

Министерство науки и высшего образования Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:  
Директор  
А. В. Замятин

Оценочные материалы по дисциплине

Разработка программного обеспечения и скриптовые языки

по направлению подготовки

**09.04.03 Прикладная информатика**

Направленность (профиль) подготовки:  
**Управление цифровой трансформацией**

Форма обучения  
**Очная**

Квалификация  
**Магистр**

Год приема  
**2025**

СОГЛАСОВАНО:  
Руководитель ОП  
Н.Л.Ерёмина

Председатель УМК  
С.П.Сущенко

Томск – 2025

## **1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами**

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-2 Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;

УК-2 Способен управлять проектом на всех этапах его жизненного цикла.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-2.1 Владеет необходимыми методами алгоритмизации и программирования для решения профессиональных задач

ИОПК-2.2 Знает современные подходы, методы применения современных интеллектуальных технологий для решения профессиональных задач

ИОПК-2.3 Использует методы современных интеллектуальных технологий для решения профессиональных задач

ИУК-2.1 Формулирует цель проекта, обосновывает его значимость и реализуемость.

ИУК-2.2 Разрабатывает программу действий по решению задач проекта с учетом имеющихся ресурсов и ограничений.

ИУК-2.3 Обеспечивает выполнение проекта в соответствии с установленными целями, сроками и затратами.

## **2. Оценочные материалы текущего контроля и критерии оценивания**

Элементы текущего контроля:

- Вопросы для устного опроса;
- Лабораторные задания;
- Кейс для командного задания в формате PBL;

**Примерные вопросы для устного опроса при текущем контроле (ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИУК-2.1, ИУК-2.2, ИУК-2.3)**

1. Какие виды типизации применяются в современных языках?
2. Как получить список всех атрибутов объекта в Python?
3. Что такое GIL в реализации CPython и как он работает?
4. Чем отличается наследование в JavaScript от наследования в Python?
5. Какие средства управления исключительными ситуациями есть в Go?
6. Как обрабатывать чтение из нескольких каналов в Go?
7. Какие механизмы приведения типов доступны в Rust?
8. Что такое типаж (trait) и как его использовать в Rust?

**Лабораторные задания (ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИУК-2.1, ИУК-2.2, ИУК-2.3)**

### **Задание 1. Wget на минималках (максимально 1 балл)**

Используя примитивы многозадачности и синхронизации реализовать скачивание файла по http с выводом количества уже принятых байтов каждую секунду.

Для реализации рекомендуется использовать библиотеку http.client.

Программа получает в качестве аргумента командной строки URL файла и скачивает его в текущую папку сохраняя исходное имя. Каждую секунду по таймеру выводится размер принятых данных.

Язык для реализации – Python (допускается другой скриптовый язык кроме JS).

### **Задание 2. Сервер времени (максимально 2 балла)**

Разработать веб-приложение (например через интерфейс wsgi), реализующее аналог сервиса time.is и предоставляющее работу с временными зонами на базе библиотеки pytz. Необходимо реализовать следующие конечные пути:

1) Веб приложение по запросу GET / отдает текущее время во временной зоне сервера в формате html

2) по запросу GET /<tz name> отдает текущее время в запрошенной зоне в формате html

3) по запросу POST /api/v1/time - отдает в формате json текущее время в зоне определенной параметром tz(если нет - то зона сервера)

4) по запросу POST /api/v1/date- отдает в формате json текущую дату в зоне определенной параметром tz(если нет - то зона сервера)

5) по запросу POST /api/v1/datediff- отдает в формате json время между датами определенными параметрами start и end (каждый - это json формата {"date": "12.20.2021 22:21:05", "tz": "EST"} или {"date": "12:30pm 2020-12-01", "tz": "Europe/Moscow"}, tz - опциональна).

Дополнительные требования к реализации:

1) Запрещено использовать веб-фреймворки

2) Код лабораторной должен представлять из себя один (!) файл

3) К решению должен прилагаться тест с использованием библиотеки requests с тестированием каждого апи

Язык для реализации – Python (допускается другой скриптовый язык кроме JS).

### **Задание 3. Мобильное приложение (только PWA) - менеджер паролей (максимально 2 балла)**

Необходимо разработать прогрессивное веб-приложение (PWA), которое возможно установить, как мобильное приложение.

Требуется реализовать минимальный комплект из манифеста, service-worker.js, иконки и html разметки с кодом (или отдельный js файл с кодом). Реализовывать отдельный сервер backend не требуется.

Приложение должно представлять из себя менеджер паролей с локальным хранением данных в localStorage(и/или в IndexedDB) и возможностью сгенерировать пароль с требуемой сложностью. Таким образом, веб приложение хранит данные в браузере пользователя, без бэкенда

При открытии пользователю должен показаться список сохраненных паролей и форма для добавления нового (логин, пароль, url).

Язык для реализации – Javascript или TypeScript

### **Задание 4: Секретный протокол (максимально 3 балла).**

Реализовать на языке Go tcp клиент и сервер (общее приложение, режим определяется параметрами), способное установить сокет соединение и последовательно обмениваться сообщениями. На каждой стороне реализуется алгоритм переменных ключей. На каждом шаге обмена вычисляется следующий ключ и сравнивается с полученным от второй стороны.

Шаг 1. Установление соединения. Клиент подключается к серверу и передает стартовую строку и первый ключ.

Шаг 2. Сервер на основе строки и ключа генерирует новый ключ и отдает его клиенту.

Шаг 3. Клиент сравнивает полученный ключ со следующим ключом, и, если все успешно, создает новый ключ и отправляет следующее сообщение на сервер.

Шаг 4..10 - аналогично

На каждом шагу приложение должно выводить в консоли текущий статус, текущий ключ и отправленное/полученное сообщение. Приложение представляет из себя аналог консольного чата и позволяет вводить сообщение и отображать ответ.

При запуске программа должна принимать два параметра командной строки:

- 1) порт - режим сервера или ip:port - режим клиента
- 2) -n 100 - кол-во одновременных подключений

Язык для реализации – Go

#### **Задание 5. Wget на минималках (максимально 2 балла)**

Используя примитивы многозадачности и синхронизации реализовать скачивание файла по http(s) с выводом количества уже принятых байтов каждую секунду.

Для реализации рекомендуется использовать раздвоения потоков данных на базе TeeReader.

Программа получает в качестве аргумента командной строки URL файла и скачивает его в текущую папку сохраняя исходное имя. Каждую секунду (по таймеру) выводится размер принятых данных

Язык для реализации – Rust

**Кейс для командного задания в формате PBL (ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИУК-2.1, ИУК-2.2, ИУК-2.3) «Как собрать спам базу за пять дней или web scraping – это просто (нет)»:**

Легенда:

В понедельник утром сисадмин Анатолий получил срочное задание от шефа, директора агентства ООО «Черный пиар». После того, как повязали их поставщика спам-баз для холодной рассылки, им срочно нужны были новые email адреса.

Решив, что это дело простое – шеф арендовал сервер на 16 vCPU с толстым каналом связи и озадачил нашего героя – добыть 10 миллионов адресов к очередной пятничной рассылке спама.

Подумав, Анатолий решил, что раз глупые пользователи оставляют адреса на всяких форумах и досках объявлений, то нужно их просто собрать. Но, даже если он перестанет спать и израсходует десяток клавиатур и мышей – вручную ему столько не найти. Так что ему ничего не остается, кроме как освежить студенческие знания языка Python и вперед, в бой!

Однако, потратив два дня на изучение регулярных выражений и получив кучу мусора вида «Ответы@Mail.ru», Анатолий осознал, что скачать и распарсить весь интернет – это долго, и нужно проверенное решение для web-scraping. Но, кроме того, стоило ему обратиться к одному сайту десяток раз – как его начинали блокировать. Нет, так в срок не успеть!

Ключевые слова:

- web-scraping – извлечение данных из сети Интернет (сайтов, не специализированных открытых источников);
- спам-база – база email адресов случайных людей, обычно по тематикам;
- холодная рассылка – отправка email произвольным людям, без согласия;
- парсинг (html страниц) – извлечение отдельных сведений из кода страницы, в т.ч. с учетом ее структуры;
- регулярные выражения – язык выражений для сопоставления текста;
- vCPU, пять дней – необходимо использовать многопоточность и/или многопроцессорность;
- блокировка – сайты блокируют подозрительные соединения, в т.ч. для защиты контента (нужны техники обхода).

Формулировка проблемы:

Прежде чем разрабатывая компьютерную программу на языке Python, предназначенную для автоматического извлечения email адресов с веб-сайтов, необходимо:

- Выделить ключевые компоненты и модули;
- Определить отдельные стадии работы программы;
- Выбрать готовую библиотеку или набор инструментов для извлечения данных из Html страниц;
- Выбрать технологию многопоточности/многопроцессорной обработки, оптимальной для 16 vCPU;
- Выбрать способ обхода блокировок со стороны целевых сайтов.

Учебные цели:

- Научить декомпозировать поставленную задачу на отдельные компоненты и разбить на этапы разработки;
- Сравнивать и выбирать готовые инструменты и библиотеки для языка Python.

### **3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания**

Оценка по дисциплине вычисляется как сумма баллов по итогам текущего контроля успеваемости по результатам выполнения лабораторных работ, представленных в электронном учебном курсе. На выбор студенту предлагаются лабораторные задания на разных языках программирования с различной максимальной стоимостью в баллах. каждая на 2 языках. При сдаче каждой лабораторной работы проверяются умения по индикаторам всех компетенций дисциплины: ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИУК-2.1, ИУК-2.2, ИУК-2.3 Итоговая оценка определяется как сумма набранных баллов по таблице:

| Оценка                | Баллы         |
|-----------------------|---------------|
| неудовлетворительно   | менее 1 балла |
| 3 (удовлетворительно) | 1 балл        |
| 4 (хорошо)            | 2 балла       |
| 5 (отлично)           | 3 балла       |

### **4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций) (ИОПК-2.1, ИОПК-2.2, ИОПК-2.3, ИУК-2.1, ИУК-2.2, ИУК-2.3)**

Вопросы для контроля остаточных знаний:

Тема 1. Классификация и применение современных языков программирования

1. Какие применяются основные подходы для трансляции/компиляции кода?

Ответ: Компиляция (преобразование в машинный код), Интерпретация (построчное выполнение), Смешанный подход (компиляция в байт-код с последующей интерпретацией)

2. Какие фазы составляют компилятор?

Ответ: Лексический анализ, Синтаксический анализ, Семантический анализ  
Генерация промежуточного кода, Оптимизация, Генерация целевого кода

3. Какие поколения ЯП существуют на данный момент?

Ответ: 1-е поколение: Машинный код, 2-е поколение: Ассемблер, 3-е поколение: Высокоуровневые языки (C, Pascal), 4-е поколение: Языки специального назначения (SQL) и схожие с естественным языком (Python), 5-е поколение: Визуальные и графические языки (Scratch, LabView)

4. Какие парадигмы считаются классическими?

Ответ: Императивная, Объектно-ориентированная, Функциональная, Логическая

5. Какие виды типизации применяются в современных языках?

Ответ: Статическая/Динамическая, Сильная/Слабая, Явная/Неявная, Структурная/Номинативная

6. Что такое синтаксический сахар и соль?

Ответ: Синтаксический сахар – это избыточные синтаксические возможности языка, упрощающие написание кода. Синтаксическая соль – конструкции, намеренно усложняющие код для предотвращения ошибок.

7. Какие проблемы необходимо решать в рамках интернационализации ЯП?

Ответ: Поддержка различных кодировок, Локализация сообщений об ошибках, Обработка различных форматов дат и чисел, Поддержка направления письма

Сортировка с учетом языковых особенностей

8. Что такое управление памятью?

Ответ: это процесс выделения и освобождения памяти в программе. Может быть Статическим, ручное управление, Автоматическое управление (сборка мусора), Освобождение памяти на основе подсчета ссылок

9. Чем отличается явное и не явное приведение типов? Какие есть ещё альтернативы?

Ответ: Явное: программист явно указывает преобразование (например, `int(x)`), Неявное: система автоматически преобразует типы. Альтернативы: умная типизация, `duck typing`, градуальная типизация

## Тема 2. Скриптовый язык Python 3

1. Как получить список всех атрибутов объекта?

Ответ: `dir(object)`

2. Как получить список всех “публичных” атрибутов объекта?

Ответ: `[attr for attr in dir(object) if not attr.startswith('_')]`

3. Как получить список методов объекта

Ответ: `[method for method in dir(object) if callable(getattr(object, method))]`

4. Есть два кортежа, как получить третий как конкатенацию первых двух?

Ответ:

5. Есть два кортежа, как получить третий как объединение уникальных элементов первых двух кортежей?

Ответ: `tuple3 = tuple1 + tuple((i for i in tuple2 if i not in tuple1))`

6. Почему если в цикле меняется список, то используется `for x in lst[:]`, что означает `[:]`?

Ответ: `[:]` создает копию списка, что предотвращает изменение списка во время итерации по нему

7. Почему операция вида `a<b=c` недопустима, а операция вида `a<b==c` – допустима?

Ответ: `a<b=c` недопустима, потому что `=` это операция присваивания, а не сравнения. `a<b==c` допустима, так как это цепочка сравнений, где `==` это операция сравнения.

8. Какие ограничения на длину строки установлены в Python?

Ответ: В Python 3 нет жестких ограничений на длину строки. Ограничение может быть только объемом доступной памяти.

9. Что такое GIL и как он работает

Ответ: GIL (Global Interpreter Lock) – это механизм в CPython, который позволяет только одному потоку выполнять Python-код в один момент времени. Он необходим для обеспечения потокобезопасности при работе со счетчиками ссылок объектов. GIL может быть освобожден при выполнении I/O операций или CPU-bound задач в расширениях на C.

### Тема 3. Скриптовый язык Javascript

1. Сколько параметров можно передать функции?

Ответ: В JavaScript функции можно передать любое количество параметров, независимо от того, сколько параметров объявлено в определении функции. Доступ к дополнительным параметрам можно получить через специальный объект `arguments` или используя `rest`-параметры (`...args`).

2. Чем отличается наследование в JavaScript от наследования в Python?

Ответ: JavaScript использует прототипное наследование и нет множественного наследования (только через миксины).

В Python используется классическое ООП наследование с поддержкой множественного наследования

3. Что представляет из себя метод объекта в JavaScript?

Ответ: Метод объекта в JavaScript – это свойство объекта, значением которого является функция

4. Зачем в JavaScript перед переменной писать `var`?

Ответ: Объявления переменной в функциональной области видимости. Однако рекомендуется использовать `let` и `const` вместо `var` т.к. они имеют блочную область видимости

5. Как создать массив в JavaScript?

Ответ: `let arr1 = [1, 2, 3];`  
`let arr2 = new Array(1, 2, 3);`

```
let arr3 = Array.from("123");
```

6. Можно ли в JavaScript использовать функцию в качестве конструктора?

Ответ: Да. При вызове функции с `new` она создает новый объект, устанавливает `this` на этот объект, возвращает объект (если явно не указан `return`)

7. Какой тип у объекта `NaN` и что вернет выражение `NaN===NaN`?

Ответ: `NaN` имеет тип `number`. `NaN === NaN` вернет `false`, потому что `NaN` не равен ничему, включая себя самого. Для проверки на `NaN` следует использовать `Number.isNaN()`.

8. Какая разница между операторами `==` и `===`?

Ответ: `==` выполняет сравнение с приведением типов,

`===` выполняет строгое сравнение без приведения типов

9. С помощью какой конструкции языка можно управлять потоком выполнения скрипта и отслеживать ошибки?

Ответ: Для управления потоком выполнения и отслеживания ошибок используется конструкция `try-catch-finally`:

### **Информация о разработчиках**

Овсянников Михаил Сергеевич, старший преподаватель кафедры теоретических основ информатики ТГУ.