

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)**

ОТДЕЛЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В ГУМАНИТАРНОЙ СФЕРЕ
*Учебно-научный центр программного и лингвистического обеспечения
интеллектуальных систем*

ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

45.03.04 Интеллектуальные системы в гуманитарной сфере

Разработка и программирование интеллектуальных систем в гуманитарной сфере

Уровень высшего образования: бакалавриат

Форма обучения очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2023

Язык программирования Python
Рабочая программа дисциплины
Составитель:
доцент
М.Е. Епифанов

.....

УТВЕРЖДЕНО
Протокол совместного заседания ОИС в ГС и кафедры МЛиИС
№ 2 от 16.03.2023

ОГЛАВЛЕНИЕ

1. Пояснительная записка

1.1 Цели и задачи дисциплины

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине

1.3. Место дисциплины в структуре образовательной программы

2. Структура дисциплины

3. Содержание дисциплины

4. Образовательные технологии

5. Оценка планируемых результатов обучения

5.1. Система оценивания

5.2. Критерии выставления оценок

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

7. Материально-техническое обеспечение дисциплины

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

9. Методические материалы

9.1. Планы практических (семинарских, лабораторных) занятий

9.2. Методические указания для обучающихся по освоению дисциплины

Приложения

Приложение 1. Аннотация дисциплины

1. Пояснительная записка

1.1. Цели и задачи дисциплины

Цели дисциплины:

- воспитать студента как квалифицированного специалиста в области разработки кроссплатформенных приложений на языке программирования Python;
- способствовать формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины:

- приобретение студентами знаний и первичного опыта, связанных с разработкой приложений на языке Python;
- дальнейшее формирование у студентов логического и алгоритмического мышления с помощью специально подобранных упражнений;
- выработка у студентов способности самостоятельно овладевать современными инструментальными средами и технологиями программирования.

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения дисциплине:

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ПК-1 Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов	<p>ПК-1.1 Знает теоретические основы построения алгоритмов обработки информации</p> <p>ПК-1.3 Имеет практический опыт разработки алгоритмов обработки информации с использованием современных математических методов</p>	<p><i>Знать:</i></p> <ul style="list-style-type: none"> - основные свойства алгоритмов, формы записи алгоритмов, базовые алгоритмические структуры; синтаксис языка программирования Python, его основные конструкции их функциональность, терминологию и основные понятия, связанные с разработкой приложений на этом языке; - встроенные типы данных и возможности для определения новых типов данных в Python; - объектную структуру пакетов стандартной библиотеки Python, поддерживаемую ими функциональность; - стандартные структуры данных (включая списки, стеки, очереди, деки, деревья, графы) и варианты их реализации в программах на языке Python. <p><i>Уметь:</i></p> <p>грамотно разрабатывать алгоритмы и их программные решения, комплексно используя применяемые для разработки клиентских Webприложений средства.</p> <p><i>Владеть</i> навыками самостоятельного специалиста в области программной реализации прикладных приложений.</p>
ПК-1 Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов	ПК-1.2 Умеет описывать алгоритмы обработки информации с использованием современных математических методов	<p><i>Знать:</i></p> <p>основные принципы использования интегрированной среды разработки (IDE) NetBeans программных приложений на языке Python.</p> <p><i>Уметь:</i></p> <p>эффективно использовать руководства пользователя (users guides/manuals), справочники по языкам (language references), стандарты и др. источники для освоения и применения средств программирования на Python.</p> <p><i>Владеть</i> навыками:</p> <ul style="list-style-type: none"> - анализа документации к программным

5	Библиотека Python (продолжение).	6	4		8			18	Оценка выполнения практических заданий
6	Практикум по программированию на языке Python.	6	6		24			48	Оценка выполнения практических заданий
	Итоговая аттестация	6							зачет с оценкой
	Итого		20		22			136	

3. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела
1	Введение в «мир» Python	Сравнение Python и C++. Python как полностью объектноориентированный язык. Структура приложения на языке Python. Кроссплатформенные приложения. Виртуальная Pythonмашина (JVM). Исходный код и byte-код. Компилятор.
2	Интегрированная среда разработки NetBeans.	Сравнение с Visual Studio. Основные инструменты. Отладка программ.
3	Язык Python.	Обзор языка Python. «Процедурные» средства (переменные и типы данных, операции, управляющие конструкции и т.п.); Объектноориентированное программирование в Python: - парадигма ООП в Python, - классы, структура и методы, инкапсуляция, полиморфизм, наследование, - пакеты и интерфейсы, - агрегация vs. наследование. Обработка исключений Ввод-вывод.
4	Библиотека Python.	Работа со строками. Пакет Python.lang. Ввод-вывод – пакет Python.io. Пакет Python.util – коллекции и сервисные классы.
5	Библиотека Python (продолжение).	Обработка событий. Системы AWT и Swing. Компоненты Python Beans. Средства для реализации распределенных приложений (включая класс Applet, сервлеты,...) Представление о многопоточном программировании в Python.

6	Практикум по программированию на языке Python.	Предлагаемый(-ые) учебный(-ые) проект(-ы) может/могут меняться год от года.
---	--	---

4. Образовательные технологии

№ п/п	Наименование раздела	Виды учебных занятий	Образовательные технологии
1	2	3	4
1	Введение в «мир» Python	Лекции 1, 2 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Консультирование посредством электронной почты
2	Интегрированная среда разработки NetBeans.	Практическое занятие 1 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Практикум по решению задач. Консультирование посредством электронной почты
3	Язык Python.	Лекции 3-5 Практические занятия 2-13 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
4	Библиотека Python.	Практические занятия 14-16 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
5	Библиотека Python (продолжение).	Лекции 6, 7 Лабораторные занятия 17-20 Самостоятельная работа	Теоретическая лекция. Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты
6	Практикум по программированию на языке Python.	Лекции 8-10 Практические занятия 21-32 Самостоятельная работа	Демонстрация (runtime) выполнения кодов. Семинар-обсуждение. Практикум по решению задач. Консультирование посредством электронной почты

5. Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Срок отчетности	Макс. количество баллов	
		За одну работу	Всего
Текущий контроль: • Опрос (1—2) • контроль за программным выполнением практических заданий (темы 3—5) • контр. работа (тема 3)	3—16 недели 2—16 недели 12 неделя	5 баллов 20 баллов	5 баллов 45 баллов 10 баллов
Промежуточная аттестация (зачет)	17 неделя		40 баллов
Итого за семестр (дисциплину)			100 баллов
Текущий контроль: • контроль за программным выполнением практических заданий (темы 6,7) • учебный проект (тема 8)	7—16 недели	40 баллов	40 баллов
Промежуточная аттестация (экзамен)	17 неделя		40 баллов
Итого за семестр (дисциплину)			100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено»	Выставляется обучающемуся, если он: - грамотно выполнил большинство (в том числе и все обязательные) практические задания (программные образцы – samples); - глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации;

		<ul style="list-style-type: none"> - исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения; - свободно ориентируется в учебной и профессиональной литературе. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	«хорошо»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - грамотно, возможно с помощью преподавателя, выполнил достаточное количество практических заданий (программные образцы – samples), в том числе и все обязательные; - знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей; - правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами; - достаточно хорошо ориентируется в учебной и профессиональной литературе. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»/ «зачтено»	<p>Выставляется обучающемуся, если он:</p> <ul style="list-style-type: none"> - выполнил (возможно с помощью преподавателя) достаточное количество практических заданий (программные образцы – samples); - знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации; - испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами; - демонстрирует достаточный уровень знания учебной литературы по дисциплине.

		<p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не явился по неуважительной/неизвестной причине на аттестацию или:</p> <ul style="list-style-type: none"> - не выполнил (несмотря на возможную помощь преподавателя) достаточное количество практических заданий (программные образцы – samples); - не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации. - испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами; - демонстрирует фрагментарные знания учебной литературы по дисциплине. <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

5.3.1. Образцы заданий для самостоятельного выполнения

1) Обзорный программный образец (sample) “Базовое «процедурное» подмножество языка Python” (простые типы (без объектных оболочек), арифметики этих типов, автоматические преобразования (коэрсии) для этих типов и т.п.)

2) Простые задачи из «начал процедурного программирования» (по книге Александра Шеня, см. список литературы в п. 6). Например.

- Задачи «без массивов».
- Разными способами программируя (цикл for, либо цикл while, либо рекурсивно) вычислить для заданного натурального числа n
 - $n!$;
 - n -й член последовательности Фибоначчи;
 - сумму первых n членов ряда разложения числа e
 - и т.п.
- Задачи «с массивами».

Здесь используются некоторые особенности массивов в Python, в частности, свойство (переменная член экземпляра класса) `length` – длина массива. Пусть A – массив длины n (`A = new int[n]`).

- Примеры простых задач.

- Заполнить массив нулями. Чуть более сложный вариант. Число нулей задается, массив заполняется, если заданное число нулей не превосходит n , иначе – сообщение об ошибке в консоль.
- Подсчитать количество вхождений заданного целого в массив.
- Копировать исходный массив в другой.
- Найти максимальный/минимальный члены массива с указанием номеров их позиций.
- Переписать элементы исходного массива в другой в обратном порядке (reverse). Более сложный вариант: проделать то же самое в исходном массиве (не используя другого массива).
- Исходный массив упорядочен ($A[0] \leq \dots \leq A[i] \leq A[i+1] \leq \dots \leq A[n]$). Найти количество различных чисел среди элементов A .
- и т.п.
- Более сложный пример. Реализовать разными способами (в цикле и рекурсивно) сортировку слиянием массива.

3) Объектноориентированное программирование в Python.

- Простые задачи на определения классов без наследования. В том числе – реализация стека и очереди через массив.
- Реализация изолированного класса объекты которого поддерживают «внутреннее состояние» через взаимосвязанные данные (см. ниже в п. 9).
- Программные образцы «иерархий по наследованию» классов.
- Простые геометрические именованные объекты (точка, окружность, отрезок по двум точкам – концам, ...).
- Простой стек и стек, удваивающий свою длину при переполнении. В одном варианте первый – суперкласс для второго, в другом – оба подклассы абстрактного класса.
- Стек и очередь (с использованием коллекции ArrayList из пакета Python.util).
- и т.п.
- Использование интерфейсов.
- Применение агрегации взамен наследования в некоторых случаях «комбинаторного» роста классов из-за «независимой функциональности».

4) Обзорные программные образцы по функциональности коллекций из пакета Python.util.

5) Обзорные программные образцы по функциональности, поддерживаемой некоторыми другими пакетами библиотеки Python.

6) В 6-м семестре в рамках «практикума по программированию на языке Python» студенты разбиваются не «рабочие» группы и совместно выполняют отдельные части некоторого «большого» проекта под руководством преподавателя, как заказчика, так и руководителя работ. Т.о. образом отчасти имитируются производственные отношения в реальных коллективах, разрабатывающих реальных программные приложения. Такой проект может длиться несколько лет и передаваться от «поколения к поколению» т.е. потока предыдущего уч. года к потоку следующего уч. года. (Так было с проектом «Ходик» вплоть до 17/18 уч. года.)

Зачеты (с оценкой) в 5-м и 6-м семестре выставляются по результатам выполнения задач в семестре.

6. Учебно-методическое и информационное обеспечение дисциплины

Замечание: многие источники полностью или в виде актуальных фрагментов имеются в электронной библиотеке (ЭБ) Учебно-методического кабинета (УМК) Отделения интеллектуальных систем (ОИС), папка [d:_ois_lib] (либо [c:_ois_lib] в случае единственного логического диска) на компьютерах деканата.

6.1. Список источников и литературы

а) Основная литература

1. *Монахов В.В.* Язык программирования Python и среда NetBeans : учебное пособие / В. В. Монахов. — 2-е изд. — Москва : ИНТУИТ, 2016. — 450 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100544>
2. *Ноутон П., Шилдт Г.* Python2. Полное руководство. – СПб.: БХВ-Петербург, 2008. – 1055 с. (ЭБ УМК ОИС: [d:_ois_lib\3 Python\Python_2_Naughton-Schildt.pdf])
3. *Шилдт Г.* Python 8. Полное руководство. – М.: Издательский дом «Вильямс», 2016 (С примерами - ЭБ УМК ОИС: [d:_ois_lib\3 Python\Шилдт-Герберт- Python-8-руководство-для- начинающих-примеры.zip])
4. Раздел портала Oracle, содержащий техническую документацию (онлайн и в виде свободно распространяемых электронных книг) о платформе Python Standard Edition (Python SE): <http://docs.oracle.com/Pythonse/>

б) Дополнительная литература

1. *Даконта М., Саганич А.* XML и PYTHON 2 (серия "Библиотека программиста"). – СПб.: Питер, 2001.
2. *Флэнаган Д.* Python в примерах. – СПб.: Символ-Плюс, 2003.
3. *Хорстманн К.С., Корнелл Г.* Python 2. Библиотека профессионала. Т. 1,2. Тонкости программирования. 8-е издание. – М.: ООО “И. Д. Вильямс”, 2009.
4. *Шилдт Г.* Полный справочник по Python. – М.: Издательский дом «Вильямс», 2007

6.2 Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины

<https://github.com/> – GitHub – веб-сервис для хостинга IT-проектов и их совместной разработки

<http://msdn.microsoft.com> – MSDN Microsoft Developer Network – техническая документация фирмы Microsoft для разработчиков ПО:

<http://severe-programmer.com/manual/ustanovka-i-nastrojka-apache-mysql-na-os-x-10-9-mavericks/> – Рекомендации по установке и настройке Apache и MySQL

[https://tproger.ru/translations/ Python-json-library-comparison/](https://tproger.ru/translations/Python-json-library-comparison/) – Сравнение четырёх популярных библиотек Python для работы с JSON

http://vana.nvtc.ee:89/evara/Программа_MySQL_Workbench_ru.pdf – Руководство по разработке БД при помощи MySQL Workbench

[http://www.oracle.com/technetwork/ Python/index-jsp-142942.html](http://www.oracle.com/technetwork/Python/index-jsp-142942.html) – Справочная документация по Oracle Python Enterprise Edition

[http://www.oracle.com/technetwork/ Python/ Pythonee/downloads/index.html](http://www.oracle.com/technetwork/Python/Pythonee/downloads/index.html) – Руководство по установке и настройке Apache Tomcat, регистрации в NetBeans

Перечень БД и ИСС

№п /п	Наименование
-------	--------------

1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2023 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2023 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis
3	Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru
4	Компьютерные справочные правовые системы Консультант Плюс, Гарант

7. Материально-техническое обеспечение дисциплины

Занятия проводятся в компьютерных классах ауд. 307 и 706, расположенных по адресу 125993, Москва, Миусская пл., д. 6, стр.2,.

Этот компьютерный класс оснащен

- достаточным количеством объединенных в локальную сеть рабочих станций,
- медиапроектором и экраном,
- меловой доской.

В классе имеются возможности

- подключения ноутбука к медиапроектору,
- одновременного доступа в Интернет для преподавателя и студентов.

1. Перечень ПО

№п /п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Adobe Master Collection CS4	Adobe	лицензионное
2	ОС «Альт Образование» 8	ООО «Базальт СПО	лицензионное
3	Windows 10 Pro	Microsoft	лицензионное
4	Kaspersky Endpoint Security	Kaspersky	лицензионное
5	Microsoft Office 2016	Microsoft	лицензионное

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.
- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа;
 - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:
 - передвижными, регулируемые эргономическими партами СИ-1;
 - компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1. Планы семинарских занятий

Лабораторные занятия по «блокам» схожих тем однотипны, они проводятся по единой схеме. На занятиях объясняются основные теоретические положения изучаемых тем, формулируются постановки для части задач, предлагаются методы, возможные сценарии

и технологические элементы для их решения. Часть задач объясняется непосредственно на практических занятиях.

Углубленно текущий материал изучается студентами самостоятельно.

Почти все практические задания также выполняются студентами в процессе самостоятельной работы.

На практических занятиях при необходимости проводится разбор текущего материала и контроль в форме опросов и дискуссий его усвоения.

Наконец, на практических занятиях осуществляется контроль за выполнением практических заданий, при необходимости проводятся индивидуальные или групповые консультации.

9.1. Пример планов лабораторных занятий

1. Реализация изолированного класса объекты которого поддерживают «внутреннее состояние» через взаимосвязанные данные (практическое задание к части «Объектноориентированное программирование в Python» из темы 3).

Цель занятий: изучение и приобретение некоторого опыта, основных навыков разработки обозначенных в заголовке плана классов.

Форма проведения – демонстрация (runtime) выполнения кодов, спецификация задачи преподавателем, семинар-обсуждение, практикум по решению задачи.

Пример учебного проекта.

Экземпляр (объект) класса `qEquation` содержит, вообще говоря, неполные данные о возможном решении квадратного уравнения. В течение жизни объекта каждый из трех коэффициентов уравнения может быть либо известен (задан) либо нет. Как только становятся известны все три коэффициента, уравнение автоматически решается, и в объекте сохраняются результаты решения. Как только один из коэффициентов удаляется, результаты предыдущего решения (если оно имело место) удаляются.

Спецификация

```
class qEquation
```

св-ва (переменные-члены):

float	aCoeff	коэффициент a
boolean	aIsKnown	a известен (true)
float	bCoeff	коэффициент b
boolean	bIsKnown	b известен (true)
float	cCoeff	коэффициент c
boolean	cIsKnown	c известен (true)
double	root1	1-й корень
double	root2	2-й корень
byte	status	статус (состояние): 0 – нельзя решить (хотя бы 1 коэфф. неизвестен) 1 – ур-е решено, 2 корня 2 – ур-е решено, 1 корень 3 – ур-е решено, нет корней

конструктор `qEquation()`

методы (функции-члены):

аксессуары get/set

toSolve

describe

deleteCoeff

собственно решить

вывод в консоль

удалить коэффициент (пока значения “a”, “b”, “c”)

Вопросы для обсуждения: - производственные вопросы реализации проекта.

Список источников и литературы: - может охватывать весь список из п. б.

АННОТАЦИЯ ДИСЦИПЛИНЫ

Дисциплина «Язык программирования Python» реализуется на Отделении интеллектуальных систем в гуманитарной сфере Учебно-научным центром программного и лингвистического обеспечения интеллектуальных систем в 5 и 6 семестрах.

Цели дисциплины:

- воспитать студента как квалифицированного специалиста в области разработки кроссплатформенных приложений на языке программирования Python;
- способствовать формированию у студента навыков работы самостоятельного программиста.

Задачи дисциплины:

- приобретение студентами знаний и первичного опыта, связанных с разработкой приложений на языке Python;
- дальнейшее формирование у студентов логического и алгоритмического мышления с помощью специально подобранных упражнений;
- выработка у студентов способности самостоятельно овладевать современными инструментальными средами и технологиями программирования.

Дисциплина направлена на формирование следующих компетенций:

ПК-1 – Способен разрабатывать алгоритмы обработки информации с использованием современных математических методов.

В результате освоения дисциплины (*модуля*) обучающийся должен:

Знать:

- основные свойства алгоритмов, формы записи алгоритмов, базовые алгоритмические структуры;
- синтаксис языка программирования Python, его основные конструкции их функциональность, терминологию и основные понятия, связанные с разработкой приложений на этом языке;
- встроенные типы данных и возможности для определения новых типов данных в Python;
- объектную структуру пакетов стандартной библиотеки Python, поддерживаемую ими функциональность;
- стандартные структуры данных (включая списки, стеки, очереди, деки, деревья, графы) и варианты их реализации в программах на языке Python.
- основные принципы использования интегрированной среды разработки (IDE) NetBeans программных приложений на языке Python.

Уметь:

- грамотно разрабатывать алгоритмы и их программные решения, комплексно используя применяемые для разработки программных приложений средства;
- эффективно использовать руководства пользователя (users guides/manuals), справочники по языкам (language references), стандарты и др. источники для освоения и применения средств программирования на Python.

Владеть навыками:

- анализа документации к программным системам
- самостоятельного специалиста в области программной реализации прикладных приложений.

Общая трудоемкость освоения дисциплины составляет 5 зачетных единиц.

