

Федеральное государственное образовательное бюджетное учреждение
высшего образования

**«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)**

**Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных**

Документ подписан усиленной неквалифицированной электронной подписью
Организация: Финансовый университет при Правительстве РФ
Утверждено: Проректор по учебной и методической работе Е.А. Каменева
Сертификат: MN1+VoeFx+hZHe1RX+jjwg8JtbqvIUti
Дата: 25.11.2025 г.

Е.П. Догадина

Основы программирования

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки:

09.03.03 - Прикладная информатика,

Образовательная программа «Прикладные информационные системы в экономике
и финансах»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 03 от 16.12.2025 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 12 от 03.12.2025 г.)*

© Москва 2025

СОДЕРЖАНИЕ

1.	Наименование дисциплины	3
2.	Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3.	Место дисциплины в структуре образовательной программы	6
4.	Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	6
5.	Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	7
5.1.	Содержание дисциплины	7
5.2.	Учебно-тематический план	8
5.3.	Содержание семинаров	9
6.	Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	11
6.1.	Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	11
6.2.	Перечень вопросов, заданий, тем для подготовки к текущему контролю	12
7.	Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	14
8.	Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	23
9.	Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	24
10.	Методические указания для обучающихся по освоению дисциплины	25
11.	Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем	26
12.	Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	27

1. Наименование дисциплины

«Основы программирования».

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПKN-2	Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования	Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки	знать: основные принципы программирования, синтаксис и семантику, основные модули и пакеты стандартной библиотеки уметь: разрабатывать алгоритмы для решения типовых задач, находить и применять нужные функции и классы из стандартной библиотеки
		Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)	знать: основные функции и возможности интегрированных сред разработки (IDE), таких как Visual Studio, PyCharm и др., принципы работы с системами контроля версий (например, Git) и платформами (например, GitHub, GitLab) уметь: настраивать среды разработки под свои нужды (плагины, настройки компиляции и отладки), использовать инструменты отладки, профилирования и тестирования в IDE
		Организовывает кодовую базу, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода	знать: основы структурирования проекта, конвенции и соглашения по оформлению кода, принципы оформления документации для кода уметь: организовывать и структурировать код; применять общепринятые стили кодирования и оформлять код согласно стандартам; читать и понимать код, написанный другими разработчиками, и вносить изменения без нарушения логики работы

		Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения	<p>знать: принципы проектирования пользовательских интерфейсов (UI) и пользовательского опыта (UX); основные принципы создания графических интерфейсов (GUI) на python</p> <p>уметь: проектировать интерфейсы, основываясь на требованиях и назначении программной системы; проводить тестирование удобства использования и анализировать полученные результаты для улучшения интерфейса</p>
ПKN-3	Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней	Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов	<p>знать: различные типы структур данных; как и когда использовать каждую из этих структур данных в зависимости от требований задачи</p> <p>уметь: реализовывать основные структуры данных на выбранном языке программирования и использовать их в проектах; создавать собственные структуры данных при необходимости для решения специфических задач</p>
		Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе	<p>знать: основные виды функциональных требований; понимать нефункциональные требования, включая производительность, безопасность, доступность, масштабируемость и удобство использования</p> <p>уметь: эффективно собирать функциональные и нефункциональные требования от различных заинтересованных сторон; формулировать требования так, чтобы они были ясными, конкретными и проверяемыми</p>
		Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии	<p>знать: синтаксис и семантику языка программирования, используемого для реализации проекта; особенности платформы или технологии, на которой будет разрабатываться система; способы проектирования интерфейсов классов для обеспечения удобства использования и гибкости</p> <p>уметь: проектировать и</p>

			<p>реализовывать классы и объекты, инкапсулируя данные и методы; разбивать сложные задачи на более простые, определяя подзадачи и их взаимосвязи</p>
--	--	--	--

3. Место дисциплины в структуре образовательной программы

Дисциплина «Основы программирования» относится к «Циклу математики и информатики».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 2 (в часах)
Общая трудоёмкость дисциплины	3/108	108
Контактная работа- Аудиторные занятия	58	58
Лекции	8	8
Семинары, практические занятия	50	50
Самостоятельная работа	50	50
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет

5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Основы программирования в современном мире.

Значение и роль программирования в современном мире. История языков программирования и их эволюция. Программные системы и приложения. Введение в графические пользовательские интерфейсы (GUI) в Python. Работа с графикой и мультимедийными элементами в Python. Разработка веб-интерфейсов с использованием Flask и Django. Современные тренды в UI/UX дизайне История и назначение платформы 1С. Обзор архитектуры 1С:Предприятие.

Тема 2. Графические интерфейсы. Обзор библиотек для создания GUI.

Понимание графических интерфейсов: основные концепции и принципы. Обзор библиотек для создания GUI: Tkinter, PyQt, pygame.

Тема 3. Введение в системы контроля версий.

Основные концепции Git. Основные команды: init, clone, add, commit, push, pull. Работа с удалёнными репозиториями на GitHub.

Тема 4. Пользовательские интерфейсы в 1С.

Основные компоненты и архитектура 1С. Обзор возможностей для автоматизации бизнес-процессов. Работа с данными в 1С. Проектирование форм и элементов интерфейса. Интеграция 1С с внешними системами.

Тема 5. Введение в веб-разработку. Основные концепции HTML, CSS и JavaScript.

Работа с тегами, атрибутами и элементами. Создание структурированной разметки. Селекторы, свойства и значения. Основы каскадирования и наследования стилей. Основные концепции и синтаксис JavaScript. Работа с переменными и типами данных. Создание интерактивных элементов на веб-странице.

5.2. Учебно-тематический план

№ п/п	Наименование тем(разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоя тельная работа
			Общая, в т.ч.:	Лекции	Семинары, практическ ие занятия	
1	Основы программирования в современном мире.	21	11	1	10	10
2	Графические интерфейсы. Обзор библиотек для создания GUI.	21	11	1	10	10
3	Введение в системы контроля версий.	22	12	2	10	10
4	Пользовательские интерфейсы в 1С.	22	12	2	10	10
5	Введение в веб-разработку. Основные концепции HTML, CSS и JavaScript.	22	12	2	10	10
	Итого	108	58	8	50	50

5.3. Содержание семинаров

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Основы программирования в современном мире.	Что такое переменные в Python и как их правильно использовать. Как работают условные операторы и циклы в Python. Что такое функции в Python и как их использовать для организации кода. Как обрабатывать исключения в Python и почему это важно. Что такое библиотеки и модули в Python, и как они помогают программистам.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Графические интерфейсы. Обзор библиотек для создания GUI.	Знакомство с созданием графического интерфейса. Обзор библиотек для создания GUI: Tkinter, PyQt, Kivy, pygame. Основы работы с Tkinter. Создание простого окна и работа с основными виджетами. Обработка событий и взаимодействие с пользователем. Компонировка интерфейса с Tkinter: использование менеджеров размещения: pack, grid, place; создание сложных макетов и структур интерфейса. Работа с PyQt. Создание приложения с использованием Qt Designer. Создание форм и валидация данных. Реализация валидации пользовательского ввода. Работа с графикой и анимацией в Python. Использование библиотеки Pillow для работы с изображениями. Основы анимации в графических интерфейсах. Рисование в pygame. Окно. Поверхность. Анимация. Обработка событий клавиатуры и мыши в pygame. Спрайты. Группы спрайтов. Коллизия между объектами в pygame.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Введение в системы контроля версий.	Что такое система контроля версий и зачем она нужна. Основные концепции Git. Основы работы с Git: установка и настройка Git. , основные команды: init, clone, add, commit, push, pull. Работа с ветками в Git: создание и управление ветками, слияние веток и разрешение конфликтов. Работа с удалёнными репозиториями на GitHub. Создание и настройка репозитория на GitHub. Основы работы с удалёнными ветками. Использование Git для командной работы. Основы совместной работы с использованием Git. Практика разрешения конфликтов в команде. Git для ведения документации и управления проектами. Использование GitHub Pages для размещения	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

	документации. Обзор GitHub Actions: введение в CI/CD и автоматизацию процессов с помощью GitHub Actions. Примеры создания простых workflows.	
Пользовательские интерфейсы в 1С.	История и назначение платформы 1С. Обзор архитектуры 1С:Предприятие. Компоненты системы 1С (конфигурации, подсистемы и объекты). Обзор возможностей для автоматизации бизнес-процессов. Создание базового приложения на 1С. Основы проектирования конфигураций. Создание форм и обработок. Работа с данными в 1С. Основы работы с таблицами и регистрами. Использование запросов для извлечения данных. Разработка пользовательских интерфейсов в 1С. Проектирование форм и элементов интерфейса. Настройка интерфейса для удобства пользователей. Интеграция 1С с внешними системами. Методы обмена данными с другими приложениями. Отладка и тестирование приложений на 1С. Практика создания тестов в 1С.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).
Введение в веб-разработку. Основные концепции HTML, CSS и JavaScript.	Структура веб-страницы и ее элементы. Основы HTML. Работа с тегами, атрибутами и элементами. Создание структурированной разметки. Основы CSS. Селекторы, свойства и значения. Основы каскадирования и наследования стилей. Введение в JavaScript. Основные концепции и синтаксис JavaScript. Работа с переменными, типами данных и операторами.	Индивидуальное выполнение заданий, групповой разбор результатов выполнения заданий (не менее 30% времени на интерактивные технологии).

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Основы программирования в современном мире.	Установка и настройка среды разработки. Работа в интерактивном режиме интерпретатора. Интерактивная оболочка Ipython notebook: принципы работы и применения. Работа с платформой 1С:Предприятие.	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Графические интерфейсы. Обзор библиотек для создания GUI.	Основы юнит-тестирования для графических интерфейсов. Инструменты для автоматизированного тестирования.	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Введение в системы контроля версий.	Использование Git для командной работы. Основы совместной работы с использованием Git. Практика разрешения конфликтов в команде.	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Пользовательские интерфейсы в 1С.	Интеграция 1С с внешними системами. Методы обмена данными с другими приложениями. Использование API для интеграции.	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.
Введение в веб-разработку. Основные концепции HTML, CSS и JavaScript.	Адаптивный веб-дизайн. Основы медиазапросов и отзывчивого дизайна. Примеры адаптации интерфейса под разные устройства. Введение в AJAX и Fetch API. Основы асинхронного программирования в JavaScript. Получение данных с сервера без перезагрузки страницы.	Индивидуальное выполнение заданий с использованием соответствующего программного обеспечения.

6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания контрольной работы

Задания для выполнения в 1С:

1. Создание справочника. Создайте справочник "Клиенты" с полями: "Наименование", "Адрес", "Телефон". Реализуйте возможность добавления, редактирования и удаления записей в справочнике.
2. Разработка формы для ввода данных. Создайте форму для ввода данных о товарах с полями: "Наименование", "Цена", "Количество". Реализуйте кнопку для сохранения данных в соответствующий справочник.
3. Вывод списка клиентов. Реализуйте обработку, которая выводит список всех клиентов из справочника "Клиенты" в виде таблицы. Добавьте возможность фильтрации по наименованию клиента.
4. Создание и обработка документа. Создайте документ "Заказ на продажу" с полями: "Клиент", "Дата заказа", "Товары". Реализуйте логику для сохранения данных документа.
5. Работа с массивами. Напишите процедуру, которая принимает массив чисел и возвращает сумму всех элементов массива. Затем протестируйте эту процедуру на массиве, содержащем 10 случайных чисел.
6. Обработка ошибок. Реализуйте обработку ошибок в коде, который получает данные из справочника "Клиенты". Если клиент не найден, выводите сообщение об этом пользователю.
7. Создание функции для расчета скидки. Напишите функцию, которая принимает цену товара и процент скидки, а затем возвращает окончательную цену после применения скидки. Проверьте функцию на нескольких примерах.

Задания для выполнения с GUI интерфейсами:

1. Создайте приложение для конвертации различных единиц измерения (например, длины, веса, температуры). Интерфейс должен включать выбор единиц для конвертации (например, метры в километры, градусы Цельсия в Фаренгейты и т.д.). Создайте текстовые поля для ввода значения и отображения

результата. Реализуйте кнопку для выполнения конверсии. Обеспечьте обработку некорректного ввода (например, ввод букв вместо чисел).

2. Разработайте приложение для генерации случайных паролей с заданной длиной и настройками. Интерфейс должен включать ползунок для выбора длины пароля и чекбоксы для включения/исключения специальных символов, цифр и заглавных букв. Создайте кнопку для генерации пароля. Отображайте сгенерированный пароль в текстовом поле, с возможностью его копирования в буфер обмена.

Задания для выполнения в современных системах контроля версий:

1. Представьте, что вы начинаете большой раздел работы. Для изоляции изменений создайте новую ветку. Назовите ее, чтобы было понятно, что вы в ней будете делать.

2. Перейдите в новую ветку и сделайте несколько коммитов.

3. Перейдите в основную ветку и обратите внимание на состояние рабочей директории.

4. Создайте еще одну ветку для работы над другим направлением в вашей программе. Обычно так работают в команде, каждый участник в собственной ветке. Либо в ветках может идти параллельная работа над разными возможностями программы. В таком случае эти ветки называются тематическими.

5. Сделайте несколько коммитов во вновь созданную ветку.

6. Перейдите в основную ветку и слейте в нее первую тематическую ветку.

7. Слейте в основную ветку вторую тематическую. Если возникли конфликты слияния, разрешите их и завершите слияние.

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. *Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине.*

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

ПКН-2 Способность разрабатывать алгоритмы и программы с использованием современных технологий программирования

1) Владеет объектно-ориентированным языком программирования на уровне знания синтаксиса и семантики, основ стандартной библиотеки

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные принципы программирования, синтаксис и семантику, основные модули и пакеты стандартной библиотеки

Уметь: разрабатывать алгоритмы для решения типовых задач, находить и применять нужные функции и классы из стандартной библиотеки

Типовые контрольные задания

1. Создание справочника. Создайте справочник "Клиенты" с полями: "Наименование", "Адрес", "Телефон". Реализуйте возможность добавления, редактирования и удаления записей в справочнике.

2. Из списка списков элементами которого являются текстовые символы собрать строку, в которой вложенные списки объединены в слова, а слова через запятую объединены в строку. Пример список вида [['E', 'e', 'n', 'y'], ['m', 'e', 'e', 'n', 'y'], ['m', 'i', 'n', 'e', 'y'], ['m', 'o', 'e']] будет преобразован в строку 'Еену, теену, mineу, мое'.

2) Использует инструментальные средства программирования (IDE, SDK, API, популярные фреймворки и библиотеки)

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные функции и возможности интегрированных сред разработки (IDE), таких как Visual Studio, PyCharm и др., принципы работы с системами контроля версий (например, Git) и платформами (например, GitHub, GitLab)

Уметь: настраивать среды разработки под свои нужды (плагины, настройки компиляции и отладки), использовать инструменты отладки, профилирования и тестирования в IDE

Типовые контрольные задания

1. Создание репозитория:

- Создайте новый репозиторий на GitHub с названием "MyProject".
- Добавьте описание проекта и выберите лицензию (например, MIT или Apache 2.0).
- Инициализируйте репозиторий с файлом README.md, в котором кратко опишите цели вашего проекта.

2. Работа с Git:

- Клонировать репозиторий на локальную машину.
- Создайте новую ветку с названием "feature/add-functionality", где вы будете работать над новой функциональностью.
- Добавьте в проект несколько файлов (например, index.html, script.js, styles.css) и внесите изменения в них.
- Используйте команды Git для добавления изменений в индекс, коммита с описанием и пуша ветки на GitHub.

3) Организует кодирование, ориентируется в существующем коде, демонстрирует знание общепринятых соглашений и политик в области оформления кода

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основы структурирования проекта, конвенции и соглашения по оформлению кода, принципы оформления документации для кода

Уметь: организовывать и структурировать код; применять общепринятые стили кодирования и оформлять код согласно стандартам; читать и понимать код, написанный другими разработчиками, и вносить изменения без нарушения логики работы

Типовые контрольные задания

Предоставлен фрагмент кода (можно использовать любой открытый проект на GitHub или создать свой). Этот код написан с нарушениями стиля и плохой структурой. Выполните анализ кода и подготовьте отчет, в котором:

- Определите нарушения стиля кодирования.
- Укажите места, где код можно улучшить (например, рефакторинг, добавление комментариев).
- Приведите примеры исправленного кода.

4) Проектирует текстовый, программный или графический интерфейс программной системы исходя из ее назначения

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: принципы проектирования пользовательских интерфейсов (UI) и пользовательского опыта (UX); основные принципы создания графических интерфейсов (GUI) на python

Уметь: проектировать интерфейсы, основываясь на требованиях и назначении программной системы; проводить тестирование удобства использования и анализировать полученные результаты для улучшения интерфейса

Типовые контрольные задания

Реализуйте графический интерфейс, который позволяет пользователю добавлять, редактировать и удалять заметки. Заметки должны сохраняться в текстовом файле. Требования:

- Используйте Tkinter для создания интерфейса.
- Интерфейс должен включать текстовое поле для ввода заметки, кнопки для добавления, редактирования и удаления заметок.
- Добавьте возможность загрузки заметок из файла при запуске приложения.

ПКН-3 Способность проектировать и реализовывать архитектуру и дизайн программной системы в соответствии с анализом задачи и требований к ней

1) Демонстрирует знание основных алгоритмов и структур данных, использует на практике простые структуры данных, оценивает сложность алгоритмов

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: различные типы структур данных; как и когда использовать каждую из этих структур данных в зависимости от требований задачи

Уметь: реализовывать основные структуры данных на выбранном языке программирования и использовать их в проектах; создавать собственные структуры данных при необходимости для решения специфических задач

Типовые контрольные задания

Выберите одну из структур данных (например, стек, очередь, двусвязный список) и реализуйте ее на выбранном языке программирования. Реализуйте основные операции для этой структуры данных (например, добавление, удаление, получение элементов).

2) Собирает, формулирует, систематизирует и анализирует функциональные и нефункциональные требования к информационной системе, выбирает архитектурные решения на их основе

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: основные виды функциональных требований; понимать нефункциональные требования, включая производительность, безопасность, доступность, масштабируемость и удобство использования

Уметь: эффективно собирать функциональные и нефункциональные требования от различных заинтересованных сторон; формулировать требования так, чтобы они были ясными, конкретными и проверяемыми

Типовые контрольные задания

Вам необходимо разработать приложение для управления библиотекой (например, для учета книг, пользователей и выдачи книг).

- Проведите интервью с потенциальными пользователями (включая библиотекарей и читателей) для сбора требований.

Требования:

- Зафиксируйте собранные требования в документе.

- Разделите их на функциональные и нефункциональные.

3) Создает объектно-ориентированный код, инкапсулирующий условия задачи, производит декомпозицию задачи и проектирует систему в пределах одной платформы или технологии

Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции

Знать: синтаксис и семантику языка программирования, используемого для реализации проекта; особенности платформы или технологии, на которой будет разрабатываться система; способы проектирования интерфейсов классов для обеспечения удобства использования и гибкости

Уметь: проектировать и реализовывать классы и объекты, инкапсулируя данные и методы; разбивать сложные задачи на более простые, определяя подзадачи и их взаимосвязи

Типовые контрольные задания

Разработать систему управления задачами (Task Management System) для небольшой команды. Изучите требования к системе, включая функции создания, редактирования и удаления задач, а также назначение задач участникам команды.

Требования:

- Напишите краткое описание системы и ее основных функций.
- Определите основные сущности, которые будут использоваться в системе (например, Задачи, Пользователи, Проекты).
- Подготовьте список функциональных и нефункциональных требований.

Примеры практико-ориентированных заданий

1. Разработайте приложение "Книга контактов"

Описание:

Создайте приложение для хранения информации о контактах (имя, номер телефона, электронная почта).

Требования:

- Интерфейс должен включать текстовые поля для ввода информации о контакте и кнопку для добавления контакта.
- Используйте Listbox для отображения списка контактов.
- Реализуйте возможность удаления выбранного контакта из списка.
- Сохраните контакты в файл (например, CSV) и обеспечьте возможность загрузки из файла.

2. Мини-игра "Угадай число"

Описание:

Разработайте простую игру, в которой компьютер загадывает число, а пользователь пытается его угадать.

Требования:

- Создайте интерфейс с текстовым полем для ввода числа и кнопкой для отправки.
- При каждом вводе выводите сообщение о том, больше, меньше или равно загаданному числу.
- Реализуйте возможность перезапуска игры.

3. Простое приложение "Калькулятор"

Описание:

Создайте графический калькулятор, который сможет выполнять основные арифметические операции: сложение, вычитание, умножение и деление.

Требования:

- Используйте Tkinter для создания интерфейса.
- Интерфейс должен включать кнопки для цифр (0-9), операций (+, -, *, /), а также кнопки "=", "C" (очистка).
- Результаты должны отображаться в текстовом поле.
- Реализуйте обработку ошибок (например, деление на ноль).

4. Создание приложения "Учет клиентов"

Описание:

Разработайте приложение для учета клиентов и их заказов в 1С.

Требования:

- Создайте справочник "Клиенты" с полями: ФИО, Адрес, Телефон, Email.
- Создайте документ "Заказ", который будет включать клиента, дату заказа, список товаров и общую сумму.
- Реализуйте отчет, показывающий информацию о заказах по клиентам.

5. Реализация "Календаря событий"

Описание:

Создайте приложение для учета событий (встреч, собраний и т.п.) в 1С.

Требования:

- Создайте справочник "События" с полями: Название, Дата, Время, Место, Описание.
- Реализуйте возможность добавления, редактирования и удаления событий.
- Разработайте форму для отображения списка событий с возможностью фильтрации по дате.

6. Создание личной страницы-портфолио

Описание:

Создайте личную страницу-портфолио, на которой будет представлена информация о вас, ваши проекты и контактные данные.

Требования:

- Используйте HTML для структуры страницы: заголовки, параграфы, списки, ссылки и изображения.
- Примените CSS для оформления страницы: шрифты, цвета, отступы и выравнивание.
- Добавьте раздел "Проекты" с мини-описанием каждого проекта и ссылками на них.

7. Создание галереи изображений

Описание:

Создайте галерею изображений, которая будет отображать изображения в виде сетки с возможностью увеличения при клике.

Требования:

- Используйте HTML для структуры галереи (например, `<div>` и ``).
- Реализуйте стиль сетки с помощью CSS (flexbox или grid).
- Используйте JavaScript для обработки кликов на изображениях: при клике на изображение оно должно открываться в полноэкранном режиме или в модальном окне.
- Добавьте кнопку для закрытия модального окна.

8. Создание формы обратной связи

Описание:

Создайте веб-форму для обратной связи, где пользователи могут оставлять свои комментарии и контактные данные.

Требования:

- Используйте HTML для создания формы с полями: имя, email, сообщение.
- Реализуйте валидацию формы с помощью JavaScript (например, обязательные поля, правильный формат email).
- Используйте CSS для стилизации формы и обеспечения ее удобного восприятия.
- Реализуйте функционал для отображения сообщения об успешной отправке после заполнения формы.

Примерные вопросы для подготовки к зачету

1. Опишите своими словами значение следующих терминов:

- a. рабочий каталог
- b. репозиторий
- c. коммит
- d. ветка

2. Что такое удаленный репозиторий?

3. Где нужно вводить команды git?

4. Для чего нужны ветки в системах контроля веток?

5. Как возникают конфликты слияния?

6. Как разрешать конфликты слияния?

7. Зачем нужен облачный хостинг репозитория?

8. Какими основными функциями обладает сайт github.com?

9. Как организовать командную работу над открытым проектом?

10. Что такое графический пользовательский интерфейс (GUI)?
11. Какие основные элементы управления вы знаете? Приведите примеры.
12. Что такое событие в контексте GUI? Как они обрабатываются?
13. Каковы основные принципы проектирования удобного интерфейса?
14. Что такое компоновка (layout) в GUI? Какие типы компоновок вы знаете?
15. Что такое "обратная связь" в интерфейсе пользователя? Почему она важна?
16. Как вы можете улучшить доступность вашего GUI для пользователей с ограниченными возможностями?
17. Что такое модель "Модель-Представление-Контроллер" (MVC)? Как она используется в GUI?
18. Каковы основные отличия между модальными и немодальными окнами?

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Шитов, В.Н. Проектирование и разработка интерфейсов пользователя : Учебное пособие / В.Н. Шитов, К.Е. Успенский Электрон. дан. Москва : КноРус, 2025 294 с. Режим доступа: book.ru Internet access <https://book.ru/book/955527> ISBN 978-5-406-13754-3. [БИК ID: RU\bookru\bibl\955527]
2. Титов, А. Н. Введение в Tkinter. Разработка графических интерфейсов в Python [Электронный ресурс] : учебно-методическое пособие / Титов А. Н.,Тазиева Р. Ф. Казань : КНИТУ, 2023 100 с. Книга из коллекции КНИТУ - Информатика <https://e.lanbook.com/book/412466> ISBN 978-5-7882-3340-6. [БИК ID: RU-LAN-BOOK-412466]
3. GIT: Создание прочной основы для эффективной разработки / Жматов Д. В. Ч. 1 : GIT: Создание прочной основы для эффективной разработки. Часть 1 : Учебное пособие . Ч. 1 / Жматов Д. В. Москва : РТУ МИРЭА, 2024 114 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/457052> ISBN 978-5-7339-2345-1. [БИК ID: RU-LAN-BOOK-457052]

Дополнительная литература:

1. Шаталова, А.Ю. Основы веб-разработки : Учебное пособие / А.Ю. Шаталова, М.В. Коротеев Электрон. дан. Москва : КноРус, 2024 282 с. Режим доступа: book.ru Internet access <https://book.ru/book/955875> ISBN 978-5-406-13930-1. [БИК ID: RU\bookru\bibl\955875]
2. Жматов, Д. В. Технологии разработки программных приложений, основанные на использовании Git, Docker, Gradle: Практикум [Электронный ресурс] / Жматов Д. В.,Петрова А. А. Москва : РТУ МИРЭА, 2025 70 с. Книга из коллекции РТУ МИРЭА - Информатика <https://e.lanbook.com/book/504851> ISBN 978-5-7339-2592-9. [БИК ID: RU-LAN-BOOK-504851]
3. Филиппов, С. А. Основы современного веб-программирования [Электронный ресурс] : учебное пособие для вузов / Филиппов С. А. Москва : НИЯУ МИФИ, 2011 160 с. Рекомендовано УМО «Ядерные физика и технологии» в качестве учебного пособия для студентов высших учебных заведений Книга из коллекции

НИЯУ МИФИ - Информатика http://e.lanbook.com/books/element.php?pl1_id=75795
ISBN 978-5-7262-1402-3. [БИК ID: 75795]

4. Заяц, А. М. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js [Электронный ресурс] / Заяц А. М., Васильев Н. П. 3-е изд., стер. Санкт-Петербург : Лань, 2021 120 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/154380> ISBN 978-5-8114-7042-6. [БИК ID: RU-LAN-BOOK-154380]

5. Шишкин, В. В. Разработка логических компьютерных игр с графическим интерфейсом в среде Питон [Электронный ресурс] : учебное пособие для студентов направления 09.03.02 «информационные системы и технологии» / Шишкин В. В.,Афонин Д. С. Ульяновск : УлГТУ, 2023 88 с. Книга из коллекции УлГТУ - Информатика <https://e.lanbook.com/book/416348> ISBN 978-5-9795-2339-2. [БИК ID: RU-LAN-BOOK-416348]

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
2. Электронно-библиотечная система издательства "Лань" <https://e.lanbook.com/>
3. Научная электронная библиотека eLibrary.ru <http://elibrary.ru>
4. •Электронно-библиотечная система Znanium <http://www.znanium.ru/>
5. •Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/>

10. Методические указания для обучающихся по освоению дисциплины

Самостоятельная работа студента направлена на подготовку к учебным занятиям и на развитие знаний, умений и навыков, предусмотренных программой дисциплины. В соответствии с учебным планом дисциплина предусматривает лекции и практические занятия (семинары). Успешное изучение дисциплины требует посещения всех видов занятий, выполнение заданий преподавателя и ознакомления с основной и дополнительной литературой.

При подготовке к лекционным занятиям студентам необходимо: перед очередной лекцией необходимо просмотреть конспект материала предыдущей лекции. При затруднениях в восприятии материала следует обратиться к основным литературным источникам. Если разобраться в материале опять не удалось, то обратитесь к лектору (по графику его консультаций) или к преподавателю на практических занятиях.

Практические занятия завершают изучение наиболее важных тем учебной дисциплины. Они служат для закрепления изученного материала, развития умений и навыков подготовки докладов, сообщений, приобретения опыта устных публичных выступлений, ведения дискуссии, аргументации и защиты выдвигаемых положений, а также для контроля преподавателем степени подготовленности студентов по изучаемой дисциплине.

При подготовке к практическому занятию студенты имеют возможность воспользоваться консультациями преподавателя.

При подготовке к практическим занятиям студентам необходимо:

приносить с собой рекомендованную преподавателем литературу к конкретному занятию;

до очередного практического занятия по рекомендованным литературным источникам проработать теоретический материал, соответствующей темы занятия;

в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;

в ходе семинара давать конкретные, четкие ответы по существу вопросов;

на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Студентам, пропустившим занятия (независимо от причин), не имеющим письменного решения задач или не подготовившихся к данному практическому занятию, рекомендуется не позже чем в 2-недельный срок явиться на консультацию к преподавателю и отчитаться по теме, изученную на занятии.

При подготовке к контрольной работе студентам следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Python 3.8
2. 1С: Предприятие 8.3
3. Microsoft Office (Windows)
4. Текстовый редактор
5. Anaconda

Современные профессиональные базы данных и информационные справочные системы:

1. Язык программирования Python 3. <https://pythonworld.ru/>
2. Программный пакет для формирования отчетности «1-с предприятие»;
3. Информационно-правовая система «Гарант»
4. Информационно-правовая система «Консультант Плюс»

5. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>
6. Система комплексного раскрытия информации «СКРИН» - <http://www.skrin.ru/>

Сертифицированные программные и аппаратные средства защиты информации:

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Учебная аудитория** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенная оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), набор демонстрационного оборудования (проектор, экран)
2. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)
3. **Помещение для самостоятельной работы** обучающихся оснащенное компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде Университета.