

Ministry of Science and Higher Education of the Russian Federation
NATIONAL RESEARCH
TOMSK STATE UNIVERSITY (NR TSU)

Institute of Applied Mathematics and Computer Science


A. V. Zamyatin

Evaluation materials of the current control and intermediate certification in the discipline

(Evaluation tools by discipline)

Data Programming in Python

in the major of training

01.04.02 Applied mathematics and informatics

Orientation (profile) of training:

Big Data and Data Science

ОС составил(и):

канд. физ.-мат. наук, доцент

доцент кафедры компьютерной безопасности

старший преподаватель

кафедры теоретических

основ информатики ТГУ

С.И. Самохина

М.С. Овсянников

Рецензент:

канд. техн. наук, доцент,

зав. кафедрой компьютерной безопасности

С.А. Остапенко

Фонд оценочных средств одобрен на заседании учебно-методической комиссии
института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от 22.05.2025 г. №

Председатель УМК ИПМКН,

д-р техн. наук, профессор

С.П. Сущенко

Evaluation tools (ET) are an element of the system for assessing the formation of competencies among students in general or at a certain stage of its formation.

The ET is developed in accordance with the work program (WP) of the discipline.

1. Competencies and training outcomes, obtained upon the discipline mastery

Competencies	Competence indicator	Code and name of planned training outcomes that characterize the stages of competency formation	Критерии оценивания результатов обучения			
			Excellent	Good	Satisfactory	Unsatisfactory

UK-2 Able to manage a project at all stages of its life cycle OPK-3 Able to develop mathematical models and analyze them when solving problems in the field of professional activity OPK-4 Able to combine and adapt existing information and communication technologies to solve problems in the field of professional PC-1. Able to develop and apply mathematical methods, algorithms, software to solve problems in research and design activities	ICP - 2.1 Formulates the goal of the project, substantiates its significance and feasibility ICM - 2.2 Develops an action program to solve project problems, taking into account available resources and limitations. ICP - 2.3 Ensures project implementation in accordance with established goals, deadlines and costs. IOPK-3.1 Develops mathematical models in the field of applied mathematics and computer science IOPK-3.2 Analyzes mathematical models for solving applied problems of professional activity IOPK-4.1 Analyzes problems of applied mathematics and computer science using information technology IOPK-4.3 Uses modern information and communication technologies to solve problems in the field of applied mathematics and computer science, taking	Know: Master C# and Python data. Familiarize yourself with nymPy, pandas, matplotlib libraries for working with artificial intelligence in Python. Be able to: . Master working with online compilers as tools for editing, debugging, compiling and executing programs. Learn to use libraries for working with big data and artificial intelligence. Practice using data structures for storing and processing data. Learn to create custom classes in C#.	Knows the basics of the C# and Python programming languages. Can compile and debug a program. Can use Python libraries. Can create custom classes in C#. Has the necessary knowledge in the field of information technology and software, including understanding the principles of their operation, applies knowledge gained in the field of information technology and software when solving problems of professional activity, uses modern information technology, including domestic production at all stages of software systems development, codes in programming languages and conducts modular testing of IS	Knows the basics of the C# and Python programming languages. Can compile and debug a program. Can use Python libraries. Can create custom classes in C#. However, when creating a program, makes inaccuracies. Has the necessary knowledge in the field of information technology and software, including an understanding of the principles of their operation, applies knowledge gained in the field of information technology and software when solving problems of professional activity, uses modern information technology, including domestic production at all stages of software systems development, codes in programming languages and conducts modular testing of the IS, codes in	Knows the basics of the C# and Python programming languages. Can compile and debug a program. Can use Python libraries. Can create custom classes in C#. However, when choosing tools and creating programs, makes significant mistakes. Has the necessary knowledge in the field of information technology and software, including an understanding of the principles of their operation, applies knowledge gained in the field of information technology and software when solving problems of professional activity, does not use modern information technologies, including domestic production at all stages of software systems development, does not code in programming languages and does not conduct modular testing of IS, does not code in programming languages and does not conduct modular testing of the IS,	Does not know the basics of the C# and Python programming languages. Can compile and debug a program. Cannot use Python libraries. Cannot create custom classes in C#. Does not have the necessary knowledge in the field of information technology and software, including does not understand the principles of their operation, does not apply knowledge gained in the field of information technology and software when solving problems of professional activity, does not use modern information technologies, including domestic production at all stages of software systems development, does not code in programming languages and does not conduct modular testing of IS, does not code in programming languages and does not conduct modular testing of IS
--	--	---	---	---	---	---

	into account information security requirements IPC -1.2 Develops new methods, models, algorithms and software for solving problems in the field of professional activity. IPK-1.3 Develops new methods, models, algorithms and software for solving problems in the field of professional activity			programming languages and conducts modular testing of the IS, but makes inaccuracies	codes in programming languages and conducts modular testing of the IS, but makes mistakes	
--	---	--	--	---	--	--

2. Этапы формирования компетенций и виды оценочных средств

№	Этапы формирования компетенций (разделы дисциплины)	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1.	Тема 1. Общее введение в теорию компиляции	Задание 1, 2, 3, лабораторная работа
2.	Тема 2. Ликбез по компиляторам и принципам трансляции	Задание 1, 2, лабораторная работа
3.	Тема 3. Обзор современных языков	Задание 3, лабораторная работа
4.	Тема 4. Языки программирования, парадигмы и классификация	Задание 3, лабораторная работа
5.	Тема 5. Скриптовый язык Python 3	Задание 4, 5. лабораторная работа
6.	Тема 6. Скриптовый язык Javascript	Задание 4, лабораторная работа
7.	Тема 7. Компилируемый язык с сборщиком мусора Go (golang). Тема 8. Компилируемый язык Rust	Задание 2, лабораторная работа

3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине

Задание 1. Составить программу на C# для перевода чисел из одной системы счисления в другую. Системы счисления произвольные. Для этого сначала реализуем алгоритм перевода чисел из системы счисления N в десятичную систему счисления, потом из десятичной переводим в систему счисления M.

Если $N = 10$ или $M = 10$ это частный случай, его надо проверить.

Составляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы во всех нужных случаях, граничных и обычных. Отчёт прикрепляем в Moodle, лучше в формате pdf.

Задание 2. Реализовать алгоритм Ханойские башни для любого N.

Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы, здесь тоже подробно и с комментариями

Задание 3. Дано арифметическое выражение в виде строки. Найти его значение.

Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и скрины результатов выполнения программы во всех нужных случаях, показать, как вычисляются выражения с одинарным минусом, скобками, ошибками.

Задание 4. Создать небольшой телеграм-бот для указанной рассылки, используя язык программирования Python. Подобрать нужные библиотеки. Прикрепляем подробный отчёт, в котором должен быть код с подробными комментариями и ссылка на разработанный бот.

Задание 5. Междисциплинарная, задача из теории графов. Реализовать алгоритм раскраски графа для графов большой размерности. Подобрать нужные библиотеки. Прикрепляем подробный отчёт, в котором должно быть обоснование выбранных средств реализации, код с подробными комментариями и скрины результатов.

Примерные вопросы для устного опроса при текущем контроле

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

Примеры заданий для лабораторных работ :

Лабораторная работа №1. «Спамеры». Цель

работы – написать скрипт,

выполняющий рекурсивный обход сайта (напр. www.csd.tsu.ru) и вывести без дубликатов

все адреса электронной почты, содержащиеся на страницах. Для ускорения работы добавьте ограничитель на переходы (напр. 10) по ссылкам – сайт может содержать очень много страниц. Для извлечения email и url следует использовать регулярные выражения.

Базовый язык - Python 2.7 или Python 3.5. Требуется использовать библиотеки requests для http запросов и re для RegEx.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Javascript). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

Лабораторная работа №2 «Сисадмины». Цель работы – Необходимо написать скрипт, обрабатывающий лог-файл Nginx и выводящий список IP адресов, с которых производились запросы. Адреса из общей

подсети \24 необходимо группировать при выводе (напр. 10.40.0.4 и 10.40.0.231 относятся к одной подсети).

Базовый язык - Python 2.7 или Python 3.5. Требуется использовать библиотеку re для RegEx.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Javascript). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

Лабораторная работа №3 «Тусовщики». Цель работы – Необходимо написать скрипт, генерирующий html страницу со списком мероприятий (событий), связанных с IT полученных с сайта https://www.meetup.com/meetup_api/.

Необходимо заранее зарегистрироваться и получить ключ для доступа API.

В скрипте константами задаете город (любой, где много событий, напр. Boston). Диапазон дат вычисляется автоматически в виде следующей недели относительно времени запуска скрипта.

Необходимо вывести на каждый день недели список событий в виде даты, заголовка, адреса и аннотации. Для фильтрации тематики используйте ключевые слова или темы (topics).

Базовый язык - Javascript. Допускается реализация как под node.js, так и в виде скрипта в браузере.

Второй язык - любой скриптовый с динамической типизацией (напр. Ruby, Lua, Perl, Python). Не допускается семейство .Net, языки на основе JVM и все компилируемые языки.

Лабораторная работа №4 «Братство кольца». Цель работы – написать программу, имитирующую сеть TokenRing. Требуется запустить N потоков, где каждый связан с последующим. Основной поток отдает первому потоку экземпляр структуры/класса Token (data:string, recipient:int). Потоки передают токен по цепочке, пока сообщение не достигнет адресата.

Базовый язык - Go (golang). Контрольный срок сдачи - 23:59 (UTC+7) 9 декабря.

Второй язык по выбору - любой компилируемый, с динамической сборкой мусора. Допускаются языки семейства .Net и JVM

Лабораторная работа №5 «Пальцем в небо». Цель работы – написать программу, реализующую расчет пересечений полупрямых с набором отрезков в двумерной плоскости. Отрезки не пересекаются, но вершины могут совпадать. Полупрямая пересекается с ближайшим к ней отрезком, т.е. имеет 1 или 0 пересечений.

Формат ввода – первая строка – координаты начала полупрямой и точка, через которую она проходит. На каждой следующей строке координаты отрезков. Каждая точка задается вещественным числом (разделитель – точка) через запятую. Пары точек разделены пробелом. Например:

1.5,2.7 2.896,3

2.68,3 4,8.6666661

Базовый язык - Rust. Контрольный срок сдачи - 23:59 (UTC+7) 9 декабря.

Второй язык по выбору - любой компилируемый, с явным выделением памяти и без динамической сборки мусора (напр. C, C++, D, Pascal).

Для промежуточной аттестации используется традиционная шкала оценивания.

Оценка	Критерии оценивания
Отлично	Обучающийся показал творческое отношение к обучению, в совершенстве овладел всеми теоретическими вопросами обязательного материала в области теории компиляции и принципов построения и классификации языков, показал все требуемые умения и навыки в работе с скриптовыми языками Python и Javascript, а также с компилируемыми языками Go и Rust
Хорошо	Обучающийся овладел всеми теоретическими вопросами обязательного материала в области теории компиляции и принципов построения и классификации языков, частично овладел навыками использования современных языков Python, Javascript, Go и Rust, показал основные умения и навыки в работе с онлайн интерпретаторами.
Удовлетворительно	Обучающийся имеет недостаточно глубокие знания по теоретическим разделам обязательного материала дисциплины, недостаточно владеет навыками разработки программ, показал не все основные умения и навыки в работе со скриптовыми языками Python и Javascript, а также с компилируемыми языками Go и Rust.
Неудовлетворительно	Обучающийся имеет существенные пробелы по отдельным теоретическим разделам специальной дисциплины и не владеет навыками написания программ на скриптовых и компилируемых языках.