

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «Башкирский государственный университет»
Институт непрерывного образования

СОГЛАСОВАНО

И.о. директора Института непрерывного
образования



(подпись)

Е.П. Кислова

« 10 » 06 2022 г.

УТВЕРЖДАЮ

Проректор по молодежной политике и
международной деятельности



Т.Б. Великжанина

_____ 2022 г.

**Программа профессиональной переподготовки
«Основы алгоритмизации на языке программирования Python»
с присвоением квалификации
«Программист»**

УФА 2022 год

I. Общие положения

1. Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «Основы алгоритмизации на языке программирования Python» (далее – Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации», с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499», приказа Министерства образования и науки РФ от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ» (указать при необходимости); паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»); приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 28 февраля 2022 г. № 143

«Об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» и признании утратившими силу некоторых приказов Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» (далее – приказ Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации № 143); федерального государственного образовательного стандарта высшего образования по направлению подготовки 01.03.02 Прикладная математика и информатика (уровень бакалавриата), утвержденного Приказом Министерства образования и науки РФ от 10 января 2018 г. № 9 (Редакция с изменениями № 1456 от 26.11.2020), а также профессионального стандарта «Программист», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от 18 ноября 2013 г. № 679н (с изменением от 12 декабря 2016 г. № 727н).

2. Профессиональная переподготовка заинтересованных лиц (далее – Слушатели), осуществляемая в соответствии с Программой (далее – Подготовка), имеющей отраслевую направленность «Информационно-коммуникационные технологии», проводится в ФГБОУ ВО «Башкирский государственный университет» (далее – Университет) в соответствии с учебным планом в очной/заочной форме обучения.

3. Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочей программы, оценочных и методических материалов. Перечисленные документы разрабатываются Университетом самостоятельно, с учетом актуальных положений законодательства об

образовании, законодательства в области информационных технологий и смежных областей знаний ФГОС ВО и профессионального стандарта «Программист».

4. Программа регламентирует требования к профессиональной переподготовке в области проектирования, разработки и тестирования программного обеспечения.

Срок освоения Программы составляет 360 академических часов, длительность 9 месяцев.

К освоению Программы в рамках проекта допускаются лица:

- получающие высшее образование по очной форме, лица, освоившие основную профессиональную образовательную программу (далее – ОПОП ВО) бакалавриата – в объеме не менее первого курса (бакалавры 2-го курса), ОПОП ВО специалитета – не менее первого и второго курсов (специалисты 3-го курса), а также магистратуры, обучающиеся по ОПОП ВО, не отнесенным к ИТ-сфере.

5. Область профессиональной деятельности задачи, методы и средства для разработки, сопровождения и эксплуатации программного обеспечения компьютерных систем; информационные технологии (в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования, создания и поддержки информационно-коммуникационных систем и баз данных); процессы управления ИТ-инфраструктурой.

II. Цель

6. Целью подготовки слушателей по Программе является получение компетенции: обучающиеся по специальностям и направлениям подготовки, не отнесенным к ИТ-сфере, необходимой для выполнения нового вида профессиональной деятельности в области информационных технологий,

теоретических знаний и практических навыков, позволяющих строить алгоритм, анализировать его работу при разных входных данных и реализовать его при помощи Python, приобретение новой квалификации «Программист».

III. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и (или) уровней квалификации

7. Виды профессиональной деятельности, трудовая функция, указанные в профессиональном стандарте по соответствующей должности программиста, представлены в таблице 1:

Таблица 1

Характеристика новой квалификации, связанной с видом профессиональной деятельности и трудовыми функциями в соответствии с профессиональным стандартом «06.015 Программист»

Область профессиональной деятельности	Тип задач профессиональной деятельности	Код и наименование профессиональной компетенции	Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования, создания и поддержки информационно-коммуникационных систем и баз данных; в сфере создания информационных ресурсов в информационно-телекоммуникационной сети «Интернет»)	производственно-технологический	ПК 1 - способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	Составление формализованных описаний решений поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов Разработка алгоритмов решения поставленных задач в соответствии с требованиями	Формализация и алгоритмизация поставленных задач	Разработка и отладка программного кода	Программист

			технического задания или других принятых в организации нормативных документов			
06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования, создания и поддержки информационно-коммуникационных систем и баз данных; в сфере создания информационных ресурсов в информационно-телекоммуникационной сети «Интернет»)	проектный	ПК 2 - способностью разрабатывать компоненты программных комплексов, использовать современные инструментальные средства и технологии программирования	Создание программного кода в соответствии с техническим заданием (готовыми спецификациями) Оптимизация программного кода с использованием специализированных программных средств	Написание программного кода с использованием языков программирования, определения и манипулирования данными	Разработка и отладка программного кода	Программист
06 Связь, информационные и коммуникационные технологии (в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования,	организационно-управленческий	ПК 3 - способностью планировать необходимые для выполнения работы ресурсы, оценивать результаты работы	Подготовка тестовых наборов данных в соответствии с выбранной методикой Оценка и согласование	Разработка тестовых наборов данных	Проверка работоспособности и рефакторинг кода программного обеспечения	Программист

<p>создания и поддержки информационно-коммуникационных систем и баз данных; в сфере создания информационных ресурсов в информационно-телекоммуникационной сети «Интернет»)</p>	
--	--

		сроков выполнения поставленных задач			
--	--	---	--	--	--

Таблица 2

**Характеристика новой и развиваемой цифровой компетенции в ИТ-сфере,
связанной с уровнем формирования и развития в результате освоения
Программы «Основы алгоритмизации на языке программирования Python»**

Наименование сферы	Код и наименование профессиональной компетенции	Пример инструментов	0 - способность не проявляется/ проявляется в степени, недостаточной для отнесения к 1 уровню сформированности компетенции	1 - способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами	2 - способность проявляется, но обучающийся эпизодически прибегает к экспертной консультации/ самостоятельно подбирает и пользуется готовыми продуктами	3 - способность проявляется системно / обучающийся модифицирует способность под определенные задачи / создает новый продукт, обучает других
Средства программной разработки	ПК 4 - применяет принципы и основы алгоритмизации	Вычислительные алгоритмы, диалоговые, графические, обработки данных управления объектами/процессам и и т.д.	Не применяет	Применяет принципы и основы алгоритмизации под внешним контролем	Применяет принципы и основы алгоритмизации под внешним контролем, подбирает и пользуется готовыми продуктами	Применяет принципы и основы алгоритмизации при решении прикладных задач, создает новый продукт
Средства программной	ПК 5 - применяет языки	Python	Не применяет	Применяет под внешним контролем язык	Применяет язык программирования Python,	Применяет язык программирования Python при

разработки	программирования для решения профессиональных задач			программирования Python для решения профессиональных задач	самостоятельно подбирает и пользуется готовыми продуктами при решении профессиональных задач	решении профессиональных задач, создает новый продукт
Средства программной разработки	ПК 6 - разрабатывает программное обеспечение	Python	Не разрабатывает	Разрабатывает под внешним контролем на языке программирования Python программное обеспечение	Разрабатывает программное обеспечение на языке программирования Python, самостоятельно подбирает и пользуется готовыми продуктами	Разрабатывает на языке программирования Python программное обеспечение, самостоятельно создает новый продукт

IV. Характеристика новых и развиваемых цифровых компетенций, формирующихся в результате освоения программы

8. В ходе освоения Программы Слушателем приобретаются следующие профессиональные компетенции:

- ПК 1 - способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения;

- ПК 2 - способностью разрабатывать компоненты программных комплексов, использовать современные инструментальные средства и технологии программирования;

- ПК 3 - способностью планировать необходимые для выполнения работы ресурсы, оценивать результаты работы;

В ходе освоения Программы Слушателем совершенствуются следующие профессиональные компетенции:

- ПК 4 - применяет принципы и основы алгоритмизации;

- ПК 5 - применяет языки программирования для решения профессиональных задач;

- ПК 6 - разрабатывает программное обеспечение.

V. Планируемые результаты обучения по ДПП III

10. Результатами подготовки слушателей по Программе является получение компетенции, необходимой для выполнения нового вида профессиональной деятельности в области информационных технологий в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования, создания и поддержки информационно-коммуникационных систем и баз данных; приобретение новой квалификации «Программист».

11. В результате освоения Программы слушатель должен:

ПК 1 - способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения:

Знать: алгоритмические и программные решения в области системного и прикладного программного обеспечения.

Уметь: применять алгоритмические и программные подходы в области системного и прикладного программного обеспечения при решении прикладных задач.

Иметь навыки: разработки алгоритмических и программных решений в области системного и прикладного программного обеспечения.

ПК 2 - способностью разрабатывать компоненты программных комплексов, использовать современные инструментальные средства и технологии программирования:

Знать: компоненты программного кода, современные инструментальные средства и технологии программирования.

Уметь: разрабатывать компоненты программных комплексов, использовать современные инструментальные средства и технологии программирования.

Иметь навыки: разработки программ и использования инструментальных сред при решении прикладных задач.

ПК 3 - способностью планировать необходимые для выполнения работы ресурсы, оценивать результаты работы:

Знать: основные ресурсы, необходимые для решения прикладных задач.

Уметь: применять при решении прикладных задач необходимые ресурсы, используя при этом современные информационные технологии; оценивать результаты работы.

Иметь навыки: планирования ресурсов и оценивания результатов выполненной работы.

ПК 4 - применяет принципы и основы алгоритмизации:

Знать: основы информатики и алгоритмизации, общую последовательность действий, которые необходимо выполнить для построения алгоритма решения задачи.

Уметь: применять математический аппарат при выборе метода решения задачи, выделять конкретные шаги алгоритмического процесса.

Иметь навыки: построения алгоритмов поставленных задач с дальнейшей реализацией на языке программирования Python.

ПК 5 - применяет языки программирования для решения профессиональных задач:

Знать: правила разработки прикладного программного обеспечения, порядок разработки, отладки, тестирования и документирования программного продукта.

Уметь: применять инструменты и методы при разработке программного обеспечения.

Иметь навыки: разработки концепции, технического задания и программного кода при решении профессиональных задач.

ПК 6 - разрабатывает программное обеспечение:

Знать: основы проектирования на языке программирования Python.

Уметь: использовать программные средства для решения профессиональных задач при разработке программного обеспечения.

Иметь навыки: программирования на Python при решении профессиональных задач.

VI. Организационно-педагогические условия реализации ДПП

12. Реализация Программы должна обеспечить получение компетенции, необходимой для выполнения нового вида профессиональной деятельности в области информационных технологий в сфере проектирования, разработки и тестирования программного обеспечения; в сфере проектирования, создания и поддержки информационно-коммуникационных систем и баз данных; приобретение новой квалификации «Программист».

13. Учебный процесс организуется с применением дистанционных образовательных технологий, инновационных технологий и методик обучения, способных обеспечить получение слушателями знаний, умений и навыков в области связи, информационных и коммуникационных технологий.

14. Реализация Программы обеспечивается научно-педагогическими кадрами Университета, допустимо привлечение к образовательному процессу высококвалифицированных специалистов ИТ-сферы и/или дополнительного профессионального образования в части, касающейся профессиональных компетенций в области создания алгоритмов и программ, пригодных для практического применения, с обязательным участием представителей профильных организаций-работодателей. Возможно привлечение региональных руководителей цифровой трансформации (отраслевых ведомственных и/или корпоративных) к проведению итоговой аттестации, привлечение работников организаций реального сектора экономики субъектов Российской Федерации.

VII. Учебный план ДПП

15. Объем Программы составляет 360 академических часов.

16. Учебный план Программы определяет перечень, последовательность, общую трудоемкость разделов и формы контроля знаний.

Учебный план программы профессиональной переподготовки «Основы алгоритмизации на языке программирования Python»

№ п/п	Наименование раздела (модуля)	Общая трудоемкость (360 часов)	Форма контроля
1.	Современные инженерные практики разработки приложений. Основы разработки программного обеспечения на языке Python.	108	Зачет
2.	Введение в алгоритмы. Структуры данных.	36	Зачет
3.	Численные алгоритмы. Сортировка. Рекурсия.	72	Зачет
4.	Графы и деревья. Хэш-таблицы в Python.	36	Зачет
5.	Базы данных. Работа с базами данных в Python на основе SQLite.	72	Зачет
6.	Тестирование и анализ кода Python.	36	Зачет
7.	Практика в профильной сфере (распределенная)	36	Зачет
	Итоговая аттестация		Защита проекта
	Итого:	360	

VIII. Календарный учебный график

18. Календарный учебный график представляет собой график учебного процесса, устанавливающий последовательность и продолжительность обучения и итоговой аттестации по учебным дням.

Календарный учебный график программы профессиональной переподготовки «Основы алгоритмизации на языке программирования Python».

IX. Рабочая программа учебных предметов, курсов, дисциплин (модулей)

19. Рабочая программа содержит перечень разделов и тем, а также рассматриваемых в них вопросов с учетом их трудоемкости.

Рабочая программа разрабатывается Университетом с учетом профессионального стандарта «Программист».

№ п/п	Наименование и краткое содержание раздела(модуля)	Объем, часов
1	Современные инженерные практики разработки приложений. Основы разработки программного обеспечения на языке Python	108
1.1	Язык программирования Python. Преимущества и недостатки. Структура программы на Python. Работа в командной строке и её настройка.	12
1.2	PyCharm Community. Основы работы.	12
1.3	Введение в программирование на Python. Синтаксис языка, описание встроенных типов. Общепринятый стиль программирования Python. Основные правила PEP 8.	24
1.4	Операторы. Двоичные операторы. Операторы присваивания. Условные операторы и циклы. Операторы сравнения. Оператор ветвления. Использование встроенных и создание собственных функций.	24
1.5	Введение в Git. Разработка в соответствии с современными инженерными практиками.	12
1.6	Работа с файлами и каталогами в Python.	10
1.7	Эффективная обработка исключений.	8
1.8	Промежуточная аттестация (Зачет)	6
2	Введение в алгоритмы. Структуры данных	36
2.1	Введение в алгоритмы. Сложность алгоритмов.	6
2.2	Асимптотический анализ алгоритма. Оценка времени работы алгоритма, анализ затраченной памяти.	8
2.3	Простые алгоритмы: числа Фибоначи, наибольший общий делитель. Комбинаторика.	8
2.4	Структуры данных: массивы, очереди, стеки.	8
2.5	Промежуточная аттестация (Зачет)	6
3	Численные алгоритмы. Сортировка. Рекурсия	72
3.1	Сортировка. Пять основных алгоритмов сортировки. Sort в Python. Линейный и бинарный поиск.	16
3.2	Механизм рекурсивных вызовов. Задача о ханойской башне. Оптимизация рекурсии, преобразование рекурсивных алгоритмов к итеративным.	16
3.3	Система линейных алгебраических уравнений (СЛАУ). Итерационные и прямые методы решения СЛАУ.	18
3.4	Генерация случайных чисел. Алгоритмы быстрого возведения в степень.	16

	Численные алгоритмы. Интерполяция и аппроксимация. Метод наименьших квадратов.	
3.5	Промежуточная аттестация (Зачет)	6
4	Графы и деревья. Хэш-таблицы в Python	36
4.1	Хэширование. Хэш-таблицы и как они устроены, Python-словари и ассоциативные массивы. Коллизии и борьба с ними.	6
4.2	Дерево. Обход деревьев в ширину и глубину. Поиск элемента в двоичном дереве. Различные виды деревьев.	8
4.3	Дерево принятий решений. Приемы оптимизации деревьев.	8
4.4	Теория графов. Алгоритм Дейкстры для нахождения кратчайшего пути между двумя точками.	8
4.5	Промежуточная аттестация (Зачет)	6
5	Базы данных. Работа с базами данных в Python на основе SQLite	72
5.1	Что такое базы данных и для чего нужны. Реляционные и не реляционные базы данных.	14
5.2	Основы структурированного языка запросов SQL. Модели данных и SQL-запросы в реляционных БД. Связи «многие ко многим» в языке SQL.	18
5.3	Сравнение SQLite с другими базами данных SQL. Работа с базами данных в Python.	16
5.4	Типы данных SQLite, создание таблиц, просмотр данных. Пример разработки реляционной базы данных и её реализация на базе SQLite.	18
5.5	Промежуточная аттестация (Зачет)	6
6	Тестирование и анализ кода Python	36
6.1	Введение в разработку и автотестирование.	6
6.2	Автоматизированное и ручное тестирование.	6
6.3	Виды тестирования. Unit-тесты, интеграционные тесты, модульное тестирование.	8
6.4	Пример написания модульного и интеграционного теста.	10
6.5	Промежуточная аттестация (Зачет)	6
7	Практика в профильной сфере (распределенная)	36
	Итоговая аттестация (Защита проекта)	

20. Учебно-тематический план Программы определяет тематическое содержание, последовательность разделов и (или) тем и их трудоемкость.

№ п/п	Наименование раздела(модуля)	Количество часов		
		аудиторных		Самостоятельной работы (изучение теоретического материала, выполнение контрольных работ, подготовка к семинарам и промежуточным аттестациям, разработка проекта и др. заданий)
		Лекции	Лабораторные занятия	
1.	Современные инженерные практики разработки приложений. Основы разработки программного обеспечения на языке Python.	26	70	6
2.	Введение в алгоритмы. Структуры данных.	6	18	6
3.	Численные алгоритмы. Сортировка. Рекурсия.	18	42	6
4.	Графы и деревья. Хэш-таблицы в Python.	4	20	6
5.	Базы данных. Работа с базами данных в Python на основе SQLite.	18	42	6
6.	Тестирование и анализ кода Python.	6	18	6
7	Практика в профильной сфере (распределенная)	36		
	Промежуточная аттестация	36		
	Итоговая аттестация			

Х. Формы аттестации

21. Слушатели, успешно выполнившие все элементы учебного плана, допускаются к итоговой аттестации.

Итоговая аттестация по Программе проводится в форме проектной работы, которая представляет собой работу, содержащую комплекс взаимосвязанных мероприятий, направленных на создание нового программного продукта.

22. Лицам, успешно освоившим Программу (в области создания алгоритмов и программ, пригодных для практического применения, или навыков использования и освоения цифровых технологий, необходимых для выполнения нового вида профессиональной деятельности) и прошедшим итоговую аттестацию в рамках проекта «Цифровые кафедры», выдается документ о квалификации: диплом о профессиональной переподготовке.

При освоении ДПП ПП параллельно с получением высшего образования диплом о профессиональной переподготовке выдается не ранее получения соответствующего документа об образовании и о квалификации (за исключением лиц, имеющих среднее профессиональное или высшее образование).

23. Лицам, не прошедшим итоговую аттестацию или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть Программы и (или) отчисленным из Университета, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому Университетом.

ХІ. Оценочные материалы

24. Контроль знаний, полученных слушателями при освоении разделов (модулей) Программы, осуществляется в следующих формах:

- текущий контроль успеваемости – обеспечивает оценивание хода освоения дисциплин Программы, проводится в форме сдачи лабораторных работ;

- промежуточная аттестация – завершает изучение отдельной дисциплины Программы, проводится в форме тестирования;

- итоговая аттестация (защита проекта) – завершает изучение всей программы.

Защита проектной работы является публичной (открытой).

Защита проектной работы не должна превышать 30 минут. Обучающийся излагает сущность и основные положения проектной работы не менее 5 и не более 10 минут.

В докладе необходимо сделать акцент на собственных, оригинальных разработках, имеющих теоретическую и практическую значимость.

Результаты защиты проектной работы определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

При оценивании проектной работы могут применяться следующие критерии:

– актуальность темы исследования;

– качество анализа, систематизации, обобщения собранного теоретического материала, обоснованность и четкость сформулированных выводов;

– четкость структуры работы, логичность изложения материала, раскрытие методологической основы исследования;

– стиль изложения, соблюдение норм русского языка;

– обоснованность и ценность полученных результатов исследования и выводов, возможность их применения в практической деятельности (при

наличии);

– соответствие формы представления проектной работы установленным требованиям;

– качество устного доклада, свободное владение материалом проектной работы;

– глубина и точность ответов на вопросы, замечания и рекомендации во время защиты проектной работы.

25. В ходе освоения Программы каждый слушатель выполняет следующие отчетные работы:

№ п/п	Наименование раздела (модуля)	Задание	Критерии оценки
1.	Современные инженерные практики разработки приложений. Основы разработки программного обеспечения на языке Python.	<i>(Лабораторные работы (п. 26.1 – п. 26.6))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
2.	Введение в алгоритмы. Структуры данных.	<i>(Лабораторные работы (п. 27.1 – п. 27.4))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
3.	Численные алгоритмы. Сортировка. Рекурсия.	<i>(Лабораторные работы (п. 28.1 – п. 28.4))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
4.	Графы и деревья. Хэш-таблицы в Python.	<i>(Лабораторные работы (п. 29.1 – п. 29.4))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
5.	Базы данных. Работа с базами данных в Python на основе SQLite.	<i>(Лабораторные работы (п. 30.1 – п. 30.4))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
6.	Тестирование и анализ кода Python.	<i>(Лабораторные работы (п. 31.1 – п. 31.4))</i>	выполнено 50% – 2 балла выполнено 65% – 3 балла выполнено 80% – 4 балла выполнено 90% – 5 баллов
7.	Практика в профильной сфере (распределенная)		
	Промежуточная аттестация	<i>(Тесты (п. 26.6, п. 27.5, п. 28.5, п. 29.5, п. 30.5, п. 31.5))</i>	0-59% – не зачтено 60-100% – зачтено
	Итоговая аттестация	<i>(Проект (п. 32))</i>	«неудовлетворительно» –

			<p>работа не выполнена; «удовлетворительно» – работа выполнена с грубейшими ошибками или выполнена на базовом уровне; «хорошо» – работа выполнена, но имеются незначительные ошибки; «отлично» – работа выполнена в полном объеме.</p>
--	--	--	--

26. Текущий контроль. Перечень лабораторных заданий

Модуль 1. Современные инженерные практики разработки приложений. Основы разработки программного обеспечения на языке Python (ПК-1, ПК-5, ПК-6)

26.1 Тема «Язык программирования Python. Преимущества и недостатки. Структура программы на Python. Работа в командной строке и её настройка».

При помощи языка программирования Python определите значение выражения 5 в степени 99. Какое число получилось? Составьте программный код, который позволит вычислить это значение, запустите программу, выведите ответ в CMD.

26.2 Тема «PyCharm Community. Основы работы. Введение в программирование на Python. Синтаксис языка, описание встроенных типов. Общепринятый стиль программирования Python. Основные правила PEP 8».

Лабораторная работа «Установка PyCharm Community».

1. Перейти на сайт JetBrains <https://www.jetbrains.com/pycharm/>. PyCharm – это проект JetBrains.

2. Нажать «Download». Произойдет перенаправление на страницу загрузки PyCharm.

3. Выбрать версию PyCharm: Community или Professional (необходимо

выбрать Community). Professional — это платная версия с полным набором функций. Она идеально подходит для профессиональной разработки. Community бесплатная. Ею можно пользоваться благодаря набору базовых возможностей. После нажатия на кнопку «Загрузить» скачивание начнется автоматически.

4. Запустить установку (~ pycharm-community-2019.2). Нажмите «Next».

5. Оставить папку установки по умолчанию. Это папка C:\Program files(x86)\JetBrains\PyCharm Community Edition 2019.2. Если выбрана другая, лучше указать именно это направление. Нажмите «Next».

Поставить галочки напротив обоих пунктов, если нужен ярлык для рабочего стола. Нажать «Next».

6. Указать имя для отображения в стартовом меню. По умолчанию это JetBrains. Нажать «Install».

Установка начнется автоматически. Дождаться завершения.

После завершения установки поставить галочку напротив «Run PyCharm Community Edition» для запуска, а затем «Finish».

7. Завершение установки PyCharm

Если PyCharm установлен впервые, то выбрать пункт «I do not have a previous version of PyCharm or I do not want to import my settings» и «OK».

PyCharm запустится. Первый раз этой займет больше времени.

8. После приветственного окна нажать «Create New Project».

Выбрать директорию для сохранения проекта. Во втором поле нужно указать местоположение интерпретатора Python (который уже должен быть установлен). Обычно PyCharm находит его самостоятельно. Если не удалось, нужно указать путь к нему и нажать «Create».

9. Нажать «File > New > Python File». Ввести название нового файла. Он откроется с чистой консолью. Можно начинать писать программу для

проверки. Напишите что-то простое (например, программу, которая выводит «Hello World!» на экран). Используйте этот код, чтобы проверить работу PyCharm.

```
print(«Hello World!»)
```

10. В любой области кликнуть правой кнопкой мыши и выбрать «Run». Программа выполнится, а вывод будет показан на экране.

26.3 Тема «Операторы. Двоичные операторы. Операторы присваивания. Условные операторы и циклы. Операторы сравнения. Оператор ветвления. Использование встроенных и создание собственных функций».

Написать программу для решения примера. Предусмотреть проверку деления на ноль. Все необходимые переменные пользователь вводит через консоль. Запись |значение| означает «взять по модулю», т.е. если значение получится отрицательным, необходимо сменить знак с минуса на плюс. Для вычисления примеров понадобится библиотека math.

26.4 Тема «Введение в Git. Разработка в соответствии с современными инженерными практиками».

Создание новых репозиториях в системе контроля Git в среде GitBash, используя команды в CMD.

26.5 Тема «Работа с файлами и каталогами в Python. Эффективная обработка исключений».

Программа должна создавать файл *.xls, записать в него сгенерированный случайным образом массив чисел. Затем, с помощью реализованного алгоритма сортировки, одного из предложенных преподавателем, записать отсортированную последовательность чисел в ранее созданный файл *.xls.

Алгоритмы сортировки:

- Сортировка выбором
- Сортировка вставками
- Сортировка «Методом пузырька»
- Сортировка Шелла
- Быстрая сортировка

26.6 Промежуточная аттестация. Модуль «Введение в алгоритмы.

Структуры данных».

Тест

Вопрос 1

Какие существуют типы переменных (выбрать несколько вариантов):

Варианты ответов

- 1) float
- 2) str
- 3) num
- 4) int
- 5) bool
- 6) real

Вопрос 2

Переменная int:

Варианты ответов

- 1) вещественная переменная
- 2) символьная строка
- 3) логическая переменная
- 4) целая переменная

Вопрос 3

Переменная float:

Варианты ответов

- 1) вещественная переменная
- 2) символьная строка
- 3) логическая переменная
- 4) целая переменная

Вопрос 4

Переменная str:

Варианты ответов

- 1) вещественная переменная
- 2) символьная строка
- 3) логическая переменная
- 4) целая переменная

Вопрос 5

Переменная bool:

Варианты ответов

- 1) вещественная переменная
- 2) символьная строка
- 3) логическая переменная
- 4) целая переменная

Вопрос 6

Имена переменных не могут включать:

Варианты ответов

- 1) Русские буквы
- 2) Латинские буквы
- 3) Пробелы

- 4) Скобки, знаки + = ! ? b др.
- 5) Цифры

Вопрос 7

Какие имена являются правильными в PYTHON

Варианты ответов

- 1) G
- 2) ABC
- 3) 41N
- 4) Game2
- 5) a+b
- 6) _ab

Вопрос 8

Что будет в результате выполнения программы:

```
a = 20
```

```
b = a + 4
```

```
a = b * 100
```

```
print(a)
```

В качестве ответа введите число

Вопрос 9

Что будет в результате следующего действия `print(2**3)`

В качестве ответа введите число

Вопрос 10

Что будет в результате выполнения следующего действия `print(23 % 2)`

Введите число:

Вопрос 11

Результатом вычисления `print(23 // 3)` будет число:

Вопрос 12

Что будет в результате выполнения следующего алгоритма:

Входные данные: 57

```
x = int(input())
```

```
if x > 0:
```

```
    print(x)
```

```
else:
```

```
    print(-x)
```

Вопрос 13

Что будет в результате выполнения следующего алгоритма:

Входные данные: -57

```
x = int(input())
```

```
if x > 0:
```

```
    print(x)
```

```
else:
```

```
    print(-x)
```

Вопрос 14

Что будет в результате выполнения следующего алгоритма программы:

Входные данные:

15

45


```
a = int(input())
b = int(input())
if a % 10 == 0 or b % 10 == 0:
    print('YES')
else:
    print('NO')
```

Варианты ответов

- 1) YES
- 2) NO

27. Текущий контроль

Модуль 2. Введение в алгоритмы. Структуры данных. (ПК-1, ПК-2, ПК-6)

27.1 Тема «Введение в алгоритмы. Сложность алгоритмов».

Дан номер года (положительное целое число). Определить количество дней в этом году. Следует обратить внимание на то, что обычный год насчитывает 365 дней, а високосный – 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400.

27.2 Тема «Асимптотический анализ алгоритма. Оценка времени работы алгоритма, анализ затраченной памяти».

I. Реализуйте одномерные методы перебора, дихотомии и золотого сечения для приближенного (с точностью $\varepsilon = 0,001$) поиска $x: f(x) \rightarrow \min$ для следующих функций и областей допустимых значений:

1. $f(x) = x^3, x \in [0, 1]$;
2. $f(x) = |x - 0,2|, x \in [0, 1]$;
3. $f(x) = x \sin 1/x, x \in [0,1, 1]$.

Подсчитайте количество вычислений функции f и количество произведенных итераций для каждого метода. Объясните различия в полученных результатах, если таковые имеются.

II. Сгенерируйте случайные значения $\alpha \in (0, 1)$ и $\beta \in (0, 1)$. С использованием этих значений сгенерируйте массив зашумленных данных (x_k, y_k) для $k = 0, \dots, 100$.

Аппроксимируйте полученные данные линейной и рациональной функциями с помощью метода наименьших квадратов путем численной (с точностью $\varepsilon = 0,001$) минимизации функции

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2$$

Для решения задачи минимизации используйте методы перебора, Гаусса и Нелдера-Мида. При необходимости самостоятельно задайте начальные приближения и прочие параметры методов. На графиках (отдельно для каждой аппроксимирующей функции) изобразите массив сгенерированных данных и графики аппроксимирующих функций, полученных с помощью указанных методов численной оптимизации. Проведите анализ полученных результатов.

27.3 Тема «Простые алгоритмы: числа Фибоначчи, наибольший общий делитель. Комбинаторика».

Представлены «неправильные» числа Фибоначчи, в которых n -е число равно сумме $n-1$ -го и $n-3$ -го. Более формально, $F_0=F_1=F_2=1$, $F_n=F_{n-1}+F_{n-3}$, при $n>2$.

Вычислите n -е «неправильное» число Фибоначчи. Попробуйте реализовать рекурсивную функцию.

27.4 Тема «Структуры данных: массивы, очереди, стеки».

Сгенерируйте случайные значения $\alpha \in (0, 1)$ и $\beta \in (0, 1)$. С использованием этих значений сгенерируйте массив зашумленных данных

(x_k, y_k) для $k = 0, \dots, 100$. Аппроксимируйте полученные данные линейной и рациональной функциями с помощью метода наименьших квадратов путем численной (с точностью $\varepsilon = 0,001$) минимизации функции

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2$$

Для решения задачи минимизации используйте градиентный спуск, метод сопряженных градиентов, метод Ньютона и алгоритм Левенберга-Марквардта. При необходимости самостоятельно задайте начальные приближения и прочие параметры методов. На графике (отдельно для каждой аппроксимирующей функции) изобразите массив сгенерированных данных и графики аппроксимирующих функций, полученных с помощью указанных алгоритмов численной оптимизации. Проведите анализ полученных результатов в терминах количества произведенных итераций.

27.5 Промежуточная аттестация. Модуль «Введение в алгоритмы. Структуры данных».

Вопрос 1

К примерам высокоуровневых языков программирования следует отнести

Варианты ответов

- 1) C++
- 2) Visual Basic
- 3) Java

Вопрос 2

Из приведенных ниже записей выделите высокоуровневые языки программирования:

Варианты ответов

- 1) Python

2) Algol

3) Ruby

Вопрос 3

Какие из приведенных ниже записей следует отнести к высокоуровневым языкам программирования?

Варианты ответов

1) Delphi

2) Perl

3) Cobol

Вопрос 4

Транслятор, выполняющий преобразование программы, составленной на исходном языке, в объектный модуль, носит название

Варианты ответов

1) компилятор

2) имитатор

3) детерминатор

Вопрос 5

Трансляция программы на язык, близкий к машинному, носит название

Варианты ответов

1) интеграция

2) компиляция

3) сегрегация

Вопрос 6

Из приведенных ниже записей выделите типы компиляторов:

Варианты ответов

- 1) векторизующий
- 2) диалоговый
- 3) отладочный

Вопрос 7

К видам компиляции следует отнести

Варианты ответов

- 1) пакетную
- 2) построчную
- 3) условную
- 4) безусловную

Вопрос 8

Построчная компиляция носит название

Варианты ответов

- 1) терминация
- 2) интерпретация
- 3) модификация

Вопрос 9

Перевод программы с низкоуровневого языка на высокоуровневый носит название

Варианты ответов

- 1) детерминация
- 2) детализация
- 3) декомпиляция

Вопрос 10

Теоретической моделью процедурного программирования служит

алгоритмическая система под названием

Варианты ответов

- 1) машина Поста
- 2) машина Тьюринга
- 3) машина Максвелла

28. Текущий контроль

Модуль 3. Численные алгоритмы. Сортировка. Рекурсия. (ПК-1, ПК-2, ПК-4)

28.1 Тема «Сортировка. Пять основных алгоритмов сортировки. Sort в Python. Линейный и бинарный поиск».

Ввести с клавиатуры два одномерных массива, количество элементов которых задана пользователем. Осуществить обмен значений двух векторов и определить их скалярное произведение.

28.2 Тема «Механизм рекурсивных вызовов. Задача о ханойской башне. Оптимизация рекурсии, преобразование рекурсивных алгоритмов к итеративным».

Легенда (о Ханойской башне). Легенда гласит, что в Великом храме города Бенарас, под собором, отмечающим середину мира, находится бронзовый диск, на котором укреплены 3 алмазных стержня, высотой в один локоть и толщиной с пчелу. Давным-давно, в самом начале времён, монахи этого монастыря провинились перед богом Брахмой. Разгневанный, Брахма поместил на один из стержней 64 диска из чистого золота, причем так, что каждый меньший диск лежит на большем. Он сказал монахам перекладывать диски со стержня на стержень по одному, причём запретил класть больший диск на меньший. Как только все 64 диска будут переложены со стержня, на который Брахма сложил их при создании мира, на другой стержень, башня

вместе с храмом обратятся в пыль и под громовые раскаты погибнет мир.

Возможно ли монахам справиться с заданием Брахмы, если монахи бессмертны? При решении такой задачи можно сначала попробовать решить её «вручную»: поперекидывать диски, стараясь переместить все на второй стержень. Даже если таким способом не получается решить задачу полностью, можно заметить какие-то закономерности и лучше разобраться с аналогичной задачей для «маленького» количества дисков.

28.3 Тема «Система линейных алгебраических уравнений(СЛАУ). Итерационные и прямые методы решения СЛАУ».

Решить СЛАУ при помощи NumPy, построить график функции и её Фурье-образа, разложить дробно-рациональную функцию на множители, на слагаемые (в символьном виде).

28.4 Тема «Генерация случайных чисел. Алгоритмы быстрого возведения в степень. Численные алгоритмы. Интерполяция и аппроксимация. Метод наименьших квадратов».

В программе генерируется случайное целое число в диапазоне от 0 до 75. У пользователя есть четыре попытки его отгадать. Необходимо использовать только конструкцию if-elif-else и выводить пользователю сообщения о том, больше или меньше предложенное им число. В случае неудачи необходимо вывести правильный ответ на экран.

28.5 Промежуточная аттестация. Модуль «Численные алгоритмы. Сортировка. Рекурсия».

Вопрос 1. Какие из видов функций являются уникальными?

Варианты ответов

- 1) генераторы
- 2) итераторы
- 3) именные

4) анонимные

Вопрос 2. Чем являются функции в Python?

Варианты ответов

- 1) классами
- 2) методами
- 3) объектами
- 4) экземплярами

Вопрос 3. Какое зарезервированное слово применяется для определения функции в Python?

Варианты ответов

- 1) procedure
- 2) define
- 3) func
- 4) def

Вопрос 4. Каково основное назначение рекурсивной функции?

Варианты ответов

- 1) замена цикла
- 2) работа со списками
- 3) возвращение необходимого результата
- 4) итерационный перебор через циклы

Вопрос 5. За создание какой функции отвечает конструкция def?

Варианты ответов

- 1) анонимной функции
- 2) функции-итератора
- 3) рекурсивной функции

4) именной функции

Вопрос 6. За что отвечает следующий код:

```
(lambda x, y: x + y)
```

Варианты ответов

- 1) за создание анонимной функции
- 2) за создание именной функции
- 3) за создание lambda-уравнения
- 4) за создание рекурсивной функции

Вопрос 7. Какая функция создается с помощью следующего кода:

```
def add(x, y):  
    return x + y
```

Варианты ответов

- 1) анонимная функция
- 2) это не является функцией
- 3) рекурсивная функция
- 4) именная функция

Вопрос 8. Какие вычисления происходят при вызове рекурсивной функции?

- 1) односторонние
- 2) циклические
- 3) обратные
- 4) ленивые

Вопрос 9. Что такое рекурсия?

- 1) вызов функции в обратном порядке ее алгоритма
- 2) возвращение результатов функции

- 3) описание какого-либо объекта внутри этого самого объекта
- 4) подбор параметров функции

Вопрос 10. Какая конструкция служит для создания анонимной функции?

Варианты ответов

- 1) omega
- 2) lambda
- 3) lambda func
- 4) anonymous

29. Текущий контроль

Модуль 4. Графы и деревья. Хэш-таблицы в Python. (ПК-1, ПК-4, ПК-5)

29.1 Тема «Хэширование. Хэш-таблицы и как они устроены, Python-словари и ассоциативные массивы. Коллизии и борьба с ними».

Составить хеш-функцию в соответствии с заданным вариантом и проанализировать ее. При необходимости доработать хеш-функцию. Разработать программный код, который должен выполнять следующие функции: создание хеш-таблицы; добавление дополнительных элементов в хеш-таблицу; просмотр хеш-таблицы; поиск элементов в хеш-таблице; удаление элементов из хеш-таблицы.

29.2 Тема «Дерево. Обход деревьев в ширину и глубину. Поиск элемента в двоичном дереве. Различные виды деревьев».

1. Разработать программный код построения дерева для массива целых чисел.
2. Построить дерево из 100, 200 ,..., 700 вершин. Распечатать обход

дерева слева направо.

3. Вычислить размер дерева, контрольную сумму, высоту и среднюю высоту как количество уровней.

29.3 Тема «Дерево принятий решений. Приемы оптимизации деревьев».

Рассмотреть понятие дерева решений, рассмотреть варианты применения данной классификации, обучить модель на основе классов, отобразить дополнительный класс на модели и посмотреть результат.

29.4 Тема «Теория графов. Алгоритм Дейкстры для нахождения кратчайшего пути между двумя точками».

Написать программу, реализующую алгоритм поиска бикомпонент орграфа, заданного своей матрицей смежности. Использовать графические библиотеки языка Python для визуализации графа и его бикомпонент.

29.5 Промежуточная аттестация. Модуль «Графы и деревья. Хэш-таблицы в Python».

Перечень контрольных вопросов

1. В чем состоит основное отличие древовидных структур от списковых?
2. Как рекурсивно определяется дерево?
3. Какие типы вершин существуют в деревьях?
4. Какие можно выделить типы деревьев?
5. Какие деревья называются двоичными?
6. Какие деревья называются упорядоченными?
7. Какие основные понятия связываются с деревьями?
8. Какие основные операции характерны при использовании деревьев?
9. Какую структуру имеют вершины двоичного дерева?
10. Почему для деревьев существует несколько правил обхода вершин?
11. Какие правила обхода вершин дерева являются основными?

12. Как выполняется обход дерева в прямом направлении?
13. Как выполняется обход дерева в симметричном направлении?
14. Как выполняется обход дерева в обратном направлении?
15. Как выполняется обход дерева в обратно-симметричном направлении?

30. Текущий контроль

Модуль 5. Базы данных. Работа с базами данных в Python на основе SQLite. (ПК-3, ПК-5, ПК-6)

30.1 Тема «Что такое базы данных и для чего нужны. Реляционные и не реляционные базы данных».

Создать программу для работы с базой данных, добавив обработку исключительных ситуаций. База данных должна содержать не менее 7 связанных таблиц. Программа должна обладать графическим интерфейсом (возможность отображать таблицы с помощью библиотеки Pandas).

30.2 Тема «Основы структурированного языка запросов SQL. Модели данных и SQL-запросы в реляционных БД. Связи «многий ко многим» в языке SQL».

Создать с помощью OpenServer базу данных PostgreSQL и пользователя с паролем для подключения.

30.3 Тема «Сравнение SQLite с другими базами данных SQL. Работа с базами данных в Python».

Разработать интерфейс к СУБД. В интерфейсе должны быть реализованы следующие функции:

- выбор таблицы для просмотра;
- постраничный просмотр записей в таблице (по одной записи);
- табличный просмотр записей в таблице (сразу вся или порциями по

10-20 строк);

- добавление записи в таблицу;
- удаление записи из таблицы.

30.4 Тема «Типы данных SQLite, создание таблиц, просмотр данных.

Пример разработки реляционной базы данных и её реализация на базе SQLite».

Изучить используемый в реляционных СУБД оператор извлечения данных из таблиц. Получить навыки работы с оператором SELECT в программе «SQL Server Managment Studio».

30.5 Промежуточная аттестация. Модуль «Базы данных. Работа с базами данных в Python на основе SQLite».

Вопрос 1

Укажите, как правильно подключать внешний модуль для работы с базой данных SQLite

Варианты ответов

- 1) `import sqlite`
- 2) `import sqlite3`
- 3) `import SQLite`
- 4) `import SQLight`

Вопрос 2

Укажите, как правильно подключать внешний модуль для работы с базой данных SQLite

Варианты ответов

- 1) `import sql`
- 2) `import sqlite3`
- 3) `import SQLITE`

4) import SQL

Вопрос 3

Укажите, как правильно подключать внешний модуль для работы с базой данных SQLite

Варианты ответов

- 1) import sql lite
- 2) import sqlite3
- 3) import SQL-lite
- 4) import SQ-Light

Вопрос 4

В результате выполнения кода

```
import sqlite3
```

```
conn = sqlite3.connect('music.db')
```

```
cur = conn.cursor()
```

```
cur.execute('DROP TABLE IF EXISTS Tracks ')
```

```
cur.execute('CREATE TABLE Tracks (title TEXT, plays INTEGER)')
```

```
conn.close()
```

был создан файл базы данных music.db. Укажите, куда он помещен

Варианты ответов

- 1) в оперативную память
- 2) на рабочий стол
- 3) в папку Python
- 4) в корневой каталог диск C:

Вопрос 5

В результате выполнения кода

```
import sqlite3
conn = sqlite3.connect('music.db')
cur = conn.cursor()
cur.execute('DROP TABLE IF EXISTS Tracks ')
cur.execute('CREATE TABLE Tracks (title TEXT, plays INTEGER)')
conn.close()
```

был создан файл базы данных music.db. Укажите, куда он помещен

Варианты ответов

- 1) в папку "Загрузки"
- 2) в папку "Downloads"
- 3) в папку Python
- 4) в корневой каталог диск D:

Вопрос 6

В результате выполнения кода

```
import sqlite3
conn = sqlite3.connect('music.db')
cur = conn.cursor()
cur.execute('DROP TABLE IF EXISTS Tracks ')
cur.execute('CREATE TABLE Tracks (title TEXT, plays INTEGER)')
conn.close()
```

был создан файл базы данных music.db. Укажите, куда он помещен

Варианты ответов

- 1) в папку "Мои документы"
- 2) в папку "Мои рисунки"
- 3) в папку Python
- 4) в корневой каталог диск C:

Вопрос 7

Укажите, какие поля составляют структуры базы данных music.db

Варианты ответов

- 1) music, Tracks
- 2) TABLE, title
- 3) title, plays
- 4) Tracks, title

Вопрос 8

Укажите, какие поля составляют структуры базы данных music.db

Варианты ответов

- 1) DROP, EXISTS
- 2) TABLE, plays
- 3) title, plays
- 4) Tracks, plays

Вопрос 9

Укажите, какие поля составляют структуры базы данных music.db

Варианты ответов

- 1) music, plays
- 2) TABLE, DROP
- 3) title, plays
- 4) Tracks, CREATE

Вопрос 10

Укажите тип поля title базы данных music.db

Варианты ответов

- 1) булевский

- 2) TEXT
- 3) кортеж
- 4) список

31. Текущий контроль

Модуль 6. Тестирование и анализ кода Python. (ПК-1, ПК-5, ПК-6)

31.1 Тема «Введение в разработку и автотестирование».

Вычислить дисперсию (D) по всем элементам массива, заменить на среднее значение те элементы массива, отклонение которых от среднего превышает σ , где $\sigma = \sqrt{D}$.

31.2 Тема «Автоматизированное и ручное тестирование».

Изучить основные методы тестирования программных систем, сформировать единое представление о реализуемом ПО, определить несоответствия функционала реализуемого ПО с ТЗ.

31.3 Тема «Виды тестирования. Unit-тесты, интеграционные тесты, модульное тестирование».

Для реального программного модуля термогазодинамического расчета, в который искусственно внесена ошибка вычислительного характера, необходимо обнаружить факт наличия ошибки и локализовать её в программном коде методом стохастических испытаний.

31.4 Тема «Пример написания модульного и интеграционного теста».

Предлагается протестировать класс TCommand, который реализует команду для склада. Этот класс содержит единственный метод TCommand.GetFullName(), спецификация которого описана следующим образом:

Операция GetFullName() возвращает полное имя команды, соответствующее ее допустимому коду, указанному в поле NameCommand. В

противном случае возвращается сообщение «ОШИБКА: Неверный код команды». Операция может быть применена в любой момент.

Необходимо разработать спецификацию тестового случая для тестирования метода `GetFullName` на основе приведенной спецификации класса.

31.5 Промежуточная аттестация. Модуль «Тестирование и анализ кода Python».

Перечень контрольных вопросов

1. Какие подходы к подготовке тестовых данных рассмотрены к настоящему моменту?

2. Что такое «парное тестирование»? В чем заключаются его достоинства?

3. Зачем оценивать качество тестирования?

4. Как измеряется покрытие кода?

5. Почему важно тестирование с самых ранних этапов разработки?

Критерии оценивания лабораторных заданий:

- 10 баллов выставляется слушателю, если он правильно и полно выполнил задание, представленный код не содержит ошибок, при написании кода использовались общепринятые нормы и принципы;

- 8-9 баллов выставляется слушателю, если он правильно и полно выполнил задание, однако в работе присутствуют некоторые недочеты, в целом не влияющие на результат работы; представленный отчет по работе не содержит ошибок, при выполнении работы в основном использовались общепринятые нормы и принципы;

- 6-7 баллов выставляется слушателю, если он выполнил задание не менее, чем на 60%; при этом не использовались общепринятые нормы и принципы;

XII. Материально-техническое и учебно-методическое обеспечение Программы

Наименование специальных помещений и помещений для самостоятельной работы	Вид занятия	Оснащенность специальных помещений и помещений для самостоятельной работы
1	2	3
<p>1. учебная аудитория для проведения занятий лекционного типа: аудитория № 405 (гуманитарный корпус), аудитория № 419 (гуманитарный корпус)</p> <p>2. учебная аудитория для проведения лабораторных работ: компьютерный класс аудитория № 404 (гуманитарный корпус), компьютерный класс аудитория № 420 (гуманитарный корпус).</p> <p>3. учебная аудитория для проведения групповых и индивидуальных консультаций: аудитория № 419 (гуманитарный корпус), компьютерный класс аудитория № 404 (гуманитарный корпус), компьютерный класс аудитория № 420 (гуманитарный корпус).</p> <p>4. учебная аудитория для текущего контроля и промежуточной аттестации: компьютерный класс аудитория № 404</p>	<p>Лекции, лабораторные занятия, самостоятельные работы</p>	<p style="text-align: center;">Аудитория № 405</p> <p>Учебная мебель, доска, вокальные радиомикрофоны AKGWMS 40 – 2шт., Интер-ая система со встроенным короткофокусным проектором PrometheanActivBoard 387 RPOMOUNTEST -1 шт., Ком-ер встраиваемый в кафедру INTELCorei3-4150/DDr3 4 Gb/HDD, Экран настенный DraperLumaAV(1:1) 96/96”244*244MV (XT1000E) -1 шт., Настольный интерактивный дисплей , ActivPanel 21S – 1 шт. , Матричный коммутатор сигналов интерфейса HDMICMPRO 4H4H – 1 шт. , Мультимедиа-проектор PanasonicPT-EW640E - 1 шт., Двух-полосный настенный громкоговоритель 20Вт/100В цвет белый(MASK4T-W)(белый) -6 шт., Петличный радиомикрофон AKGWMS45 – 1 шт. , Терминал видео конференц-связи LifeSizeIcon 600 Camera 10xPhone 2ndGeneration – 1 шт., Экран настенный DraperLumaAV(1:1) 96/96”244*244MV (XT1000E) -1 шт.</p> <p style="text-align: center;">Аудитория № 419</p> <p>Учебная мебель, Проектор</p>

<p>(гуманитарный корпус), компьютерный класс аудитория № 420 (гуманитарный корпус).</p> <p>5. помещения для самостоятельной работы: читальный зал библиотеки аудитория 402 (гуманитарный корпус), аудитория № 613 (гуманитарный корпус).</p>		<p>Optoma Ex542 i – 1 шт., Экран настенный Dinon – 1 шт.</p> <p>Аудитория № 613</p> <p>Учебная мебель, доска, моноблок стационарный – 15 шт.</p> <p>Компьютерный класс аудитория № 420</p> <p>Учебная мебель, моноблоки стационарные 15 шт.</p> <p>Компьютерный класс аудитория № 404</p> <p>Учебная мебель, компьютеры - 15 штук.</p> <p>Аудитория 402 читальный зал библиотеки</p> <p>Учебная мебель, доска, компьютеры в комплекте (5 шт.): монитор Samsung, системный блок Asus, клавиатура, мышь, стеллажи, шкафы картотечные, комбинированные.</p>
---	--	--

12.1 Ресурсы информационно-телекоммуникационной сети «Интернет», включая профессиональные базы данных и информационно-справочные системы (при необходимости):

1. <https://habrahabr.ru/flows/develop/> – Профессиональный ресурс: раздел, посвященный разработке ПО.
2. <http://www.consultant.ru/> – КонсультантПлюс.
3. <https://urait.ru/> – Электронно-библиотечная система издательства «Юрайт».
4. <https://www.prlib.ru/> – Президентская библиотека им. Б.Н.Ельцина (база данных различных профессиональных областей).

12.2 Основное оборудование:

- Компьютеры в комплекте (системный блок, мышь, клавиатура);
- Проектор;
- Мониторы Acer;

- Система аудиовизуального представления информации.

12.3 Программное обеспечение:

- Пакет языка программирования Python;

- Конструктор PyCharm;

- VMware Workstation для ОС Windows.

ХIII. Список литературы

13.1 Основная литература:

1. Глазкова А. В., Пушкарев А. Н. Технологии программирования: Учебники и учебные пособия для вузов [Электронный ресурс] – Тюмень: Тюменский государственный университет, 2018 – 44 с. – Режим доступа: http://biblioclub.ru/index.php?page=book_red&id=572375

2. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем. 2-е изд., испр. и доп. Учебник для вузов [Электронный ресурс], 2020 – 432 с. – Режим доступа: <https://urait.ru/book/programmnyaya-inzheneriya-i-tehnologii-programmirovaniya-slozhnyh-sistem-452137>

3. Сборник задач по программированию [Электронный ресурс], 2019 - 96 с. - Режим доступа: <https://lib.rucont.ru/efd/706661>

13.2 Дополнительная литература:

1. Гэддис Т. Начинаем программировать на Python. – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.

2. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.

3. Рейтц К., Шлюссер Т. Автостопом по Python. – СПб.: Питер, 2017. – 336 с.

4. Федоров Д.Ю. Программирование на языке высокого уровня Python:

учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2019. – 161 с.

Разработчик: Исмагилова Альбина Сабирьяновна, д-р физ.-мат. наук, доцент, заведующая кафедрой управления информационной безопасностью ФГБОУ ВО «Башкирский государственный университет».