данных о коде Анализ результатов и предсказание уязвимостей Применение моделей для реальных проектов (6 ч.)	Работа с литературой. Подготовка к практическим занятиям (12 часов).
Тема 8. Интеграция ИИ-инструментов в рабочий процесс разработки (16 ч.)	Подходы к интеграции ИИ-инструментов в DevOps Примеры успешных интеграций Будущее ИИ в разработке ПО (8 ч.)	Настройка CI/CD pipeline с интеграцией ИИ-инструментов Автоматизация анализа и тестирования кода Мониторинг и улучшение процесса разработки (8 ч.)	Работа с литературой. Подготовка к практическим занятиям (16 часов).

часов

2.4. Оценка качества освоения программы

2.4.1. Формы контроля и аттестации

Оценка качества освоения программы включает итоговую аттестацию.

К итоговой аттестации допускается слушатель, успешно завершивший в полном объеме освоение программы повышения квалификации.

Итоговая аттестация проводится в форме письменного экзамена по билетам.

2.4.2. Оценочные материалы

Примерные вопросы экзамена по билетам:

Билет 1

Вопрос 1: Использование OpenAI API для генерации кода

- Объясните, как подключиться к OpenAI API и выполнить первый запрос на генерацию кода.
- Приведите пример кода на Python, который использует OpenAI API для генерации простой функции.

Вопрос 2: Основы промт-инжиниринга

- Что такое промт-инжиниринг и почему он важен?
- Опишите основные принципы создания эффективных промтов для генерации кода.

Билет 2

Вопрос 1: Статический анализ кода

- Объясните, что такое статический анализ кода и почему он важен.
- Приведите пример использования Pylint для анализа Python-кода и опишите, какие ошибки и предупреждения могут быть обнаружены.

Вопрос 2: Оптимизация кода с использованием ИИ

- Какие методы оптимизации кода можно применить с помощью ИИ?
- Приведите пример оптимизации кода на Python с использованием инструментов ИИ.

Билет 3

Вопрос 1: Динамический анализ и тестирование кода

- Объясните разницу между статическим и динамическим анализом кода.
- Приведите пример использования библиотеки unittest для тестирования функции на Python.

Вопрос 2: Обнаружение ошибок с помощью ИИ

- Как ИИ может помочь в автоматическом обнаружении ошибок в коде?
- Приведите пример использования ИИ для автоматического исправления ошибок в Python-коде.

Билет 4

Вопрос 1: Применение машинного обучения для анализа кода

- Опишите процесс предобработки данных для анализа кода с использованием машинного обучения.
- Приведите пример модели машинного обучения, используемой для предсказания качества кода.

Вопрос 2: Интеграция ИИ-инструментов в CI/CD

- Какие преимущества дает интеграция ИИ-инструментов в CI/CD pipeline?
- Приведите пример настройки CI/CD pipeline с использованием ИИ-инструментов для анализа кода.

Билет 5

Вопрос 1: Основы объектно-ориентированного программирования

- Объясните основные принципы ООП: инкапсуляция, наследование и полиморфизм.
- Приведите пример класса на Python, который демонстрирует использование этих принципов.

Вопрос 2: Введение в библиотеки Python для научных вычислений

- Какие библиотеки используются для научных вычислений в Python?
- Приведите примеры использования библиотек NumPy и SciPy для решения задач научных вычислений.

Критерии оценки результатов:

Оценка «неудовлетворительно» ставится, если:

•в ответе обнаруживается отсутствие владения материалом в объёме изучаемой ДПП;

•в ответе на вопросы не дается трактовка основных понятий, при их употреблении;

•ответы на вопросы не имеют логически выстроенного характера, не используются такие мыслительные операции как сравнение, анализ и обобщение.

Оценка «удовлетворительно» ставится, если:

•в ответах на вопросы при раскрытии содержания вопросов недостаточно раскрываются и анализируются основные противоречия и проблемы;

•при раскрытии особенностей развития тех или иных профессиональных идей, а также описания профессиональной деятельности недостаточно используются материалы современных пособий и первоисточников, допускаются фактические ошибки;

•при ответе используется терминология и дается её определение без ссылки на авторов (теоретиков и практиков);

•ответы на вопросы не имеют логически выстроенного характера, редко используются такие мыслительные операции как сравнение, анализ и обобщение;

•личная точка зрения слушателя носит формальный характер без умения ее обосновывать и доказывать.

Оценка «хорошо» ставится, если:

•ответы на вопросы частично носят проблемный характер, при раскрытии особенностей развития тех или иных профессиональных идей, а также описании профессиональной деятельности используются материалы современных пособий и первоисточников;

•в ответе используется терминология соответствующая конкретному периоду развития теории и практики профессиональной деятельности, где определение того или иного понятия формулируется без знания контекста его развития в системе профессионального понятийного аппарата;

•ответы на вопрос не имеют логически выстроенного характера, но используются такие мыслительные операции как сравнение, анализ и обобщение;

•имеется личная точка зрения слушателя, основанная на фактическом и проблемном материале, приобретенном на лекционных, семинарских, практических занятиях и в результате самостоятельной работы.

Оценка «отлично» ставится, если:

•ответы на вопросы носят проблемный характер, при раскрытии особенностей развития тех или иных профессиональных идей, их описании используются материалы современных учебных пособий и первоисточников;

•в ответе используется терминология, соответствующая конкретному периоду развития теории и практики, и четко формулируется определение, основанное на понимании контекста;

•ответы на вопрос имеют логически выстроенный характер, часто используются такие мыслительные операции как сравнение, анализ и обобщение;

•ярко выражена личная точка зрения слушателя при обязательном владении фактическим и проблемным материалом, полученным на лекционных, практических, семинарских занятиях и в результате самостоятельной работы.

3. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

3.1. Требуемые материально-технические условия:

Наименование специализированных учебных помещений	Вид занятий	Наименование оборудования, программного обеспечения
Учебный класс	Лекция	Мультимедийное оборудование, проектор, компьютеры
Учебный класс	Практическое занятие	Мультимедийное оборудование, проектор, компьютеры

3.2. Учебно-методическое и информационное обеспечение

Основная литература

1. Зыкова, Г. В. Основы программирования на языке Python : учебно-методическое пособие / Г. В. Зыкова, А. С. Попов, Т. Н. Сапуглецева ; под. ред. Г. В. Зыковой. - 2-е изд., стер. - Москва : ФЛИНТА, 2020. - 135 с. - ISBN 978-5-9765-4430-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1860057 (дата обращения: 30.06.2024). – Режим доступа: по подписке.
2. Целых, А. Н. Извлечение знаний методами машинного обучения : учебное пособие по курсам «Модели и методы инженерии знаний», «Методы машинного обучения» / А. Н. Целых, Э. М. Котов ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2022. - 105 с. - ISBN 978-5-9275-4215-4. - Текст : электронный. - URL: https://znanium.com/catalog/product/2132253 (дата обращения: 30.06.2024). – Режим доступа: по подписке.

Дополнительная литература

1. Лимановская, О. В. Основы машинного обучения : учебное пособие / О. В. Лимановская, Т. И. Алферьева. - 2-е изд., стер. - Москва : ФЛИНТА : Изд-во Урал. ун-та, 2022. - 88 с. - ISBN 978-5-9765-5006-3 (ФЛИНТА) ; ISBN 978-5-7996-3015-7 (Изд-во Урал. ун-та). - Текст : электронный. - URL: https://znanium.com/catalog/product/1891377 (дата обращения: 30.06.2024). – Режим доступа: по подписке.
2. Лекун, Я. Как учится машина : Революция в области нейронных сетей и глубокого обучения : научно-популярное издание / Я. Лекун. - Москва : Альпина ПРО, 2021. - 335 с. - ISBN 978-5-907394-92-6. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2138131 (дата обращения: 30.06.2024). – Режим доступа: по подписке.
3. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное

пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. - 146 с. - ISBN 978-5-9275-2649-9. - Текст : электронный. - URL: https://znanium.com/catalog/product/1021662 (дата обращения: 30.06.2024). – Режим доступа: по подписке.

3.3. Кадровые условия

Кадровое обеспечение программы осуществляет профессорско-преподавательский состав из числа докторов и кандидатов наук и высококвалифицированных специалистов Казанского федерального университета.

4. РУКОВОДИТЕЛЬ И АВТОР ПРОГРАММЫ

Руководитель:
Гарнаева Гузель Ильдаровна, заместитель директора по образовательной деятельности Института физики.
Автор:
Корчагин Павел Анатольевич, старший преподаватель кафедры радиофизики Института физики.