



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ:

Проректор по образовательной деятельности,
председатель методического совета

И.Р. Абдулвелсев

9 февраля 2023 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ОСНОВЫ ООП И MVC НА PYTHON

Для основных образовательных программ
с индивидуальной образовательной траекторией

Уровень высшего образования – бакалавриат

Форма обучения

Очная

Курс 3
Семестр 5

Магнитогорск
2023 г.

Рабочая программа рассмотрена и одобрена на заседании методического совета
09.02.2023, протокол № 1.

Согласовано с руководителями ООП:

Зав. кафедрой ЭПП

А.В. Варганова

Зав. кафедрой экономики

А.Г. Васильева

Зам. директора ИЕиС по воспитательной работе,
доцент кафедры ТССА

А.С. Лимарев

Доцент кафедры ПОиД

Т.Г. Неретина

Зам. директора ИЕиС по учебной работе,
доцент кафедры ПОиБЖД

Ю.В. Сомова

Зав. кафедрой УиИС

М.М. Суровцов

Зав. кафедрой ЛПиМ

Н.А. Феоктистов

Зав. кафедрой ЛиУТС

О.В. Фридрихсон

Зав. кафедрой МиХТ

А.С. Харченко

1 Цели освоения дисциплины (модуля)

Целью дисциплины «Основы ООП и MVC на Python» является изучение основ объектно-ориентированного программирования, концепции Model-View-Controller и формирование у обучающихся навыков их практического применения на примере языка программирования Python.

Дисциплина «Основы ООП и MVC на Python» предусматривает освоение методологий объектно-ориентированного и функционального программирования, концепции Model-View-Controller с целью приобретения навыков построения эффективных и легко сопровождаемых приложений, а также освоение методов тестирования и отладки программ.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Основы ООП и MVC на Python входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Основы ООП и MVC на Python» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ДПК-004-3	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ
ДПК-004-3.1	Проводит комплексное тестирование и отладку программных систем
ДПК-004-3.2	Проектирует программное обеспечение с использованием средств автоматизации

2.1 Основные принципы объектно-ориентированного подхода к разработке программного обеспечения			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к тестированию	Тестирование	ДПК-004-3.2
2.2 Python как объектно-ориентированный язык программирования. Создание классов и объектов. Реализация инкапсуляции средствами языка программирования Python			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ДПК-004-3.1, ДПК-004-3.2
2.3 Реализация наследования и полиморфизма средствами языка программирования Python			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ДПК-004-3.1, ДПК-004-3.2
Итого по разделу			6	12			
3. 3. Построение отказоустойчивых приложений. Исключения. Тестирование кода в Python							
3.1 Принципы построения отказоустойчивых приложений. Исключения в языке программирования Python. Генерирование и перехват исключений			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1

3.2 Общие принципы тестирования программного обеспечения. Тестирование кода на Python: библиотеки unittest, pytest.			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ДПК-004-3.1
Итого по разделу			4	8			
4. 4. Функциональное программирование в Python							
4.1 Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
4.2 Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, оператор yield.			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
4.3 Функциональное программирование в Python: функции как объект, кастомизированные функции, замыкания, функторы.			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
Итого по разделу			6	12			
5. 5. Концепция MVC как способ построения веб-приложений							

5.1 Протокол HTTP: назначение, принцип работы, виды HTTP-запросов. API как средство интеграции приложений. Работа с протоколом HTTP и API средствами языка программирования Python (библиотека requests)			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
5.2 Основные принципы концепции MVC. Использование концепции MVC для построения веб-приложений. Обзор фреймворков для построения веб-приложений			2	4	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
5.3 Обзор основных инструментов для разработки веб-приложений: HTML, CSS, JavaScript, библиотека jQuery. Подход AJAX			4	8	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Подготовка к аудиторной контрольной работе	Аудиторная контрольная работа	ДПК-004-3.1, ДПК-004-3.2
5.4 Разработка веб-приложения с использованием микрофреймворка Flask			6	8	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ДПК-004-3.1, ДПК-004-3.2
5.5 Понятие веб-сервиса. Построение веб-сервиса архитектуры REST на языке программирования Python			4	8	Самостоятельное изучение учебной и научно литературы, работа с материалами образовательного портала и ЭБС. Выполнение ИДЗ	Индивидуальное домашнее задание	ДПК-004-3.1, ДПК-004-3.2

Итого по разделу				18	32			
6. 6. Подготовка к зачету								
6.1	Подготовка к промежуточной аттестации				3,9	Изучение учебной и научной литературы, работа с материалами образовательного портала и ЭБС	зачет	ДПК-004-3.1, ДПК-004-3.2
Итого по разделу					3,9			
Итого за семестр				36	71,9		зачёт	
Итого по дисциплине				36	71,9		зачет	

5 Образовательные технологии

Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Основы программирования на Python» используются традиционная и модульно-компетентностная технологии.

Реализация компетентностного подхода предусматривает использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

При проведении учебных занятий преподаватель обеспечивает развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств посредством проведения интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализа ситуаций, учета особенностей профессиональной деятельности выпускников и потребностей работодателей.

Формы учебных занятий с использованием традиционных технологий:

- обзорные лекции – для рассмотрения общих вопросов информатики и информационных технологий, для систематизации и закрепления знаний;
- информационные – для ознакомления с техническими средствами реализации информационных процессов, со стандартами организации сетей, основными приемами защиты информации, и другой справочной информацией;
- лекции-визуализации – для наглядного представления способов решения алгоритмических и функциональных задач, визуализации результатов решения задач;
- семинар – практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

Формы учебных занятий с использованием технологий проблемного обучения:

- проблемная лекция – изложение материала, предполагающее постановку проблемных и дискуссионных вопросов, освещение различных научных подходов, авторские комментарии, связанные с различными моделями интерпретации изучаемого материала (для развития исследовательских навыков и изучения способов решения задач);
- лекции с заранее запланированными ошибками – направленные на поиск обучающимися синтаксических и алгоритмических ошибок при решении алгоритмических и функциональных задач, с последующей диагностикой слушателей и разбором сделанных ошибок;
- практическое занятие в форме практикума – организация учебной работы, направленная на решение комплексной учебно-познавательной задачи, требующей от обучающегося применения как научно-теоретических знаний, так и практических навыков;
- практическое занятие на основе кейс-метода – обучение в контексте моделируемой ситуации, воспроизводящей реальные условия научной, производственной, общественной деятельности; обучающиеся должны проанализировать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них; кейсы базируются на реальном фактическом материале или же приближены к реальной ситуации;
- подготовка тематических рефератов, содержащих разделы, частично или полностью выносимые на самостоятельное изучение.

Формы учебных занятий с использованием игровых технологий:

- учебная игра – форма воссоздания предметного и социального содержания будущей профессиональной деятельности специалиста, моделирования таких систем отношений, которые характерны для этой деятельности как целого;

– деловая игра – моделирование различных ситуаций, связанных с выработкой и принятием совместных решений, обсуждением вопросов в режиме «мозгового штурма», реконструкцией функционального взаимодействия в коллективе и т.п.

Технологии проектного обучения:

– творческий проект – учебно-познавательная деятельность обучающихся осуществляется в рамках рамочного задания, подчиняясь логике и интересам участников проекта, жанру конечного результата (газета, фильм, праздник, издание, экскурсия, подготовка заданий конкурсов и т.п.);

– информационный проект – учебно-познавательная деятельность с ярко выраженной эвристической направленностью (поиск, отбор и систематизация информации о каком-то объекте, ознакомление участников проекта с этой информацией, ее анализ и обобщение для презентации более широкой аудитории).

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва : Издательство Юрайт, 2023. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511891> (дата обращения: 02.05.2023).

2. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2022. - 242 с. - ISBN 978-5-9275-4108-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2057604> (дата обращения: 02.05.2023).

3. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем : учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 432 с. — (Высшее образование). — ISBN 978-5-534-07604-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513067> (дата обращения: 02.05.2023).

4. Маркин, А. В. Программирование на SQL в 2 ч. Часть 1 : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 429 с. — (Высшее образование). — ISBN 978-5-534-15817-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/509818> (дата обращения: 02.05.2023).

5. Маркин, А. В. Программирование на SQL в 2 ч. Часть 2 : учебник и практикум для вузов / А. В. Маркин. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 385 с. — (Высшее образование). — ISBN 978-5-534-15818-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/509819> (дата обращения: 02.05.2023).

6. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 4-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 214 с. — (Высшее образование). — ISBN 978-5-534-15733-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515076> (дата обращения: 02.05.2023).

7. Черткова, Е. А. Программная инженерия. Визуальное моделирование

программных систем : учебник для вузов / Е. А. Черткова. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2023. — 147 с. — (Высшее образование). — ISBN 978-5-534-09172-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513696> (дата обращения: 02.05.2023).

8. 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: 02.05.2023).

б) Дополнительная литература:

1. Гагарина, Л. Г. Технология разработки программного обеспечения : учебное пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0707-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1971872> (дата обращения: 02.05.2023).

2. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2023. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-017142-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1913856> (дата обращения: 02.05.2023).

3. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2023. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1915716> (дата обращения: 02.05.2023).

4. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 155 с. — (Высшее образование). — ISBN 978-5-534-00850-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512425> (дата обращения: 02.05.2023).

5. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512894> (дата обращения: 02.05.2023).

6. Нестеров, С. А. Базы данных : учебник и практикум для вузов / С. А. Нестеров. — Москва : Издательство Юрайт, 2023. — 230 с. — (Высшее образование). — ISBN 978-5-534-00874-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/511650> (дата обращения: 02.05.2023).

7. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2023. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519949> (дата обращения: 02.05.2023).

в) Методические указания:

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое ПО	бессрочно
Adobe Dreamweaver CS 5 Academic Edition	К-113-11 от 11.04.2011	бессрочно
Anaconda Python	свободно распространяемое ПО	бессрочно
MS SQL Server Management Studio	свободно распространяемое ПО	бессрочно
MS Visual Studio 2017 Community Edition	свободно распространяемое ПО	бессрочно
Adobe Reader	свободно распространяемое ПО	бессрочно
Браузер Mozilla Firefox	свободно распространяемое ПО	бессрочно
Браузер Yandex	свободно распространяемое ПО	бессрочно
MS Office 2003 Professional	№ 135 от 17.09.2007	бессрочно
Git	свободно распространяемое ПО	бессрочно
MS Visual Studio Code	свободно распространяемое ПО	бессрочно
JetBrains PyCharm Community Edition	свободно распространяемое ПО	бессрочно
NotePad++	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/
Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

1. учебные аудитории для проведения занятий лекционного типа:
 - мультимедийные средства хранения, передачи и представления информации;
2. учебные аудитории для проведения практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации:
 - персональные компьютеры с ПО, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета;
 - мультимедийные средства хранения, передачи и представления информации;
 - комплекс тестовых заданий для проведения промежуточных и рубежных контролей;
3. помещения для самостоятельной работы обучающихся с выходом в Интернет и с доступом в электронную информационно-образовательную среду университета;
4. помещения для хранения и профилактического обслуживания учебного оборудования:
 - шкафы для хранения учебно-методической документации, учебного оборудования и учебно-наглядных пособий.

Учебно-методическое обеспечение самостоятельной работы обучающихся

Аудиторная самостоятельная работа обучающихся на практических занятиях осуществляется под контролем преподавателя в виде решения задач и выполнения упражнений, которые определяет преподаватель для обучающегося с использованием методов ИТ.

Внеаудиторная самостоятельная работа обучающихся осуществляется в виде чтения литературы по соответствующему разделу с проработкой материала и выполнения домашних заданий с консультациями преподавателя, а так же с применением кейс-технологий.

1. Контрольные вопросы и задания для проведения текущего контроля

1. Основные парадигмы программирования.
2. Способы организации кода программного обеспечения. Стандарты кодирования.
3. Особенности языка программирования Python. Парадигмы программирования, поддерживаемые языком программирования Python. Организация кода в модули и пакеты в языке программирования Python
4. Объектно-ориентированное программирование: основные принципы, сфера применения.
5. Создание классов и объектов, реализация инкапсуляции средствами языка программирования Python.
6. Реализация наследования и полиморфизма средствами языка программирования Python.
7. Принципы построения отказоустойчивых приложений.
8. Исключения в Python. Генерирование и перехват исключений.
9. Общие принципы тестирования программного обеспечения.
10. Тестирование кода на Python: библиотеки unittest, pytest.
11. Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.
12. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.
13. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, оператор yield.
14. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов. Работа с протоколом HTTP средствами языка программирования Python
15. API как средство интеграции приложений. Работа с протоколом с API средствами языка программирования Python
16. Основные принципы концепции MVC. Использование концепции MVC для построения веб-приложений. Привести примеры.
17. Фреймворки для разработки веб-приложений. Привести примеры.
18. Микрореймворк Flask: особенности, принципы работы, основные этапы создания веб-приложения
19. Фреймворк Flask: HTML-шаблоны (templates), подключение и использование static-файлов (JS, CSS), работа с формами
20. Микрофреймворк Flask: технология объектно-реляционного отображения, работа с БД и моделями (models)
21. Микрофреймворк Flask: механизмы сессий, cookie, авторизация и права пользователей
22. Понятие веб-сервиса. Построение веб-сервисов с использованием архитектуры REST.
23. Микрофреймворк Flask: проектирование RESTful API.

2. Примеры индивидуальных домашних заданий

1. Используя микрофреймворк Flask и его расширение flask-restful, создайте RESTful API с базовой CRUD-функциональностью для работы с заказами и клиентское приложение в виде чат-бота. Для заказов необходимо хранить следующую информацию: дата, номер, клиент, общая сумма заказа, статус, список товаров в заказе. Для каждого товара указывается артикул, наименование, количество и цена за единицу. Заказы могут оплачиваться по частям. Клиентами могут быть физические и юридические лица.

2. Используя фреймворк Flask, создайте веб-приложение для отгадывания загадок. Предусмотрите 2 вида пользователей: у администратора будет возможность загружать загадки, а у пользователей – возможность их отгадывать.

3. Примеры аудиторных контрольных работ:

1. Создайте приложение, которая будет выводить на экран полный адрес и координаты достопримечательности г. Магнитогорска по выбору пользователя.

2. Используя библиотеку requests языка Python, получите данные в json формате от ресурса <https://jsonplaceholder.typicode.com/todos> и определите, какие пользователи выполнили наибольшее число задач. Результат представьте в формате: users 5 and 10 completed 12 TODOs.

Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Оценочные средства
<p>ДПК-004-3: Способен создавать программы на языках общего назначения, применять методы и <i>инструментальные</i> средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ</p> <ul style="list-style-type: none"> – ДПК-004-3.1: Проводит комплексное тестирование и отладку программных систем – ДПК-004-3.2: Проектирует программное обеспечение с использованием средств автоматизации 	
<p>ДПК-004-3.1: Проводит комплексное тестирование и отладку программных систем</p>	<p>Примеры практических заданий для зачета:</p> <ol style="list-style-type: none"> 1. Реализуйте на языке Python функцию <code>time2min(h, m)</code>, которая переводит часы и минуты в минуты с начала суток (00:00). Проведите тестирование функции, используя библиотеки <code>unittest</code> или <code>pytest</code>. 2. Используя язык Python, реализуйте генератор <code>factorials(n)</code>, генерирующий последовательность факториалов натуральных чисел. Проведите тестирование функции, используя библиотеки <code>unittest</code> или <code>pytest</code>. <p>Теоретические вопросы к зачету:</p> <ol style="list-style-type: none"> 1. Принципы построения отказоустойчивых приложений. 2. Исключения в Python. Генерирование и перехват исключений. 3. Общие принципы тестирования программного обеспечения.
<p>ДПК-004-3.2: Проектирует программное обеспечение с использованием средств автоматизации</p>	<p>Примеры практических заданий для зачета:</p> <ol style="list-style-type: none"> 1. На языке программирования Python объявите класс Автомобиль, содержащий не менее 4-х полей и 2-х методов. Создайте список из объектов класса Автомобиль. 2. Реализуйте сложение и скалярное умножение векторов на языке Python, используя функции высшего порядка <code>zip</code>, <code>map</code>, <code>reduce</code> и др. Если длина векторов не совпадает, должно генерироваться исключение. 3. Используя язык Python, реализуйте генератор <code>factorials(n)</code>, генерирующий последовательность случайных чисел без повторяющихся цифр. 4. Используя механизм шаблонов фреймворка Flask, создайте веб-сайт для научного общества. На главной странице разместите небольшой текст и ссылку на страницу со статьями. Для генерации страницы со списком статей используйте циклы в шаблонах, добавьте ссылку на главную страницу. Для оформления предусмотрите простейшие стили. <p>Теоретические вопросы к зачету:</p> <ol style="list-style-type: none"> 1. Основные парадигмы программирования. 2. Способы организации кода программного обеспечения. Стандарты кодирования. 3. Особенности языка программирования Python. Парадигмы программирования, поддерживаемые языком программирования Python. Организация кода в модули и пакеты в языке программирования Python 4. Объектно-ориентированное программирование: основные принципы, сфера применения. 5. Создание классов и объектов, реализация инкапсуляции средствами языка программирования Python. 6. Реализация наследования и полиморфизма средствами языка программирования Python.

Структурный элемент компетенции	Оценочные средства
	<p>7. Функциональное программирование в Python: списковые включения, встроенные функции высших порядков, лямбда-функции.</p> <p>8. Функциональное программирование в Python: функции как объект, декораторы функций, замыкания, функторы.</p> <p>9. Функциональное программирование в Python: итераторы и итерируемые объекты, функции-генераторы, оператор yield.</p> <p>10. Протокол HTTP: назначение, принцип работы, виды HTTP-запросов. Работа с протоколом HTTP средствами языка программирования Python</p> <p>11. API как средство интеграции приложений. Работа с протоколом с API средствами языка программирования Python</p> <p>12. Основные принципы концепции MVC. Использование концепции MVC для построения веб-приложений. Привести примеры.</p> <p>13. Фреймворки для разработки веб-приложений. Привести примеры.</p> <p>14. Микрорејворк Flask: особенности, принципы работы, основные этапы создания веб-приложения</p> <p>15. Фреймворк Flask: HTML-шаблоны (templates), подключение и использование static-файлов (JS, CSS), работа с формами</p> <p>16. Микрофреймворк Flask: технология объектно-реляционного отображения, работа с БД и моделями (models)</p> <p>17. Микрофреймворк Flask: механизмы сессий, cookie, авторизация и права пользователей</p> <p>18. Понятие веб-сервиса. Построение веб-сервисов с использованием архитектуры REST.</p> <p>19. Микрофреймворк Flask: проектирование RESTful API.</p>

б) Примерный перечень индивидуальных заданий:

1. Разработка утилиты для групповой обработки файлов на языке программирования Python.
2. Разработка утилиты для работы с изображениями с использованием библиотек NumPy, SciPy.
3. Разработка утилиты для работы с аудиозаписями на языке программирования Python.
4. Разработка утилиты командной строки Site Connectivity Checker для проверки доступности сайта на языке программирования Python.
5. Разработка клиент-серверного приложения для приема численных метрик – характеристик ПК, используя возможности языка программирования Python.
6. Разработка клиент-серверного приложения для выполнения арифметических действия над двоичными или шестнадцатеричными числами, используя возможности языка программирования Python.
7. Разработка клиент-серверной системы для сжатия текстовой информации на русском языке и последующей распаковки, используя возможности языка программирования Python.
8. Создание инструмента обнаружения хостов UDP, используя возможности языка программирования Python.
9. Создание сниффера для захвата учетных данных SMTP, POP3, IMAP, используя возможности языка программирования Python.
10. Разработка генератора поддельных IP-пакетов для проведения smurf-атак, используя возможности языка программирования Python.
11. Разработка REST-сервиса и веб-клиента для ведения заметок, используя возможности языка программирования Python.

12. Разработка REST-сервиса и веб-клиента для поиска в тексте на основе регулярных выражений, используя возможности языка программирования Python.

13. Разработка REST-сервиса и веб-клиента для учета и отслеживания ежедневных расходов, используя возможности языка программирования Python.

14. Разработка REST-сервиса и веб-клиента для аптечного склада, используя возможности языка программирования Python.

15. Разработка виртуального помощника (чат-бота или голосового помощника) для проведения тестирования в сфере информационной безопасности, используя возможности языка программирования Python.

16. Разработка виртуального помощника (чат-бота или голосового помощника) для подбора вакансий в сфере информационной безопасности, используя возможности языка программирования Python.

17. Разработка виртуального помощника (чат-бота или голосового помощника) для сети аптек, используя возможности языка программирования Python.

в) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Показатели и критерии оценивания зачета:

– **«зачтено»** – обучающийся должен показать пороговый уровень знаний на уровне воспроизведения и объяснения информации, навыки решения типовых задач;

– **«не зачтено»** – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.