



**Образовательная автономная некоммерческая организация
высшего образования**

«Дагестанский гуманитарный институт»

Адрес: 367000, Республика Дагестан, г. Махачкала, ул. Дахадаева, 136; ОГРН 1180571012203; ИНН 0572022227 /
КПП 057201001, тел. +7 (8722) 94-00-60, e-mail: mail@daggum.ru, контактное лицо: Магомедова З.Р.

Утверждаю

Декан гуманитарно-экономического
факультета

_____ У.Д.Давлетмурзаева

« 29 » января 2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.16 «Программирование на языке Python»

Код и наименование специальности: 09.03.03 Прикладная информатика

Профиль: «Информационные системы в экономике»

Квалификация выпускника: бакалавр

Махачкала, 2026

Рабочая программа дисциплины «Программирование на языке Python» разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению 09.03.03 Прикладная информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017г., № 922, в соответствии с приказом от 06 апреля 2021г., № 245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры».

Утверждена на Ученом совете ОАНО ВО "ДГИ" 29.01.2026 года
протокол №6

Содержание

	Стр.
Раздел 1. Перечень планируемых результатов обучения по дисциплине	4
Раздел 2. Место дисциплины в структуре образовательной программы	6
Раздел 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), на самостоятельную работу обучающихся и форму(ы) промежуточной аттестации	6
Раздел 4. Содержание дисциплины, структурированное по темам с указанием отведенного на них количества академических часов и видов учебных занятий	7
Раздел 5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины ...	19
Раздел 6. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины ...	23
Раздел 7. Перечень лицензионного программного обеспечения, информационных справочных систем и профессиональных баз данных	23
Раздел 8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине ...	24
Раздел 9. Образовательные технологии ...	24

Раздел 1. Перечень планируемых результатов обучения по дисциплине

Дисциплина «Программирование на языке Python» имеет целью формирование компетенций, связанных с разработкой программного обеспечения и применением технологий программирования для решения профессиональных задач с учетом встроенного функционала языка программирования Python.

Задачами дисциплины являются формирование у обучающихся способностей разрабатывать программы, ориентированные на области системного, прикладного и специального назначения с последующим использованием инструментальной программной среды и языка программирования Python для решения профессиональных задач.

1.1. Компетенции выпускников, формируемые в результате освоения дисциплины «Программирование на языке Python» как часть планируемых результатов освоения образовательной программы высшего образования

Код компетенции	Формулировка компетенции
ПК	ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ
ПК-2	Способен разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем

1.2 Планируемые результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
ПК-2 Способен разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем	ИПК-2.1. Разрабатывает и внедряет прикладное программное обеспечение и web-сервисы информационных систем	<p><u>Знать:</u></p> <p>- современные средства разработки программ на языках высокого уровня, методы программирования и методы разработки эффективных алгоритмов решения прикладных задач; базовые структуры данных, основные алгоритмы сортировки и поиска и способы их эффективной реализации.</p> <p><u>Уметь:</u></p> <p>- формализовывать поставленную задачу, выбирать необходимые инструментальные средства для разработки программ в различных операционных системах и средах; составлять, тестировать, отлаживать и оформлять программы на языках высокого уровня, включая объектно-ориентированные;</p> <p><u>Владеть:</u></p>

		- навыками разработки программ на языке программирования высокого уровня; основными подходами к организации процесса разработки Web-приложений.
--	--	---

1.3. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Код компетенции	Этапы формирования компетенций				
	Модуль 1. Целые числа, ввод / вывод данных, простые операции со строками.	Модуль 2. Условный оператор и оператор цикла «while». Изучение логических выражений с целью использования ветвлений и циклов.	Модуль 3. Вещественные числа. Использование в программах вещественных чисел, используемых при решении реальных задач с учетом их специфики.	Модуль 4. Функции и рекурсия, создание и использование функций, позволяющих повторно использовать код и делать его более структурированным.	Модуль 5. Кортежи, оператор цикла «for», списки. Изучение коллекций элементов – кортежей и списков, а также возможностей цикла «for» для обработки элементов коллекций
ПК-2	+	+	+	+	+
Код компетенции	Этапы формирования компетенций (модулей дисциплины)				
	Модуль 6. Сортировка. Применение методов сортировки для решения прикладных задач.	Модуль 7. Множества и словари. Изучение структур данных – множеств и словарей. Сопоставление различных объектов в разнообразных прикладных задачах	Модуль 8. Функциональное программирование. Использование стандартных функций языка Python для обработки последовательностей.	Модуль 9. Применение языка Python для создания Web-приложений (Python-фреймворки, решение проблем выбора, практика Web – разработки).	Модуль 10. Работа с файлами, организация ввода/вывода информации с использованием файлов.
ПК-2	+	+	+	+	+

Раздел 2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование на языке Python» относится к части, формируемой участниками образовательных отношений блока 1 учебного плана по направлению подготовки 09.03.03 Прикладная информатика, профиль «Информационные системы в экономике».

Для успешного освоения данной дисциплины необходимы и востребованы знания, умения и навыки, полученные в рамках таких дисциплин, как - «ИТ и программирование», «Дискретная математика», «Web-программирование».

Знания, умения и навыки, полученные в результате изучения дисциплины, будут востребованы в процессе освоения таких дисциплин, как «Создание и продвижение web-сайта», «Информационные системы экономического анализа» и др. дисциплин, которые изучаются в рамках учебного плана направления подготовки «Прикладная информатика».

Раздел 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), на самостоятельную работу обучающихся и форму промежуточной аттестации

Объем дисциплины в зачетных единицах составляет 4 зачетные единицы.

Очная форма обучения

Количество академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), составляет 64 часа, в том числе:

- на занятия лекционного типа – 16 ч., - на занятия семинарского типа – 48 ч.

Количество академических часов, выделенных на самостоятельную работу обучающихся – 44 ч.

Форма промежуточной аттестации: экзамен, 36 час.

Очно-заочная форма обучения

Количество академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), составляет 16 часов, в том числе:

на занятия лекционного типа – 8 ч., на занятия семинарского типа – 24 ч.

Количество академических часов, выделенных на самостоятельную работу обучающихся – 76 ч.

Форма промежуточной аттестации: экзамен, 36 час.

Заочная форма обучения

Количество академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий), составляет 8 часа, в том числе:

на занятия лекционного типа – 2 ч., на занятия семинарского типа – 6 ч.

Количество академических часов, выделенных на самостоятельную работу обучающихся – 132 ч.

Форма промежуточной аттестации: экзамен, 4 час.

Раздел 4. Содержание дисциплины, структурированное по модулям с указанием отведенного на них количества академических часов и видов учебных занятий.

4.1. Очная форма обучения

№ п/п	Тема дисциплины	Всего академических часов	в т.ч. занятия лекционного типа	в т.ч. занятия семинарского типа:					Самостоятельная работа	Форма текущего контроля успеваемости.
				семинары	Практические занятия	Лабораторные занятия (лабораторные работы, лабораторный практикум)	Коллоквиумы	Иные занятия		
1.	Модуль 1. Целые числа, ввод / вывод данных, простые операции со строками.	9	1		1	3	-	-	4	проведение опроса; тестирование; решение задач на программирование
2.	Модуль 2. Условный оператор и оператор цикла «while». Изучение логических выражений с целью использования ветвлений и циклов.	10	1		2	3	-	-	4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
3.	Модуль 3. Вещественные числа. Использование в программах веществен-	11	2		2	3	-		4	проведение опроса; тестирование; решение задач на программирование

	ных чисел, используемых при решении реальных задач с учетом их специфики.									
4.	Модуль 4. Функции и рекурсия, создание и использование функций, позволяющих повторно использовать код и делать его более структурированным.	11	2	2	3	-	-	4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.	
5.	Модуль 5. Списки, кортежи, оператор цикла «for», списки. Изучение коллекций элементов – кортежей и списков, а также возможностей цикла «for» для обработки элементов коллекций	11	2	2	3	-	-	4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.	
6.	Модуль 6. Сортировка. Применение методов сортировки данных для решения прикладных задач.	10	1	2	3	-	-	4	проведение опроса; тестирование; решение задач на программирование;	

										выполнение практико-ориентированных задач на программирование.
7.	Модуль 7. Множества и словари. Изучение структур данных – множеств и словарей. Сопоставление различных объектов в разнообразных прикладных задачах	11	2		2	3	-	-	4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
8.	Модуль 8. Функциональное программирование. Использование стандартных функций языка Python для обработки последовательностей. Разные парадигмы программирования и сферы их применения.	16	2		1	5	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
9.	Модуль 9. Применение языка Python для создания Web - приложений (Python -	10	2		1	3	-	-	4	проведение опроса; тестирование; решение задач на программирование;

	фреймворки, решение проблем выбора, практика Web – разработки).									выполнение практико-ориентированных задач на программирование.
10.	Модуль 10. Работа с файлами, организация ввода/вывода информации с использованием файлов.	9	1		1	3			4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
Итого:		108	16	-	16	32	-	-	44	
Экзамен (групповая консультация в течение семестра, групповая консультация перед промежуточной аттестацией, экзамен)		36								контроль

4.2. Очно-заочная форма обучения

№ п/п	Тема дисциплины	Всего академических часов	в т.ч. занятия лекционного типа	в т.ч. занятия семинарского типа:					Самостоятельная работа	Форма текущего контроля успеваемости.
				семинары	Практические занятия	Лабораторные занятия (лабораторные работы, лабораторный практикум)	Коллоквиумы	Иные занятия		
1.	Модуль 1. Целые числа, ввод / вывод данных, простые операции со строками.	9	-		-	1	-	-	8	проведение опроса; тестирование; решение задач на программирование
2.	Модуль 2. Условный оператор и оператор цикла «while». Изучение логических выражений с целью использования ветвлений и циклов.	10	1		1	1	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
3.	Модуль 3. Вещественные числа. Использование в программах вещественных чисел, используемых при решении реальных задач с	11	1		1	2	-		8	проведение опроса; тестирование; решение задач на программирование

	учетом их специфики.									
4.	Модуль 4. Функции и рекурсия, создание и использование функций, позволяющих повторно использовать код и делать его более структурированным.	11	1		1	2	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
5.	Модуль 5. Списки, кортежи, оператор цикла «for», списки. Изучение коллекций элементов – кортежей и списков, а также возможностей цикла «for» для обработки элементов коллекций	11	1		1	2	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
6.	Модуль 6. Сортировка. Применение методов сортировки данных для решения прикладных задач.	10	1		1	1	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.

7.	Модуль 7. Множества и словари. Изучение структур данных – множеств и словарей. Сопоставление различных объектов в разнообразных прикладных задачах	11	1		1	2	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
8.	Модуль 8. Функциональное программирование. Использование стандартных функций языка Python для обработки последовательностей. Разные парадигмы программирования и сферы их применения.	16	1		1	3	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
9.	Модуль 9. Применение языка Python для создания Web - приложений (Python - фреймворки, решение проблем выбора, практика Web – разработки).	10	-		-	1	-	-	8	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.

10.	Модуль 10. Работа с файлами, организация ввода/вывода информации с использованием файлов.	9	1		1	1			4	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
Итого:		108	8	-	8	16	-	-	76	
Экзамен (групповая консультация в течение семестра, групповая консультация перед промежуточной аттестацией, экзамен)		36								контроль

4.3. Заочная форма обучения

№ п/п	Тема дисциплины	Всего академических часов	в т.ч. занятия лекционного типа	в т.ч. занятия семинарского типа:					Самостоятельная работа	Форма текущего контроля успеваемости.
				семинары	Практические занятия	Лабораторные занятия (лабораторные работы, лабораторный практикум)	Коллоквиумы	Иные занятия		
1.	Модуль 1. Целые числа, ввод / вывод данных, простые операции со строками.	13	-		-	-	-	-	13	проведение опроса; тестирование; решение задач на программирование
2.	Модуль 2. Условный оператор и оператор цикла «while». Изучение логических выражений с целью использования ветвлений и циклов.	15	1		-	1	-	-	13	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
3.	Модуль 3. Вещественные числа. Использование в программах вещественных чисел, используемых при решении	14	-		-	1	-		13	проведение опроса; тестирование; решение задач на программирование;

	реальных задач с учетом их специфики.									выполнение практико-ориентированных задач на программирование.
4.	Модуль 4. Функции и рекурсия, создание и использование функций, позволяющих повторно использовать код и делать его более структурированным.	15	1	-	1	-	-	13	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.	
5.	Модуль 5. Кортежи, оператор цикла «for», списки. Изучение коллекций элементов – кортежей и списков, а также возможностей цикла «for» для обработки элементов коллекций	13	-	-	-	-	-	13	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.	
6.	Модуль 6. Сортировка. Применение методов сортировки данных для решения прикладных задач.	13	-	-	-	-	-	13	проведение опроса; тестирование; решение задач на программирование;	

										выполнение практико-ориентированных задач на программирование.
7.	Модуль 7. Множества и словари. Изучение структур данных – множеств и словарей. Сопоставление различных объектов в разнообразных прикладных задачах	14	-		1	-	-	-	13	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
8.	Модуль 8. Функциональное программирование. Использование стандартных функций языка Python для обработки последовательностей. Разные парадигмы программирования и сферы их применения.	15	-		-	-	-	-	15	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
9.	Модуль 9. Применение языка Python для создания Web-приложений (Python-	14	-		-	1	-	-	13	проведение опроса; тестирование; решение задач на программирование;

	фреймворки, решение проблем выбора, практика Web – разработки).									выполнение практико-ориентированных задач на программирование.
10	Модуль 10. Работа с файлами, организация ввода/вывода информации с использованием файлов.	14	-		1	-			13	проведение опроса; тестирование; решение задач на программирование; выполнение практико-ориентированных задач на программирование.
Итого:		140	2		2	4	-	-	132	
Экзамен (групповая консультация в течение семестра, групповая консультация перед промежуточной аттестацией, экзамен)		4								контроль

**Раздел 5. Перечень основной и дополнительной учебной литературы,
необходимой для освоения дисциплины**

№ п/п	Автор	Название основ- ной и дополни- тельной учебной литературы, не- обходимой для освоения дисци- плины	Выходные данные	Количество экзем- пляров в библиотеке ДГУНХ/ адрес доступа
Основная учебная литература				
1.	Федоров Д. Ю.	Программирова- ние на Python: учебное пособие для вузов / Д. Ю. Федоров. — 6-е изд., перераб. и доп.	Москва: Издатель- ство Юрайт, 2025. — 187 с. — (Высшее образова- ние). — ISBN 978- 5-534- 19666-5. — Текст: электрон- ный // Об- разователь- ная плат- форма Юрайт	https://urait.ru/bcode/ 556864
2.	Чернышев С. А.	Основы про- граммирования на Python: учеб- ник для вузов / С. А. Чернышев. — 2-е изд., пере- раб. и доп.	Москва: Издатель- ство Юрайт, 2025. — 349 с. — (Высшее образова- ние). — ISBN 978- 5-534- 17139-6. — Текст: электрон-	https://urait.ru/bcode/ 567821

			ный // Образовательная платформа Юрайт	
3.	Северенс Ч.	Введение в программирование на Python.	Москва: НОУ «ИНТУИТ», 2016 - 231с.	http://biblioclub.ru/index.php?page=book&id=429184
4.	Хахаев И.А.	Практикум по алгоритмизации и программированию на Python: курс.	Москва: НОУ «ИНТУИТ», 2016 -179 с.	http://biblioclub.ru/index.php?page=book&id=429256
5.	Шелудько В.М.	Основы программирования на языке высокого уровня Python: учебное пособие / ФГАОУ ВО «Южный федеральный университет», Инженерно - технологическая академия.	Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2017 –147с.	http://biblioclub.ru/index.php?page=book&id=500056
6.	Шелудько В.М.	Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули / ФГАОУ ВО «Южный федеральный университет», Инженерно-технологическая академия.	Ростов-на-Дону; Таганрог: Издательство Южного федерального университета, 2017 – 108с.	http://biblioclub.ru/index.php?page=book&id=500060

Дополнительная литература				
а) Дополнительная учебная литература				
1.	Sweigart A.	Разработка компьютерных игр на языке Python.	М.: НОУ «ИН-ТУИТ», 2016. - 505с.	http://biblioclub.ru/index.php?page=book&id=429009
2.	Sweigart A.	Разработка компьютерных игр с помощью Python и Pygame.	М.: НОУ «ИН-ТУИТ», 2016.-290 с.	http://biblioclub.ru/index.php?page=book&id=429001
3.	Сузи Р.А.	Язык программирования Python: курс.	М.: Интернет - университет информационных технологий, 2007. - 327 с.	http://biblioclub.ru/index.php?page=book&id=233288

<i>б) Официальные издания: сборники законодательных актов, нормативно-правовых документов и кодексов РФ</i>
<ol style="list-style-type: none"> 1. Федеральный закон от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и о защите информации" (с изменениями и дополнениями). www.standartgost.ru 2. ГОСТ 34.320-96. Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы. 2001 г. www.standartgost.ru 3. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. 2008 г. www.standartgost.ru 4. ГОСТ Р ИСО/МЭК 17799-2005. Информационная технология. Практические правила управления информационной безопасностью. www.standartgost.ru 5. ГОСТ Р ИСО 11442-2014. Техническая документация на продукцию. Управление документацией. 2015 г. www.standartgost.ru 6. ГОСТ Р ИСО/МЭК 15288-2005. Информационная технология. Системная инженерия. Процессы жизненного цикла систем. 2006 г. www.standartgost.ru 7. ГОСТ Р ИСО/МЭК 12119-2000. Информационная технология. Пакеты программ. Требования к качеству и тестирование. 2005 г. www.standartgost.ru 8. ГОСТ Р ИСО/МЭК ТО 16326-2002. Программная инженерия. Руководство по применению ГОСТ Р ИСО/МЭК 12207 при управлении проектом.

2002 г. www.standartgost.ru

9. ГОСТ Р ИСО/МЭК 15910-2002. Информационная технология. Процесс создания документации пользователя программного средства. 2002 г. www.standartgost.ru
10. ГОСТ Р ИСО/МЭК ТО 12182-2002. Информационная технология. Классификация программных средств. 2002 г. www.standartgost.ru
11. ГОСТ Р ИСО/МЭК 15026-2002. Информационная технология. Уровни целостности систем и программных средств. 2002 г. www.standartgost.ru
12. ГОСТ Р ИСО/МЭК ТО 9294-93. Информационная технология. Руководство по управлению документированием программного обеспечения. www.standartgost.ru
13. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. 2009 г. www.standartgost.ru
14. ГОСТ 28195-89. Оценка качества программных средств. Общие положения. 2001 г. www.standartgost.ru
15. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы. www.standartgost.ru

в) Периодические издания

1. Научный журнал «Прикладная дискретная математика», <http://journals.tsu.ru/pdm>
2. Научный журнал «Информатика и ее применение», <http://www.ipiran.ru/journal/issues/>
3. Рецензируемый научный журнал «Прикладная информатика», <http://appliedinformatics.ru>

г) Справочно-библиографическая литература

1. Документация по Python 3.10. <https://www.python.org/>

Е) Информационные базы данных (профильные)

2. Реферативная база данных Web of Science. База данных по научному цитированию Web of Science Института научной информации. <http://isiknowledge.com/>
3. –Web-портал, содержащий необходимые дистрибутивы и полную информацию для языка программирования Python. <https://www.python.org/>
4. Политематическая реферативная база данных SCOPUS. <http://www.scopus.com/>
5. Сайт, посвященный свободно распространяемому пакету SymPy, представляющему собой библиотеку Python символьных вычислений. <http://github.com/sympy/sympy>
6. .Web-портал, созданный для студентов средних и высших учебных заведений, представляющий научно-информационный ресурс по криптографии и теории кодирования, а также по связанным с ними областями теоретической и прикладной математики, <http://gouspo.ru/>

Раздел 6. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины

Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к одной или нескольким электронно-библиотечным системам и к электронной информационно-образовательной среде университета (<http://e-dgunh.ru>). Электронно-библиотечные системы и электронная информационно-образовательная среда обеспечивают возможность доступа, обучающегося из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети "Интернет", как на территории университета, так и вне ее.

Для самостоятельного изучения материала и ознакомления с регламентирующими документами и текущей практикой в области программирования на языках высокого уровня, рекомендуется использовать следующие Интернет-ресурсы сайтов в области ИТ-технологий, и в частности, относящиеся к сфере разработки программного обеспечения:

1. www.intuit.ru – Сайт НОУ «ИНТУИТ».
2. stackoverflow.com/ - Сайт вопросов и ответов для программистов.
3. www.hse.ru – Сайт Высшей школы экономики.
4. <http://www.consultant.ru/> – онлайн-версия информационно-правовой системы "КонсультантПлюс"
5. www.standartgost.ru - Официальный портал Росстандарта.

Раздел 7. Перечень лицензионного программного обеспечения, информационных справочных систем и профессиональных баз данных

7.1. Необходимый комплект лицензионного программного обеспечения

1. Windows 10
2. Microsoft Office Professional
3. Adobe Acrobat Reader DC
4. VLC Media player
5. 7-zip
6. Microsoft Visual Studio
7. Python 10.

7.2. Перечень Интернет-сервисов и электронных ресурсов

1. <http://www.python.org/> – официальный сайт Python.
2. [http:// docs.python.org/](http://docs.python.org/) – официальный сайт Python, документация по языку программирования.
3. <http://www.consultant.ru/> – онлайн-версия информационно-правовой системы "КонсультантПлюс".

7.3. Перечень профессиональных баз данных:

- Научная электронная библиотека «e-library» (<https://elibrary.ru>);

- Библиотеки для создания графического интерфейса -<http://qt.nokia.com>,
<http://wxpytho.org>, <http://pyside.org>, <http://pygtk.org>,
<http://pyfltk.sourceforge.net>, <http://sourceforge.net/projects/pywin32/>;
- Открытая база ГОСТов - <http://Standartgost.ru>.
- Единый реестр российских программ для электронных вычислительных машин и баз данных (<https://reestr.minsvyaz.ru/reestr/>).

Раздел 8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для осуществления образовательного процесса по дисциплине необходима следующая *материально-техническая база*:

Учебная аудитория для проведения занятий лекционного типа и занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Специализированная мебель, мультимедиапроектор, проекционный экран, ноутбук, средства звуковоспроизведения, интерактивная доска, выход в сеть Интернет. Наборы демонстрационного оборудования и учебных наглядных пособий, обеспечивающие тематические иллюстрации.

Кабинет информатики №1. Специализированная мебель. 15 автоматизированных рабочих мест, оснащенных лицензионным программным обеспечением: Windows 7, Microsoft Office 2010, Компьютерная справочная правовая система (КСПС) КонсультантПлюс

Учебная аудитория для самостоятельной работы обучающихся, 6 автоматизированных рабочих мест с выходом в сеть Интернет и доступом в электронную информационно-образовательную среду, 3 принтера.

Раздел 9. Образовательные технологии

В ходе освоения дисциплины «Программирование на языке Python» целесообразно в рамках образовательной деятельности комплексно применять спектр различных организационных форм с использованием различных методов обучения, преподавания и оценивания, направленный на достижение результатов и формирование на их основе запланированной компетенции.

При проведении учебных занятий по данной дисциплине необходимо обеспечить развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия результативных решений. Целесообразно проведение разборов постановок задач, изучение и анализ алгоритмов, часто применяемых для решения различных прикладных задач, большое внимание уделять СРС, работе с учебной и справочной литературой по языку программирования Python.

Необходимо использовать при этом возможности электронной информационно-образовательной среды вуза и информационные ресурсы глобальной сети Интернет, а также программные продукты различных фирм и компаний (в частности, материалы официального сайта www.python.org), ресурсы ЭБС. Инновационными в настоящее время признаны подходы, основанные на применении интерактивных методов обучения, которые позволяют решать следующие задачи:

- мотивация обучающихся;
- эффективное усвоение учебного материала;
- самостоятельный поиск обучающимися путей и вариантов решения поставленной прикладной задачи;
- работа в команде, формирование жизненных и профессиональных навыков, выход на уровень осознанной компетентности обучающегося.



**Образовательная автономная некоммерческая организация
высшего образования**

«Дагестанский гуманитарный институт»

Адрес: 367000, Республика Дагестан, г. Махачкала, ул. Дахадаева, 136; ОГРН 1180571012203; ИНН 0572022227 /
КПП 057201001, тел. +7 (8722) 94-00-60, e-mail: mail@daggum.ru, контактное лицо: Магомедова З.Р.

Утверждаю

Декан гуманитарно-экономического
факультета
_____ У.Д.Давлетмурзаева

« 29 » января 2026 г.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

ПО ДИСЦИПЛИНЕ

«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»

**НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 ПРИКЛАДНАЯ
ИНФОРМАТИКА, ПРОФИЛЬ «ИНФОРМАЦИОННЫЕ
СИСТЕМЫ В ЭКОНОМИКЕ»**

Уровень высшего образования - бакалавриат

Махачкала – 2026

Рабочая программа дисциплины «Программирование на языке Python» разработана в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г., № 922, в соответствии с приказом Министерства образования и науки Российской Федерации от 06.04.2021 г. № 245 «Об утверждении Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры».

Утверждена на Ученом совете ОАНО ВО "ДГИ" 29.01.2026 года
протокол №6

СОДЕРЖАНИЕ

Назначение оценочных материалов.....	4
РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины	5
1.1 Перечень формируемых компетенций.....	5
1.2 Перечень компетенций с указанием индикаторов и видов оценочных средств	5
РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине.....	9
РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания	33
РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций.....	36
Лист актуализации оценочных материалов по дисциплине	41

Назначение оценочных материалов

Оценочные материалы для текущего контроля успеваемости (оценивания хода освоения дисциплин), для проведения промежуточной аттестации (оценивания промежуточных и окончательных результатов обучения по дисциплине) обучающихся по дисциплине «Программирование на языке Python» на соответствие их учебных достижений поэтапным требованиям образовательной программы высшего образования 09.03.03 Прикладная информатика, профиль «Информационные системы в экономике».

Оценочные материалы по дисциплине «Программирование на языке Python» включают в себя: перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины О; описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания; типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения ОПОП; методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Оценочные материалы сформированы на основе ключевых принципов оценивания:

- валидности: объекты оценки должны соответствовать поставленным целям обучения;
- надежности: использование единообразных стандартов и критериев для оценивания достижений;
- объективности: разные обучающиеся должны иметь равные возможности для достижения успеха.

Основными параметрами и свойствами оценочных материалов являются:

- предметная направленность (соответствие предмету изучения конкретной дисциплины);
- содержание (состав и взаимосвязь структурных единиц, образующих содержание теоретической и практической составляющих дисциплины);
- объем (количественный состав оценочных материалов);
- качество оценочных материалов в целом, обеспечивающее получение объективных и достоверных результатов при проведении контроля с различными целями.

**РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств
в процессе освоения дисциплины**

1.1 Перечень формируемых компетенций

Код компетенции	Формулировка компетенции
ПК	ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ
ПК-2	Способен разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем

1.2. Перечень компетенций с указанием индикаторов и видов оценочных средств

<i>Формируемые компетенции</i>	<i>Код и наименование индикатора достижения компетенции</i>	<i>Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций</i>	<i>Уровни освоения компетенций</i>	<i>Критерии оценивания сформированности компетенций</i>	<i>Виды оценочных средств</i>
ПК-2. Способен разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем	ИПК-2.1. Разрабатывает и внедряет прикладное программное обеспечение и web-сервисы информационных систем	Знать: - современные средства разработки программ на языках высшего уровня, методы программирования и методы разработки эффективных алгоритмов решения прикладных задач; базовые структуры дан-	Пороговый уровень	Демонстрирует частичное понимание используемых средств разработки программ, методов программирования и использования требуемых алгоритмов решения прикладных задач, базовых структур данных и базовых алгоритмов сортировки и поиска и способов их эффективной реализации. Большинство требований, предъявляемых, к заданию выполнены. Имеет место владение элементами заданного материала. В основном выполненный материал понятен и носит целостный характер.	Блок А – задания репродуктивного уровня: – тестовые задания; – вопросы для обсуждения
			Базовый уровень	Демонстрирует значительное понимание задач, обозначенных дисциплиной.	

<i>Формируемые компетенции</i>	<i>Код и наименование индикатора достижения компетенции</i>	<i>Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций</i>	<i>Уровни освоения компетенций</i>	<i>Критерии оценивания сформированности компетенций</i>	<i>Виды оценочных средств</i>
		ных, основных алгоритмы сортировки и поиска и способы их эффективной реализации.		Все требования, предъявляемые к заданиям выполнены. Содержание выполненных заданий раскрыто и рассмотрено с разных точек зрения. Демонстрирует на хорошем уровне применение сред разработки программ, эффективных методов программирования, основанных на применении выбранных алгоритмов решения прикладных задач. Успешно использует имеющиеся структуры данных, алгоритмы сортировки и поиска при решении поставленных задач.	
			Продвинутый уровень	Демонстрирует полное понимание поставленных задач, все требования, предъявляемые к заданиям выполнены. Продемонстрировано уверенное владение материалом дисциплины. Выполненные задания носят целостных характер, выполнены в полном объеме, структурированы, имеет место творческий подход.	
		Уметь: - формализовывать поставленную задачу, выбирать необходимые инструментальные средства для	Пороговый уровень	Частично осуществляет выбор требуемых инструментальных сред программирования при написании программ под управлением операционных систем. Умеет формализовывать поставленную задачу, составлять программы среднего уровня сложности на языке про-	Блок В – задания реконструктивного уровня – тестовые задания; – задачи на

<i>Формируемые компетенции</i>	<i>Код и наименование индикатора достижения компетенции</i>	<i>Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций</i>	<i>Уровни освоения компетенций</i>	<i>Критерии оценивания сформированности компетенций</i>	<i>Виды оценочных средств</i>
		разработки программ в различных операционных системах и средах; составлять, тестировать, отлаживать и оформлять программы на языках высокого уровня, включая объектно-ориентированные.		граммирования Python, тестировать и отлаживать выполненные задания на программирование, оформлять пояснительную документацию к ним.	программирование.
			Базовый уровень	Владеет базовыми умениями в области формализации поставленных задач и последующей реализации с использованием встроенного функционала языка программирования Python. Владеет уверенными навыками составления, тестирования, отладки и оформления программ, включая объектно-ориентированные.	
			Продвинутый уровень	Демонстрирует умения, позволяющие решать на уровне постановки и реализации задач достаточно сложные, квазипрофессиональные задачи по программированию. Полученные умения позволяют целостно увязывать возможности языка программирования с поставленными задачами, которые решаются в полном объеме и в заданные сроки, структурированы и логически увязаны. Показаны альтернативные варианты написания программ.	
		Владеть: - навыками разработки программ на языке про-	Пороговый уровень	В определенной степени владеет навыками программирования, при этом имеют место частичные недоработки и ошибки в плане выбора подхода и организации процесса	Блок С – задания практико-ориентированного

<i>Формируемые компетенции</i>	<i>Код и наименование индикатора достижения компетенции</i>	<i>Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций</i>	<i>Уровни освоения компетенций</i>	<i>Критерии оценивания сформированности компетенций</i>	<i>Виды оценочных средств</i>
		граммирования высокого уровня; основными подходами к организации процесса разработки Web-приложений.		разработки программного обеспечения.	уровня: – практико-ориентированные задачи на программирование.
	Базовый уровень		Демонстрирует навыки решения типовых задач с использованием возможностей языка программирования, владеет основными подходами к организации процесса разработки программного обеспечения. Все требования, предъявляемые к заданию, выполняются, проработана возможность стабильной работы программного кода.		
	Продвинутый уровень		Демонстрирует уверенные навыки в ходе программирования и апробации предложенных практических задач повышенной сложности, включая нестандартные задачи.		

РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине

Для проверки сформированности компетенции

ПК-2: Способен разрабатывать, адаптировать, тестировать и внедрять прикладное программное обеспечение информационных систем.

ИПК-2.1. Разрабатывает и внедряет прикладное программное обеспечение и Web-сервисы информационных систем.

Блок А. Задания репродуктивного уровня («знать»)

А.1 Фонд тестовых заданий по дисциплине

Тесты типа А.

1. Выберите правильные с Вашей точки зрения, утверждения по синтаксису языка Питон:

- а) Конец строки является концом инструкции;
- б) Вложенные инструкции объединяются в блоки по величине отступов;
- в) Отступ может быть не одинаковым в пределах одного вложенного блока;
- г) Основная инструкция языка программирования завершается двоеточием, вслед за которым располагается вложенный блок кода, обычно с отступом под строкой основной инструкции.

2. Выберите правильные специальные случаи, возможные для инструкций языка Питон:

- д) Допустимо записывать одну инструкцию в нескольких строках, для чего можно ее заключить в пару круглых, квадратных или фигурных скобок;
- е) Возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой
- ж) Тело составной инструкции не может располагаться в той же строке, что и тело основной, если тело составной инструкции не содержит составных инструкций;
- з) Комментарии к программе на языке Питон начинаются с знака #.

3. Установите соответствие, в качестве чего выступает передаваемый функции печати параметр:

а) <code>print('Нет – корона вирусу!')</code>	б) Арифметическое выражение
в) <code>print(2 + 3)</code>	г) Строка
д) <code>print('2' + '3')</code>	е) Конкатенация двух строк

4. К функции печати `print ()` параметр `sep=''` означает:

- а) интегратор;
- б) разделитель;
- в) кодификатор;
- г) идентификатор.

5. Установите приоритет выполнения арифметических операций, операций сравнения и логических операций (установите соответствие) в языке Питон:

а) арифметические операции	2
б) операции сравнения	3
в) логические операции	1

6. Допустимы ли в языке Питон логические выражения, содержащие несколько знаков сравнения, например `x < y < z`:

- а. да;
- б. нет;

7. Применение именованных параметров `end=''` и `sep=''` в рамках функции печати при выполнении операторов

```
1 print (1, 2, 3, 4, sep=' + ', end="")
2 print (' = ', 1 + 2 + 3 + 4, sep="")
```

приведет к выводу следующей строки _____:(введите)

8. Переменные логического типа, как и числа и строки, являются _____ объектами.

9. Приведенный ниже оператор проверяет, является ли число _____.

```
isEven = number % 2 == 0:
```

Впишите пропущенное слово.

8. Для решения какой задачи может быть использован следующая инструкция языка Питон \rightarrow `answer = start1 <= finish2 and start2 <= finish1`:

- а) задачу о интеграции двух длительных по времени событий;
- б) задачу о пересечении двух длительных по времени событий;
- в) задачу о зависимости одного события от другого;
- г) задачу о объединении двух длительных по времени событий.

9. Какую задачу решает следующий блок команд:

```
1 x = int(input ())
2 if x >= 0:
3     print(x)
4 else:
5     print(-x)
```

- а) Изменяет знак переменной x;
- б) Находит модуль числа x;
- в) Ничего не делает;
- г) Осуществляет ввод только положительных чисел.

10. Для чего используется специальная конструкция `elif`, обозначающая "иначе, если", после которой записывается условие, в языке Питон:

- а) Для расширения функциональности условного оператора;
- б) Для улучшения читаемости программ;
- в) Для того, чтобы не пользоваться ветвью `else`;
- г) Для уменьшения отступа в тексте программы.

11. Оператор цикла `While` позволяет:

- а) выполнять команды, до тех пор, пока условие верно;
- б) организовать «вечный» цикл;
- в) не позволяет повторять блок команд в теле цикла;
- г) Не требует следить за тем, чтобы в блоке команд происходили изменения, которые приведут к тому, что в какой-то момент условие перестанет быть истинным.

12. Какую задачу решает следующий блок команд:

```
1 n = int(input())
2 i = 1
3 while i <= n:
4     print(i)
5     i = i + 1
```

- а) выводит все числа по возрастанию от 1 до N-1;
- б) выводит числа, меньшие N;
- в) демонстрирует неверное использование оператора `while`;
- г) выводит все числа по возрастанию от 1 до N.

13. Какую задачу решает следующий блок команд:

```
1   now = int(input())
2   nowMin = now
3   while now != 0:
4       if now < nowMin:
5           nowMin = now
6       now = int(input())
7   print(nowMin)
```

- а) выводит все индексы минимальных чисел, если их несколько;
- б) максимальное число введенной последовательности;
- в) находит минимальное число в последовательности с использованием оператора while, заканчивающееся 0;
- г) выводит все числа, не равные 0.

14. Можно ли в языке Питон к циклу while приписать блок else:

- а) да;
- б) нет.

15. Команды в блоке else оператора цикла while будут выполняться, если цикл завершил свою работу нормальным образом (условие в какой-то момент перестало быть истинным) и не будут выполняться только в случае, если выход из цикла произошел с помощью команды _____.

Впишите пропущенный оператор.

16. Какую задачу решает следующий блок команд:

```
1   now = int(input())
2   seqSum = 0
3   while now != 0:
4       seqSum = seqSum + now
5       now = int(input())
6   print(seqSum)
```

- а) выводит все индексы введенных чисел;
- б) подсчитывает сумму введенной последовательности;
- в) находит произведение чисел введенных чисел последовательности с использованием оператора while, заканчивающееся 0;
- г) выводит все числа, не равные 0.

17. Какую задачу решает следующий блок команд:

```
1     now = -1
2     while now != 0:
3         now = int(input())
4         if now <= 0:
5             continue
6         print(now)
```

- а) выводит все отрицательные введенные числа;
- б) выводит все положительные числа введенной последовательности;
- в) приводит пример использования команды `continue`;
- г) выводит все числа, не равные -1.

18. Любое вещественное число на языке Питон представимо в виде дроби, где в числителе хранится целое число, а в знаменателе находится какая-либо степень _____. Впишите пропущенное слово.

19. Для преобразования вещественных чисел в целые можно использовать несколько видов функций округления, установите соответствие этих функций решаемым задачам:

1. int	а). Округляет в сторону нуля (отбрасывает дробную часть)
2. round	б). Округляет до ближайшего целого, если ближайших целых несколько (дробная часть равно 0.5), то к чётному
3. floor	в). Округляет в меньшую сторону
4. ceil	г). Округляет в большую сторону

20. Выберите правильные утверждения по теме – «Срезы строк»:

- а) Для определения длины строки `s` можно воспользоваться функцией `len(s)`;
- б) Срез - это способ извлечь из строки отдельные символы или подстроки;
- в) Чтобы получить `i`-ый символ строки нужно написать `s[i]`;
- г) Верны все варианты.

21. Какую задачу решает следующий блок команд:

```
1     string = input ()
2     substring = input ()
3     pos = string.find (substring)
```

```

4   while pos != -1:
5       print(pos)
6       pos = string.find (substring, pos + 1)

```

- а) находит и выводит все вхождения подстроки в строку;
- б) выводит все вхождения подстроки, начиная с конца строки;
- в) находит и выводит все вхождения подстроки в строку, включая накладывающиеся;
- г) находит и выводит все вхождения подстроки в строку, исключая накладывающиеся.

22. Выберите правильные утверждения по теме – «Функции»:

- а) len(s) и print(x) – являются функциями;
- б) Функции - части программы, которые можно повторно вызывать с разными параметрами;
- в) Описание функции в Питоне начинается со служебного слова Function;
- г) Некоторые логические блоки программы иногда удобно оформлять в виде функции.

23. Какую задачу решает следующий блок команд:

```

1   def f(m):
2       t = 1
3       i = 2
4       while i <= m:
5           t *= i
6           i += 1
7       return t
8   n = int(input())
9   print(f(n))

```

- а) находит и выводит введенных n чисел;
- б) выводит сумму введенных чисел;
- в) вычисляет значение n! ;
- г) находит и выводит все вхождения подстроки в строку, исключая накладывающиеся.

24. Пусть задается последовательность натуральных чисел, заканчивающаяся нулем. Какой результат выдаст приведенный набор команд языка Питон:

```

1   def rec ():
2       n = int (input ())
3       if n != 0:
4           rec ()
5       print(n)
6   rec()

```

- а) выводит список введенных n чисел в том же порядке;
 - б) выводит числа, не равные 0;
 - в) выводит список введенных n чисел в обратном порядке;
 - г) выводит только целые неотрицательные числа.
25. Выберите правильные утверждения по теме – «Кортежи»:
- а) функции `len()`, `print()`, `str()` - применимы к кортежам;
 - б) функция `tuple` применительно к строке не позволяет сформировать кортеж;
 - в) в кортеже могут храниться элементы различных типов;
 - г) к кортежу нельзя обращаться к элементам по индексу.
26. Выберите правильные утверждения по теме – «Кортежи»:
- а) операция распаковки не применима к кортежам;
 - б) кортеж – изменяемый тип данных;
 - в) для кортежей можно применять операцию «срез»;
 - г) кортеж, содержащий в себе один элемент, называется «синглтон».
27. Выберите правильные утверждения по теме – «Цикл `for`»:
- а) работа цикла `for` может быть прервана с помощью команды `break` или может быть осуществлен переход к следующей итерации с помощью `continue`;
 - б) цикл `for` позволяет поочередно перебрать элементы из чего-нибудь итерируемого (`iterable` или `tuple`);
 - в) внутри оператора цикла `for` может быть расположен и другой `for`;
 - г) все действия, которые должны выполняться в `for`, не должны выделяться отступом».
28. Какой результат выдаст приведенный ниже набор команд языка Питон:
- ```
1 for i in range(1, 11):
2 for j in range(1, 11):
3 print(i * j, end=' ')
4 print()
```
- а) выводит значения произведения двух случайных чисел в диапазоне от 1 до 11;
  - б) выводит таблицу сложения;
  - в) выводит таблицу умножения;
  - г) демонстрирует возможность составления программ с вложенными операторами цикла `for`.

29. Выберите правильные утверждения по теме – «Списки»:

- а) список – это неизменяемый тип данных в языке Питон;
- б) список в Питоне является аналогом массивов в других языках программирования
- в) список - это набор ссылок на объекты;
- г) константные списки записываются в квадратных скобках.

30. В результате выполнения следующего программного кода:

```
1 myList = [1, 2, 3]
2 myList[1] = 4
3 print(myList)
```

будет напечатано \_\_\_\_\_. (введите результат в виде списка).

## **A2. Вопросы для обсуждения (применительно к языку Python)**

1. Ввод и вывод целых чисел
2. Работа с вещественными числами
3. Форматный ввод и вывод данных
4. Функционал условного оператора
5. Специфика работы с операторами цикла в языке Питон
6. Работа с файлами.
7. Работа с функциями и методами при обработке файлов.

## **Блок В – задания реконструктивного уровня**

### **Тестовые задания типа В.**

1. Что выведет следующий код:

```
print(list(map(len, ['Амина', 'Магомед', 'Анастасия', 'Корона_вирус'])))
```

Приведите ответ в виде списка языка Питон.

2. Для чего удобно использовать приведенную ниже конструкцию:

```
numList = list(map(int, input().split())):
```

- а) для ввода строк неограниченной длины;
- б) для ввода логических значений;
- в) для считывания списка чисел, записанных в одну строку и разделенных пробелом;
- г) все ответы верны.

3. Что будет выведено в результате работы следующей программы при вводе следующей последовательности чисел – 1 2 3 4 5 6 7 8 9 12 14 16 18 33:

```
1 numbers = list(map(int, input().split()))
2 newList = []
```

```

3 for i in range(len(numbers)):
4 if numbers[i] % 2 == 0:
5 newList.append(numbers[i])
6 print(' '.join(map(str, newList)))

```

Результат выведите в виде последовательности чисел, разделенных пробелом

4. Можно ли применить приведенный ниже в программе подход сортировки списка с использованием метода `sort()` к кортежам, строкам и данным типа `iterable`:

```

1 myList = list(map(int, input().split()))
2 myList.sort()
3 print(' '.join(map(str, myList)))

```

- а) Можно;
- б) Нельзя.

5. Изменяет ли функция `sorted()` переданный ей в качестве аргумента список в рамках нижеприведенного кода программы:

```

1 myList = list (map(int, input().split()))
2 sortedList = sorted(myList)
3 print (' '.join(map(str, sortedList)))

```

- а) Изменяет;
- б) Не изменяет.

6. Выберите правильные утверждения:

- а) функция `sorted` может принимать в качестве параметра список;
- б) функция `sorted` может принимать в качестве параметра кортеж;
- в) функция `sorted` может принимать в качестве параметра `iterable`;
- г) функция `sorted` может принимать в качестве параметра строку.

7. Какой список выведет на экран ПК команда:

```
print(sorted(range(10, -1, -2)))
```

8. Что выдаст следующая команда на экран ПК:

```
print(' '.join(map(lambda x: str(x**2), range(5, 33))))
```

- а) список квадратов всех чисел от 5 до 33;
- б) список строк, возвращаемых лямбда функцией;
- в) результат работы лямбда функции с списком в интервале от 5 до 33 с шагом 2;
- г) выдаст ошибку.

9. Что будет выведено в результате работы следующей программы:

```

1 inFile = open('input.txt', 'r', encoding='utf8')
2 outFile = open('output.txt', 'w', encoding='utf8')
3 lines = inFile.readlines()

```

```

4 for line in lines:
5 print(line[-2::-1], file=outFile)
6 inFile.close()
7 outFile.close()

```

- а) исходный файл, перезаписан в выходной файл без изменения;
- б) ничего;
- в) срез исходного файла;
- г) выходной файл, в котором строки исходного файла развернуты в обратную сторону.

10. Какой вид сортировки реализован в рамках приведенной ниже функции.

```

def sort(array):
 for i in range(len(array)):
 indxMin = i
 for j in range(i+1, len(array)):
 if array[j] < array[indxMin]:
 indxMin = j
 tmp = array[indxMin]
 array[indxMin] = array[i]
 array[i] = tmp
 return array

```

Введите ответ, как продолжение предложения. Это сортировка \_\_\_\_\_.

11. Какой вид сортировки реализован в рамках приведенной ниже функции:

```

def sort(array):
 for i in range(len(array)):
 v = array[i]
 j = i
 while (array[j-1] > v) and (j > 0):
 array[j] = array[j-1]
 j = j - 1
 array[j] = v
 print array
 return array

```

Введите ответ, как продолжение предложения. Это сортировка \_\_\_\_\_.

12. Какой вид сортировки реализован в рамках приведенной ниже функции

```
def sort (array):
 a = array
 for i in range(len(a), 0, -1):
 for j in range(1, i):
 if a[j-1] > a[j]:
 tmp = a[j-1]
 a[j-1] = a[j]
 a[j] = tmp
 print a
 return a
```

Введите ответ, как продолжение предложения. Это \_\_\_\_\_ сортировка.

13. Какую задачу решает приведенная ниже программа:

```
marks = map(int, input().split())
cntMarks = [0] * 11
for mark in marks:
 cntMarks[mark] += 1
for nowMark in range(11):
 print((str(nowMark) + ' ') * cntMarks[nowMark], end="")
```

- а) вводит и выводит введенные баллы по 11 дисциплинам
- б) подсчитывает количество различных баллов по сданным дисциплинам
- в) подсчитывает, сколько раз встречалась каждая из оценок и выводит её столько раз
- г) подсчитывает, сколько раз встречалась каждая из оценок и выводит её столько раз в порядке возрастания

14. Какие операции доступны при работе с множествами, выберите правильные утверждения:

- а) в множество можно добавлять и удалять объекты, проверять принадлежность объекта множества и перебирать все объекты множества;
- б) над множествами можно совершать такие групповые операции, как пересечение и объединение множеств;
- в) множество – это всегда упорядоченный набор данных;
- г) множество – изменяемый объект, и не может быть элементом другого общества.

15. Существуют ли неизменяемые множества:

- а) да;
- б) нет.

16. Как называют неизменяемые множества. Введите используемый для этого в языке Питоне термин.

17. Какие из утверждений правильны с Вашей точки зрения для множеств:
- а) число элементов в множестве можно узнать с помощью функции `len`;
  - б) применение функции `str` к множеству даст нам текстовое представление;
  - в) из множества можно сделать список или кортеж с помощью функций `list` и `tuple` соответственно;
  - г) к множеству можно применять функцию `map`.
18. Какие из утверждений правильны с Вашей точки зрения для множеств:
- а) для проверки, входит ли элемент  $X$  в множество  $A$  можно использовать конструкцию - `if X in A`;
  - б) чтобы проверить, что элемент  $X$  не является элементом множества допустима запись - `not X in A`;
  - в) чтобы проверить, что элемент  $X$  не является элементом множества допустима запись - `X not in A`;
  - г) для удаления элемента из множества нельзя пользоваться методами - `discard` и `remove`.
19. Установите соответствие между операциями с множествами, которые создают новые множества и их описанием:

| Операция     | Описание                                                       |
|--------------|----------------------------------------------------------------|
| $A \cup B$   | Объединение множеств                                           |
| $A \& B$     | Множество, элементы которого входят в $A$ , но не входят в $B$ |
| $A - B$      | Элементы входят в $A \cup B$ , но не входят в $A \& B$         |
| $A \wedge B$ | Пересечение множеств                                           |

20. Установите соответствие операций сравнения с их описаниями:

| Операция | Описание                      |
|----------|-------------------------------|
| $A == B$ | Все элементы совпадают        |
| $A != B$ | Есть различные элементы       |
| $A <= B$ | Все элементы $A$ входят в $B$ |
| $A < B$  | $A <= B$ и $A != B$           |
| $A >= B$ | Все элементы $B$ входят в $A$ |
| $A > B$  | $B <= A$ и $B != A$           |

21. Для операций  $A \cup B$ ,  $A \& B$ ,  $A - B$ ,  $A \wedge B$  создается новое множество, однако для них в Питоне определены и соответствующие сокращенные записи: `|`, `=`, `&`, `-`, `^`. Что изменяют на выходе такие сокращенные операции над множествами:

- а) такие операции изменяют множество, находящееся справа от знака операции;
- б) такие операции генерируют одно новое множество;

- в) такие операции модифицируют оба множества, участвующие в этой операции;
- г) такие операции изменяют множество, находящееся слева от знака операции.

22. Какие из приведенных ниже утверждений правильны для словарей в языке Питон:

- а) словарь можно конструировать из других объектов с помощью функции `dict`;
- б) удаление элемента из словаря осуществляется специальной командой `del <имя_словаря [ключ]>`;
- в) пустой словарь можно создать, набрав пустые фигурные скобки;
- г) все правильные.

23. Какую задачу решает следующая программа на языке Питон:

```
1 seq = map(int, input().split())
2 countDict = { }
3 for elem in seq:
4 countDict[elem] = countDict.get(elem, 0) + 1
5 for key in sorted(countDict):
6 print(key, countDict[key], sep=' : ')
```

- а) подсчитывает сколько различных чисел было в рамках введенной последовательности;
- б) подсчитывает сколько раз входит в последовательность каждое из введенных чисел;
- в) формирует словарь из элементов введенной числовой последовательности;
- г) формирует неотсортированный частотный словарь.

24. Определитель с примененным алгоритмом решения задачи и определите результат работы программы по данному алгоритму:

```
from functools import reduce
elts = [1,2,3,4,5]
res = reduce(lambda x, y: x + y, elts)
print (res)
```

Введите ответ решения задачи (цифрами).

25. С учетом примененного алгоритма решения задачи определите результат работы программы:

```
from functools import reduce
nms = [111, 33, 777, 55, 999, 22, 666, 44, 888]
result = reduce(lambda a,b: a if (a > b) else b, nms)
print (result)
```

Введите ответ решения задачи (цифрами).

26. Определитель с примененным алгоритмом решения задачи и определите результат работы программы по данному алгоритму:

```
from functools import reduce
list1 = [11, 22, 33, 44, 55, 66, 77, 88, 99]
list2 = [9, 8, 7, 6, 5, 4, 3, 2, 1]
new_res = list(map(lambda x,y: x // y, list1, list2))
res = reduce(lambda x,y: x + y, new_res)
print (new_res)
print(res)
```

Введите ответ решения задачи в формате: список, число.

27. Какой алгоритм реализован в рамках приведенной ниже программы:

```
from random import randint
a = []
for i in range(15):
 a.append(randint(1, 50))
a.sort()
print(a)
value = int(input())
mid = len(a) // 2
low = 0
high = len(a) - 1
while a[mid] != value and low <= high:
 if value > a[mid]:
 low = mid + 1
 else:
 high = mid - 1
 mid = (low + high) // 2
if low > high:
 print("No value")
else:
 print("ID =", mid)
```

- а) алгоритм сортировки списка, введенного с клавиатуры;
- б) алгоритм поиска и вывода заданного числа с указанием его индекса;
- в) алгоритм бинарного поиска в списке заданного числа с выводом его местоположения;
- г) алгоритм одного из вариантов сортировки списка.

28. Что вычисляет приведенная ниже программа:

```
n = int(input())
f = 1
while n > 1:
 f *= n
 n -= 1
print(f)
```

Ответ введите одним словом в именительном падеже.

29. Что вычисляет приведенная ниже программа:

```
n = int(input())
f = 1
for i in range(2, n+1):
 f *= i
print(f)
```

Ответ введите одним словом в именительном падеже.

30. Что вычисляет приведенная ниже программа:

```
def f(n):
 if n == 0:
 return 1
 return f(n-1) * n
print(f(5))
```

Ответ введите одним словом в именительном падеже.

## **В1. Задачи**

1. Составьте три варианта программы вычисления факториала с помощью операторов цикла `while`, `for` и с помощью рекурсии.
2. Составьте программу, которая реализует двоичный или бинарный поиск заданного значения в списке для упорядоченной последовательности (отсортированных по возрастанию или убыванию) и определяет, содержит ли список искомое значение, и указывает его индекс (в случае наличия).
3. Составьте программу, которая проверяет простоту числа перебором делителей (Алгоритм перебора делителей заключается в последовательном делении заданного натурального числа на все целые числа, начиная с двойки и заканчивая значением меньшим или равным квадратному корню из тестируемого числа. Если хотя бы один делитель делит исследуемое число без остатка, то это число является составным. Если ни одного такого делителя не находится, то число признается простым).

4. Составьте программу вычисления чисел Фибоначчи тремя способами – с помощью цикла (for, while) и с помощью рекурсии.
5. Составьте программу перевода из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.
6. Опишите алгоритм решения, приведенный в приведенной ниже программе:

```
numbers = list(map(int, input().split()))
i = 0
while i < len(numbers):
 if numbers[i] % 2 != 0:
 numbers.pop(i)
 else:
 i += 1
print(' '.join(map(str, numbers)))
```

7. Опишите алгоритм решения, приведенный в приведенной ниже программе:

```
a = list(map(int, input().split()))
x = int(input())
a.append(x)
a.sort(reverse=True)
k = a.count(x)
print(a.index(x) + k)
```

8. Какую задачу решает приведенный ниже оператор:  
`print(len(set(open('input.txt', 'r', encoding = 'utf8').read().split())))`

9. Какую задачу решает приведенная ниже программа:

```
import random
a = []
for i in range(20):
 n = round(random.random() * 100)
 a.append(n)
print("A =", a)
b = []
i = 0
while i < len(a):
 if 35 < a[i] < 65:
 b.append(a[i])
 del a[i]
 else:
```

```
 i += 1
print("A =",a)
print("B =",b)
```

10. Решите задачу формирования списка, являющегося пересечением двух произвольно заданных в программе списков, введенных из файлов.

11. Составьте блок-схему алгоритма, задействованного в нижеприведенной программе. Какую задачу решает приведенная ниже программа:

```
s = input()
l = len(s)
integ = []
i = 0
while i < l:
 s_int = ""
 a = s[i]
 while '0' <= a <= '9':
 s_int += a
 i += 1
 if i < l:
 a = s[i]
 else:
 break
 i += 1
 if s_int != "":
 integ.append(int(s_int))
print(integ)
```

12. В рамках одной программы решите следующие задачи на обработку строк:

- Вывести длину самого короткого слова в строке;
- Рассчитать процент строчных и прописных букв в строке;
- Отсортировать слова в строке по возрастанию их длины.

13. Определитесь с алгоритмом, приведенным в нижеприведенной программе. Какие задачи решаются в этой программе?

```
class Hand:
 def __init__(self, *cards):
 self.cards = list(cards)
 def get_card(self, n):
 return self.cards[n]
 def show_all(self):
```

```

 for i in range(len(self.cards)):
 print(i, '-', self.cards[i])
 your_hand = Hand('A', '10', 'K', '6', '8')
 print(your_hand.get_card(2))
 your_hand.show_all()

```

14. В программе решите следующий спектр задач:

- Рассчитайте в считанном текстовом файле количество строк;
- Определите, сколько слов имеется в считанном текстовом файле;
- Подсчитайте в считанном текстовом файле количество букв русского алфавита.

15. Какую задачу решает приведенная ниже программа:

```

str = input("Write down or insert some text:\n")
punctuation = [',', ';', ':', '!', '?', '(', ')']
wordList = str.split()
i = 0
for word in wordList:
 if word[-1] in punctuation:
 wordList[i] = word[:-1]
 word = wordList[i]
 if word[0] in punctuation:
 wordList[i] = word[1:]
 i += 1
i = 0
while i < len(wordList):
 print(wordList[i], end=' ')
 i += 1
 if i%5 == 0:
 print()

```

16. Напишите программу, которая переводит введенное пользователем десятичное число в двоичную, восьмеричную, шестнадцатеричную системы счисления, и выведите полученные результаты.

17. Решите следующую задачу с приведением подробных комментариев по используемому алгоритму:

Задан случайный массив чисел из 77 элементов. Составьте отдельные три функции, которые вычисляют сумму этих чисел: с for - циклом, с while - циклом, с рекурсией.

18. Решите следующую задачу с приведением подробных комментариев по используемому алгоритму:

Приведите функцию, которая интегрирует исходные два списка следующим образом:

[1, 3, 5, 7, 9] (\*) [11, 33, 55, 77, 99] -> [11, 99, 555, 777, 999]

19. Решите следующую задачу с приведением подробных комментариев по используемому алгоритму:

Имеется введенный массив чисел из текстового файла, составьте из них максимальное число. Например,

[61, 228, 9] -> 961228

20. Решите следующую задачу с приведением подробных комментариев по используемому алгоритму:

Дана последовательность девяти цифр: 1, 2, ..., 9 именно в таком порядке. В составленной Вами программе можно вставлять между ними знаки «+», «-» или ничего. При этом могут генерироваться выражения вида 123+45-6+7+89. Найдите все возможные комбинации из них, которые в процессе вычислений дадут на выходе число 111.

## Блок С. Задания практико-ориентированного уровня для диагностирования сформированности компетенции («владеть»)

### С1. Практико-ориентированные задачи на программирование

1. Вклад в банке составляет  $X$  рублей. Ежегодно он увеличивается на  $P$  процентов, после чего дробная часть копеек отбрасывается. Каждый год сумма вклада становится больше. Определите, через сколько лет вклад составит не менее  $Y$  рублей. Результат оформите в виде функции. Функция получает на вход три натуральных числа:  $X$ ,  $P$ ,  $Y$  и должна вывести одно целое число.

2. У исполнителя «Посейдон» есть два сосуда, первый объемом  $A$  литров, второй объемом  $B$  литров, а также источник с водой. Водолей может выполнять следующие операции:

- Наполнить сосуд  $A$  (обозначается  $> A$ ).
- Наполнить сосуд  $B$  (обозначается  $> B$ ).
- Вылить воду из сосуда  $A$  (обозначается  $A <$ ).
- Вылить воду из сосуда  $B$  (обозначается  $B <$ ).
- Перелить воду из сосуда  $A$  в сосуд  $B$  (обозначается как  $A > B$ ).
- Перелить воду из сосуда  $B$  в сосуд  $A$  (обозначается как  $B > A$ ).

Команда переливания из одного сосуда в другой приводят к тому что, либо первый сосуд полностью опустошается, либо второй сосуд полностью наполняется.

Программа получает на вход три натуральных числа  $A$ ,  $B$ ,  $N$ , не превосходящих  $10^4$ . Необходимо вывести алгоритм действий Посейдона, который позволяет получить в точности  $N$  литров в одном из сосудов, если же такого алгоритма не существует, то программа должна вывести текст «Невозможно». Количество операций в алгоритме не должно превышать  $10^5$ . Гарантируется что, если задача имеет решение, то есть решение, которое содержит не более, чем  $10^5$  операций. Составить программу для решения данной задачи.

3. В операционной системе Windows полное имя файла состоит из буквы диска, после которого ставится двоеточие и символ «\», затем через такой же символ перечисляются подкаталоги (папки), в которых находится файл, в конце пишется имя файла. Пример: `C:\Windows\System32\calc.exe`

Дано некоторое полное имя файла. «Разобрать» его на части, разделенные символом «\». Каждую часть вывести в отдельной строке выходного файла `output.txt`.

4. Дана фраза, в которой, кроме букв, имеются пробелы. Проверить, является ли она палиндромом без учета пробелов (например, фраза «АРГЕНТИНА МАНИТ НЕГРА» палиндромом является).

5. Дано положительное вещественное число. Определить количество цифр:

- а) в его целой части;
- б) в его дробной части.

Функции для работы с вещественными числами не использовать.

6. Дана строка, в которой без пробелов записано арифметическое выражение в виде суммы  $N$  натуральных чисел, например - «11+27+ ... + 33». Вычислить эту сумму.

7. Имеется файл, в каждой строке которого записано целое число. Определить:

- а) сумму всех чисел;
- б) среднее арифметическое всех чисел;
- в) сумму чисел, записанных на 2-й, 4-й, 6-й, ... строках;
- г) количество четных чисел;
- д) среднее арифметическое отрицательных чисел;
- е) максимальное число в файле;
- ж) номер строки, в которой записано первое минимальное число файла;
- з) имеется ли в файле заданное число.

Списки не использовать.

8. Определить количество шестизначных «счастливых» чисел. «Счастливым» будем называть такое шестизначное число, в котором сумма его первых трех цифр равна сумме его последних трех цифр. Разработайте два варианта программы:

а) в котором определена и используется функция для расчета суммы цифр трехзначного числа;

б) в котором определены и используются функция для расчета суммы цифр трехзначного числа и функция, проверяющая, является ли «счастливым» некоторое шестизначное число.

## Блок Д. Задания для использования в рамках промежуточной аттестации

### Д1. Перечень примерных задач на программирование в ходе промежуточной аттестации

1. Неизвестный водитель совершил ДТП и скрылся с места происшествия. Полиция опрашивает свидетелей. Каждый из них говорит, что запомнил какие-то буквы и цифры номера. Но при этом свидетели не помнят порядок этих цифр и букв. Полиция хочет проверить несколько подозреваемых автомобилей. Будем говорить, что номер согласуется с показанием свидетеля, если все символы, которые назвал свидетель, присутствуют в этом номере (не важно, сколько раз).

Формат ввода

Сначала задано число – количество свидетелей. Далее идет  $M$  строк, каждая из которых описывает показания очередного свидетеля. Эти строки непустые и состоят из не более чем 20 символов. Каждый символ в строке – либо цифра, либо заглавная латинская буква, причем символы могут повторяться.

Формат вывода

Выпишите номера автомобилей, согласующиеся с максимальным количеством свидетелей. Если таких номеров несколько, то выведите их в том же порядке, в котором они были заданы на входе.

2. Необходимо реализовать упрощенный аналог утилиты `cut`. Эта утилита должна принимать на вход таблицу, в которой строки разделены переносом строки, а столбцы разделены символом табуляции. Утилита удаляет из таблицы все столбцы кроме одного.

Формат ввода

В первой строке файла идет описание опций программы. Она может быть пустой или содержать опцию `-c` и далее целое число. В последующих строках идут строки таблицы (столбцы разделены символом табуляции `\t`). Гарантируется, что в таблице не более 1000 строк и не более 1000 столбцов.

Формат вывода

Если столбец не задан (первая строка пустая), то выведите таблицу без изменений. Если же столбец задан, то выведите только этот столбец (строки опять же отделяются друг от друга переносом строки).

Количество столбцов может быть разным в разных строках. Если в какой – то строке нет требуемого столбца (их слишком мало), то выведите вместо него пустую строку.

3. Известный алгоритм Soundex (<https://ru.wikipedia.org/wiki/Soundex>) определяет, похожи ли два английских слова по звучанию. На вход он принимает слово и заменяет его на некоторый четырехсимвольный код. Если коды двух слов совпадают, то слова, как правило, звучат похоже.

Требуется реализовать этот алгоритм. Он работает так:

Первая буква слова сохраняется.

В остальной части слова:

буквы, обозначающие, как правило, гласные звуки: a, e, h, i, o, u, w и y — отбрасываются;

оставшиеся буквы (согласные) заменяются на цифры от 1 до 6, причём похожим по звучанию буквам соответствуют одинаковые цифры:

1: b, f, p, v

2: c, g, j, k, q, s, x, z

3: d, t

4: l

5: m, n

6: r

Любая последовательность одинаковых цифр сокращается до одной такой цифры.

Итоговая строка обрезается до первых четырёх символов. Если длина строки меньше требуемой, недостающие символы заменяются знаком 0.

Формат ввода

На вход подаётся одно непустое слово, записанное строчными латинскими буквами. Длина слова не превосходит 20 символов.

Формат вывода

Напечатайте четырёхбуквенный код, соответствующий слову.

4. Пешеход подошел к перекрестку и увидел, что на светофоре в этот момент загорелся красный свет. Вася залюбовался тем, как четко переключаются сигналы светофора:

красный – желтый – зеленый – желтый – красный – желтый – зеленый - ...

Когда в очередной раз загорелся зеленый свет, Вася решил-таки перейти дорогу. К этому моменту зеленый свет зажегся в  $i$ -ый раз. Напишите программу, которая определит, сколько раз за это время загорался красный свет (считая и тот момент, когда Вася только подошел к перекрестку) и сколько раз — желтый.

Входные данные

Во входном файле задано одно число  $i$ , задающее, в какой раз загорелся зеленый свет ( $1 \leq i \leq 100$ ).

Выходные данные

В выходной файл выведите два числа. Первое — сколько раз загорался красный свет, второе — сколько раз загорался желтый.

5. Даны два четырёхзначных числа  $A$  и  $B$ . Выведите в порядке возрастания все четырёхзначные числа в интервале от  $A$  до  $B$ , запись которых содержит ровно три одинаковые цифры.

Входные данные: вводятся два целых числа  $A$  и  $B$ .

Выходные данные: Найденные числа в порядке возрастания.

### РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Балльно-рейтинговая система является базовой системой оценивания сформированности компетенций обучающихся очной формы обучения.

Итоговая оценка сформированности компетенции обучающихся в рамках балльно-рейтинговой системы осуществляется в ходе текущего контроля успеваемости, промежуточной аттестации и определяется как сумма баллов, полученных обучающимися в результате прохождения всех форм контроля.

Оценка сформированности компетенции по дисциплине складывается из двух составляющих:

✓ первая составляющая – оценка преподавателем сформированности компетенции в течение семестра в ходе текущего контроля успеваемости (максимум 100 баллов). Структура первой составляющей определяется технологической картой дисциплины, которая в начале семестра доводится до сведения обучающихся;

✓ вторая составляющая – оценка сформированности компетенции(й) обучающихся на экзамене (максимум – 30 баллов).

Для студентов очно-заочной и заочной форм обучения применяются 4-балльная и бинарная шкалы оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся.

Шкала оценки преподавателем сформированности компетенции в течение семестра

| Уровни освоения компетенций | продвинутый уровень | базовый уровень | пороговый уровень | допороговый уровень |
|-----------------------------|---------------------|-----------------|-------------------|---------------------|
| <b>100 – балльная шкала</b> | 85 и $\geq$         | 70 – 84         | 51 – 69           | 0 – 50              |
| <b>4 – балльная шкала</b>   | отлично             | хорошо          | удовлетворительно | неудовлетворительно |

#### Шкала оценок при текущем контроле успеваемости по различным показателям

| <i>Показатели оценивания сформированности компетенций</i> | <i>Баллы</i> | <i>Оценка</i>                                                         |
|-----------------------------------------------------------|--------------|-----------------------------------------------------------------------|
| Тестирование                                              | 0-15         | «неудовлетворительно»<br>«удовлетворительно»<br>«хорошо»<br>«отлично» |
| Задачи на программирование                                | 0-30         | «неудовлетворительно»<br>«удовлетворительно»<br>«хорошо»<br>«отлично» |
| Практико-ориентированные задачи                           | 0-40         | «неудовлетворительно»                                                 |

|                                                             |        |                                                                       |
|-------------------------------------------------------------|--------|-----------------------------------------------------------------------|
| на программирование                                         |        | «удовлетворительно»<br>«хорошо»<br>«отлично»                          |
| Разбор и обсуждение вопросов, связанных с программированием | 0 - 15 | «неудовлетворительно»<br>«удовлетворительно»<br>«хорошо»<br>«отлично» |

**Соответствие критериев оценивания уровню освоения компетенций по текущему контролю успеваемости**

| <i><b>Баллы</b></i> | <i><b>Оценка</b></i>  | <i><b>Уровень освоения компетенций</b></i> | <i><b>Критерии оценивания</b></i>                                                                                                                                                                                                                                                                     |
|---------------------|-----------------------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-50                | «неудовлетворительно» | Допороговый уровень                        | Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины                                                                                                                                                                                |
| 51-69               | «удовлетворительно»   | Пороговый уровень                          | Не менее 50% заданий, подлежащих текущему контролю успеваемости, выполнены без существенных ошибок                                                                                                                                                                                                    |
| 70-84               | «хорошо»              | Базовый уровень                            | Обучающимся выполнено не менее 75% заданий, подлежащих текущему контролю успеваемости, или при выполнении всех заданий допущены незначительные ошибки; обучающийся показал владение навыками систематизации материала и применения его при решении практических заданий; задания выполнены без ошибок |
| 85-100              | «отлично»             | Продвинутый уровень                        | 100% заданий, подлежащих текущему контролю успеваемости, выполнены самостоятельно и в требуемом объеме; обучающийся проявляет умение обобщать, систематизировать материал и применять его при решении практических заданий; задания выполнены с подробными пояснениями и аргументированными выводами  |

### Шкала оценок по промежуточной аттестации

| <i>Наименование формы промежуточной аттестации</i> | <i>Баллы</i> | <i>Оценка</i>         |
|----------------------------------------------------|--------------|-----------------------|
| Экзамен                                            | 0-50         | «неудовлетворительно» |
|                                                    | 51-69        | «удовлетворительно»   |
|                                                    | 70-84        | «хорошо»              |
|                                                    | 85-100       | «отлично»             |

### Соответствие критериев оценивания уровню освоения компетенций по промежуточной аттестации обучающихся

| <i>Баллы</i> | <i>Оценка</i>         | <i>Уровень освоения компетенций</i> | <i>Критерии оценивания</i>                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|-----------------------|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-50         | «неудовлетворительно» | Допороговый уровень                 | Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины; обучающийся не смог ответить на вопросы                                                                                                                                                                                                                                           |
| 51-69        | «удовлетворительно»   | Пороговый уровень                   | Обучающийся дал неполные ответы на вопросы, с недостаточной аргументацией, практические задания выполнены не полностью, компетенции, осваиваемые в процессе изучения дисциплины сформированы не в полном объеме.                                                                                                                                                                                          |
| 70-84        | «хорошо»              | Базовый уровень                     | Обучающийся в целом приобрел знания и умения в рамках осваиваемых в процессе обучения компетенций; обучающийся ответил на все вопросы, точно дал определения и понятия, но затрудняется подтвердить теоретические положения практическими примерами; обучающийся показал хорошие знания по предмету, владение навыками систематизации материала и полностью выполнил практические задания                 |
| 85-100       | «отлично»             | Продвинутый уровень                 | Обучающийся приобрел знания, умения и навыки в полном объеме, закрепленном рабочей программой дисциплины; терминологический аппарат использован правильно; ответы полные, обстоятельные, аргументированные, подтверждены конкретными примерами; обучающийся проявляет умение обобщать, систематизировать материал и выполняет практические задания с подробными пояснениями и аргументированными выводами |

#### **РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций**

Процедура оценивания – порядок действий при подготовке и проведении аттестационных испытаний и формировании оценки, характеризующей степень освоения запланированных компетенций в рамках данной дисциплины.

Процедура промежуточной аттестации проходит в соответствии с Положением о промежуточной аттестации знаний студентов и учащихся ДГУНХ.

- Аттестационные испытания проводятся преподавателем, ведущим лекционные занятия по данной дисциплине, или преподавателями, ведущими практические и лабораторные занятия. Присутствие посторонних лиц в ходе проведения аттестационных испытаний без разрешения ректора или проректора по учебной работе не допускается (за исключением работников университета, выполняющих контролирующие функции в соответствии со своими должностными обязанностями). В случае отсутствия ведущего преподавателя аттестационные испытания проводятся преподавателем, назначенным письменным распоряжением по кафедре (структурному подразделению).
- Инвалиды и лица с ограниченными возможностями здоровья, имеющие нарушения опорно-двигательного аппарата, допускаются на аттестационные испытания в сопровождении ассистентов-сопровождающих.
- Во время аттестационных испытаний обучающиеся могут пользоваться программой дисциплины, а также с разрешения преподавателя справочной и нормативной литературой, персональным компьютером с установленной средой программирования языка Python.
- Время написания предложенных во время экзамена программ при его сдаче в компьютерном классе составляет не менее 90 минут.
- Оценка результатов устного аттестационного испытания объявляется обучающимся в день его проведения. При проведении письменных аттестационных испытаний или компьютерного тестирования – в день их проведения или не позднее следующего рабочего дня после их проведения.
- Результаты выполнения аттестационных испытаний должны быть объявлены обучающимся и выставлены в зачётные книжки не позднее следующего рабочего дня после их проведения.

Итоговой формой контроля по дисциплине является экзамен.

## Оценивание выполнения практико-ориентированных задач на программирование

| Баллы | Оценка       | Показатели                                                       | Критерии                                                                                                                                                 |
|-------|--------------|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34-40 | «зачтено»    | Полнота выполнения практико-ориентированных заданий;             | Выполнено 85 и более % заданий предложенного спектра практико-ориентированных заданий, в которых запрограммирован необходимый спектр предложенных задач. |
| 28-33 | «зачтено»    | Своевременность выполнения;                                      | Выполнено от 70 до 84% заданий предложенного спектра практико-ориентированных заданий.                                                                   |
| 21-27 | «зачтено»    | Правильность написания программ;                                 | Выполнено от 51 до 69% заданий предложенного спектра практико-ориентированных заданий.                                                                   |
| 0-20  | «не зачтено» | Самостоятельность в ходе программирования;                       | Выполнено менее 51% заданий предложенного спектра практико-ориентированных заданий.                                                                      |
|       |              | Уверенный, обоснованный ответ на вопросы по предложенной задаче; |                                                                                                                                                          |
|       |              | и т.д.                                                           |                                                                                                                                                          |

## Методика оценивания вопросов, вынесенных на обсуждение

| Баллы | Оценка    | Показатели                                                          | Критерии                                                                                                                                                                                                                                                                                        |
|-------|-----------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13-15 | «зачтено» | Полнота данных ответов на знание синтаксиса языка программирования; | Полно и аргументировано даны ответы по содержанию вопроса. Обнаружено понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные. Изложение материала последовательно и правильно. |
| 10-12 | «зачтено» | Аргументированность данных ответов по языку Python;                 | Студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1-2 ошибки, которые сам же исправляет.                                                                                                                                                         |
| 5-9   | «зачтено» | Правильность ответов на вопросы при решении устных задач            | Студент обнаруживает знание и понимание основных положений данного задания, но:<br>1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил;                                                                                                           |

|         |              |                                        |                                                                                                                                                                                                                                                                                                                                   |
|---------|--------------|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         |              | на программирование у доски;<br>и т.д. | 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры;<br>3) излагает материал непоследовательно и допускает ошибки.                                                                                                                                                                     |
| Менее 5 | «не зачтено» |                                        | Студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом. |

### Оценивание выполненных задач на программирование

| Баллы | Оценка    | Показатели                                                                                                                                                                                                                                          | Критерии                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26-30 | «зачтено» | 1. Полнота решения задач, связанных с программированием;<br>2. Своевременность выполнения;<br>3. Правильность использованного алгоритма;<br>4. Эффективность использованного алгоритма с точки зрения использования памяти ПК и времени выполнения. | Основные требования к программированию поставленных задач выполнены. Продемонстрированы умение четко сформулировать постановку предложенной задачи, определиться с алгоритмом решения и находить оптимальное решение, умение работать с исходными данными различного формата, навыки четкого и точного изложения собственной точки зрения в устной и письменной форме, убедительного отстаивания своей точки зрения, применение эффективных алгоритмов, четкое владение функционалом языка программирования. |
| 21-25 | «зачтено» | 5. Самостоятельность в выполнении;<br>6. Умение четко и лаконично форм-                                                                                                                                                                             | Основные требования к решению задач выполнены, но при этом допущены недочеты. В частности, использован не самый эффективный алгоритм, структура входных данных не универсальна, недостаточно учтены возможные ошибки при вводе неструктурированных данных, проведен не полный анализ задачи,                                                                                                                                                                                                                 |

|       |              |                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                        |
|-------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       |              | мулировать постановку задачи, алгоритм ее решения                                                                                                                                                                                                                                                                    | нет самоконтроля и самооценки, нестандартности предлагаемых решений                                                                                                                                                                                                                                                                                                    |
| 16-20 | «зачтено»    | и дать описание выполненной программы на языке программирования;<br>7. Владение функциональными особенностями изученного языка программирования (использование имеющихся библиотек, дополнительных программных модулей, собственных функций, наличие компетенций в сфере объектно-ориентированного программирования. | Имеются существенные отступления от постановки предлагаемой задачи, задача решена частично, все условия задачи не выполнены. В частности, отсутствуют навыки умения моделировать алгоритм решения в соответствии с заданием, представлять различные подходы к решению задачи, использование упрощенных алгоритмов решения задач, ориентированных на конечный результат |
| 0-15  | «не зачтено» | пользование имеющихся библиотек, дополнительных программных модулей, собственных функций, наличие компетенций в сфере объектно-ориентированного программирования.                                                                                                                                                    | Задача не решена, обнаруживается существенное непонимание постановки задачи и алгоритма ее решение, отсутствие умений и навыков составления программ на основе типовых алгоритмов, незнание основных операторов языка программирования и практики их использования в процессе решения прикладных учебных задач.                                                        |

### Методика оценивания выполнения тестов

| Баллы | Оценка    | Показатели                                                            | Критерии                                                                                                                                                                                            |
|-------|-----------|-----------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13-15 | «зачтено» | 1. Полнота выполнения тестовых заданий;                               | Выполнено более 85 % заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос.                                                                   |
| 10-12 | «зачтено» | 2. Своевременность выполнения;<br>3. Правильность ответов на вопросы; | Выполнено 70-84% заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос; однако были допущены неточности в определении понятий, терминов и др. |
| 6-9   | «зачтено» | 4. Самостоятельность тестирования;                                    | Выполнено 51-69% заданий предложенного теста, в заданиях открытого типа дан неполный ответ на поставленный во-                                                                                      |

|     |              |           |                                                                                                                                                                                       |
|-----|--------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     |              | 5. и т.д. | прос, в ответе не присутствуют доказательные примеры, текст со стилистическими и орфографическими ошибками.                                                                           |
| 0-5 | «не зачтено» |           | Выполнено менее 51% заданий предложенного теста, на поставленные вопросы ответ отсутствует или неполный, допущены существенные ошибки в теоретическом материале (терминах, понятиях). |