

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Владивостокский государственный университет»

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

ОП.14 Язык программирования Python

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

Форма обучения: *очная*

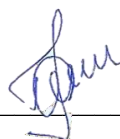
Владивосток 2023

Рабочая программа учебной дисциплины *ОП.14 Язык программирования Python* разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности / профессии *09.02.07 Информационные системы и программирование* утвержденного приказом Минобрнауки России от 09.12.2016 № 1547, *примерной образовательной программой*.

Разработчик(и): *Г.С. Завалин, преподаватель*

Рассмотрено и одобрено на заседании педагогического совета ИТ-колледжа
Протокол № 4 от «22» мая 2023 г.

Председатель педагогического совета _____



Ю. С. Кравченко

СОДЕРЖАНИЕ

- 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ
УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 3 УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

1 ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1 Место дисциплины в структуре основной образовательной программы

Учебная дисциплина *ОП.14 Язык программирования Python* является частью *общеобразовательного* учебного цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по специальности/профессии *09.02.07. Информационные системы и программирование*.

1.2 Цель и планируемые результаты освоения дисциплины

По итогам освоения дисциплины, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

| Код компетенции | Умения | Знания |
|-----------------|--|--|
| ОК 01 ОК 02 | <ul style="list-style-type: none">– составлять алгоритмы для решения задач и реализовывать эти алгоритмы на компьютере в виде программ, написанных на языке Python;– работать в интерактивной среде программирования и использовать её основные возможности для отладки и тестирования программ;– создавать и отлаживать несложные приложения на языке программирования Python;– использовать важнейшие стандартные структуры данных. | <ul style="list-style-type: none">– создание компьютерных программ, и современными концепциями программирования;– особенности и возможности языка программирования Python и области его применения;– основ функционального программирования на языке Python;– основы объектно-ориентированного программирования на языке Python;– базовые элементы языка Python. |

2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1 Объем учебной дисциплины и виды учебной работы

| Вид учебной работы | Объем часов |
|---|-------------|
| Объем образовательной программы учебной дисциплины | 88 |
| в том числе: | |
| – теоретическое обучение | 44 |
| – практические занятия | 44 |
| – лабораторные занятия | - |
| – курсовая работа (проект) | - |
| – самостоятельная работа | - |
| – консультации | - |
| – промежуточная аттестация – дифференцированный зачет | |

2.2 Тематический план и содержание учебной дисциплины

| Наименование разделов и тем | Содержание учебного материала и формы организации деятельности обучающихся | Объем в часах | Коды компетенций, формированию которых способствует элемент программы |
|---|---|---------------|---|
| <i>1</i> | <i>2</i> | <i>3</i> | <i>4</i> |
| Раздел 1. Искусство программирования | | | |
| Тема 1.1. Введение в дисциплины | Содержание учебного материала | | |
| | 1. О дисциплине. О языке программирования Python. Установка Python на компьютер. Использование в терминале. IDE и редакторы для разработки на Python. | 2 | OK 01 OK 02 |
| Тема 1.2. Основные элементы программирования | Содержание учебного материала | | |
| | 1. The Zen of Python. Литеральные константы. Типы данных. Операторы и выражения. Переменные. Краткая запись операций. Комментарии. Встроенные функции. Методы. Форматирование строки. Логические и физические строки, отступы. Операторы сравнения. Логические операторы. Операторы ветвления. Оператор цикла, операторы прерывания цикла и итерации. | 6 | OK 01 OK 02 |
| Тема 1.3. Практическая работа | Содержание учебного материала | | |
| | 1. Написание программы с консольным интерфейсом, поддерживающая ввод пользователя. | 2 | OK 01 OK 02 |
| Тема 1.4. Как писать программы | Содержание учебного материала | | |
| | 1. Цикл for. Функции. Аргументы функции. Значения аргументов по умолчанию. Возврат значений. Локальные переменные. Глобальные переменные. Документация. Ключевые аргументы. Нелокальные переменные. Модули и пакеты. Создание собственных модулей. Просмотр содержимого. Установка модулей и пакетов. | 6 | OK 01 OK 02 |
| | Практическая работа | | |
| | 1. Написание программы с консольным интерфейсом, написание логики программы, использование функций для декомпозиции задачи. | 2 | OK 01 OK 02 |
| Наименование | Содержание учебного материала и формы организации деятельности | Объем | Коды компетенций, |

| разделов и тем | обучающихся | в часах | формированию которых способствует элемент программы |
|---|---|---------|---|
| Тема 1.5. Сложные типы данных | Содержание учебного материала | | |
| | 1. Структуры данных. Список, методы списков. Кортж. Словарь, методы словарей. Переменное число аргументов функций. Множество, методы множеств. Последовательности, операции над последовательностями. Ссылки. Функция zip, генераторы списков, итераторы, функции-генераторы. | 6 | ОК 01 ОК 02 |
| | Практическая работа | | |
| | 1. Улучшение программы из предыдущей темы. | 2 | ОК 01 ОК 02 |
| Тема 1.6. Работа с файлами | Содержание учебного материала | | |
| | 1. Режимы работы с файлами: чтение, запись. Атрибуты файлового объекта. Работа с бинарными файлами. Менеджер контекста. Работа с файловой системой. | 2 | ОК 01 ОК 02 |
| | Практическая работа | | |
| | 1. Добавление функционала по работе с файлами в проект из предыдущей темы. | 2 | ОК 01 ОК 02 |
| Тема 1.7. Работа над ошибками | Содержание учебного материала | | |
| | 1. Ошибки и исключения. Ошибки разбора кода. Ошибки времени исполнения кода. Иерархия, описания, обработка принудительное поднятие исключений. Финализация. Изучение конструкции try-except-finally. | 4 | ОК 01 ОК 02 |
| Тема 1.8. Регулярные выражения | Содержание учебного материала | | |
| | 1. Регулярные выражения. Синтаксис шаблонов. Флаги. Модуль re и основные его методы. | 4 | ОК 01 ОК 02 |
| | Практическая работа | | |
| | 1. Чтение файла, выбор нужной информации и запись результата в новый файл. Обработка электронной почты, номера телефона и наборы текстовых флагов. | 2 | ОК 01 ОК 02 |

| Наименование разделов и тем | Содержание учебного материала и формы организации деятельности обучающихся | Объем в часах | Коды компетенций, формированию которых |
|-----------------------------|--|---------------|--|
|-----------------------------|--|---------------|--|

| | | | способствует элемент программы | |
|---|--------------------------------------|---|--------------------------------|----------------|
| Тема 1.9. Элементы функционального программирования | Содержание учебного материала | | | |
| | 1. | Функциональное программирование. Декларативный подход. Lambda выражения. Функции map, filter, reduce. Замыкания. Композиция функций. Декораторы. Кэширование результатов функций. Каррирование функций. Преобразование функции в набор вложенных функций. | 8 | OK 01 OK 02 |
| Тема 1.10. Объектно-ориентированное программирование | Содержание учебного материала | | | |
| | 1. | Написание программы с консольным интерфейсом, поддерживающая пользовательский ввод и основывающаяся на взаимодействии классов внутри программы. | 6 | OK 01 OK 02 |
| | Практическая работа | | | |
| | 1. | Создание классов и объектов. | 2 | OK 01 OK 02 |
| Тема 1.11. Практическая работа | Практическая работа | | | |
| | 1. | Создание консольной игры с помощью ООП | 4 | OK 01 OK 02 |
| Тема 1.12. Объектно-ориентированный дизайн | Содержание учебного материала | | | |
| | 1. | Принципы SOLID. Single Responsibility, Open/Closed, Liskov's Substitution. Interface Segregation, Dependency Inversion. Взаимосвязи классов: Обобщение, реализация, ассоциация, агрегация, композиция. | 6 | OK 01 OK 02 |
| Тема 1.13. Практическая работа | Практическая работа | | | |
| | 1. | Работа с абстрактными классами. | 2 | OK 01 OK 02 |
| Тема 1.14. Сложность алгоритмов | Содержание учебного материала | | | |
| | 1. | Изучение описания предельного поведения функций. Асимптотический анализ. Сложность алгоритмов. Линейный и бинарный поиск (метод половинного деления). | 4 | OK 01 OK 02 |

| Наименование | Содержание учебного материала и формы организации деятельности | Объем | Коды компетенций, |
|--------------|--|-------|-------------------|
|--------------|--|-------|-------------------|

| разделов и тем | обучающихся | в часах | формированию которых способствует элемент программы |
|---|--------------------------------------|---|---|
| Тема 1.15. Рекурсия | Содержание учебного материала | | ОК 01 ОК 02 |
| | 1. | Что такое рекурсия. Алгоритм Евклида. Бинарный поиск. Факториал. Сортировка слиянием | |
| Тема 1.16. Связный список | Содержание учебного материала | | ОК 01 ОК 02 |
| | 1. | Что такое связный список. Возможные операции над списками. | |
| Тема 1.17. Стек и очереди | Содержание учебного материала | | ОК 01 ОК 02 |
| | 1. | Что такое стек. Реализация стека. Операции над стеком. Что такое очередь. Реализация очереди. Операции над очередью | |
| Тема 1.18. Практическая работа | Практическая работа | | ОК 01 ОК 02 |
| | 1. | Написание программы, которая решает задачу, используя новые изученные типы данных | |
| Тема 1.19. Практическая работа | Практическая работа | | ОК 01 ОК 02 |
| | 1. | Написание программы, которая решает задачу, используя стек, а также, обрабатывающая различные ошибки | |

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрено наличие следующих специальных помещений:

Лаборатория: Программного обеспечения компьютерных сетей, программирования и баз данных.

Основное оборудование: Рабочее место для преподавателя, оборудованное ПК; Мультимедийный комплект (проектор Casio XJ-V2, экран Lumien Eco Picture); Компьютеризированные посадочные места с выходом в интернет; Стулья; ЖК панель; Маркерная доска.

Программное обеспечение: 1. Microsoft Windows 7 Professional (ООО "Пасифик Компьютеры Групп", ГК №55 от 03.05.2011 г., лицензия №48467770 от 06.05.2011 г.). 2. Интегрированные среды программирования Python 3.3 - Python 3.5 и выше. Google Chrome (свободное). Adobe Acrobat Reader (свободное).

3.2 Информационное обеспечение обучения (перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы)

Основная литература

1. Л.Н. Горбунов и др. УМК Здоровье и безопасность детей в мире компьютерных технологий и Интернет. Электронное приложение. Москва. САЛОН-ПРЕСС. 2010;

2. Федеральный портал Единая коллекция образовательных ресурсов (school-collection.edu.ru/)

3. Домашняя страница Python www.python.org. Справочные материалы, официальная документация;

4. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>;

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

| Результаты обучения | Критерии оценки | Методы оценки |
|---|--|---------------------|
| - структура программы на Python | Способность выполнить простейшую программу в интерактивной среде | Практическая работа |
| - типы данных | Способность использовать различные структуры данных для решения задач | Практическая работа |
| - формат записи цикла с параметром | Способность определять вид цикла, наиболее удобный для решения поставленной задачи | Практическая работа |
| - принципы структурного программирования | Способность создавать сложные условия с помощью логических операторов | Практическая работа |
| - принципы объектно-ориентированного программирования | Способность создавать классы, описывающие различные сущности | Практическая работа |

Для оценки достижения запланированных результатов обучения по дисциплине разработаны контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации, которые прилагаются к рабочей программе дисциплины.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования

«Владивостокский государственный университет»

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА

для проведения текущего контроля и промежуточной аттестации
по учебной дисциплине

ОП.14 Язык программирования Python

программы подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование

Форма обучения: *очная*

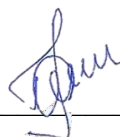
Владивосток 2023

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации по учебной дисциплине *ОП.14 Язык программирования Python* разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности / профессии *09.02.07 Информационные системы и программирование* утвержденного приказом Минобрнауки России от *09.12.2016 № 1547*, примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик(и): *Г.С. Завалин, преподаватель*

Рассмотрено и одобрено на заседании педагогического совета ИТ-колледжа
Протокол № 4 от «22» мая 2023 г.

Председатель педагогического совета _____



Ю. С. Кравченко

1 Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины *ОП.14 Язык программирования Python*.

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта.

2 Планируемые результаты обучения по дисциплине, обеспечивающие результаты освоения образовательной программы

| Код ОК, ПК | Код результата обучения | Наименование результата обучения |
|----------------|-------------------------|--|
| ОК 01 ОК 02 | 31 | создание компьютерных программ в современных концепциях программирования |
| | 32 | особенности и возможности языка программирования Python и области его применения |
| | 33 | основы функционального программирования на языке Python |
| | 34 | основы объектно-ориентированного программирования на языке Python |
| | 35 | базовые элементы языка Python |
| | У1 | составлять алгоритмы для решения задач и реализовывать эти алгоритмы на компьютере в виде программ, написанных на языке Python |
| | У2 | работать в интерактивной среде программирования и использовать её основные возможности для отладки и тестирования программ |
| | У3 | создавать и отлаживать несложные приложения на языке программирования Python |
| | У4 | использовать важнейшие стандартные структуры данных |

3 Соответствие оценочных средств контролируемым результатам обучения

3.1 Средства, применяемые для оценки уровня практической подготовки

| Краткое наименование раздела (модуля) / темы дисциплины | Код результата обучения | Показатель овладения результатами обучения | Наименование оценочного средства и представление его в КОС | |
|---|-------------------------|--|--|--|
| | | | Текущий контроль | Промежуточная аттестация |
| Раздел 1. Искусство программирования | | | | |
| Тема 1.1. Введение в дисциплины | 31 | Способность создать простейшие программы с помощью языка программирования Python | <i>Устный опрос (п. 5.1, вопросы 1-3)⁵</i> | <i>Вопросы на экзамен 1-6 (п. 6.1)⁵</i> |
| | 35 | Способность оперировать операциями ввод-вывода | <i>Устный опрос (п. 5.1, вопросы 4-7) Реферат (п. 5.2, темы 1-7)</i> | <i>Вопросы на экзамен 7,8 (п. 6.1)</i> |
| | У1 | Способность создавать программы в интерактивной среде разработки | <i>Тест № 1 (п.5.3, варианты 1-3)</i> | <i>Практическое задание к экзамену 1 (...)</i> |
| | У2 | Способность организовывать работу в среде разработки | | |
| Тема 1.2. | 32 | Способность создавать и | <i>Контрольная</i> | <i>Практическое</i> |

| | | | | |
|---|----|---|--|---|
| Основные элементы программирования | | манипулировать переменными | <i>работа № 1 (п.5.4, задания 1-3)</i> | <i>задание к экзамену 1 (...)</i> |
| | У1 | Способность использовать логические операторы и операторы сравнения | | |
| | У3 | Способность использовать встроенные функции Python | | |
| Тема 1.3. Практическая работа | 32 | Способность создавать компьютерные программы | <i>Конспект (чего)</i> | <i>Вопросы на экзамен 9-14 (п. 6.1)</i> |
| | У1 | Способность работы с консольным интерфейсом | | |
| | У2 | Способность оперировать операциями ввод-вывода | | |
| Тема 1.4. Как писать программы | 33 | Способность использовать циклы и функции при написании программ | | |
| | У1 | Способность использовать функциональный подход в программировании | | |
| | У4 | Способность устанавливать различные модули и пакеты | | |
| Тема 1.5. Сложные типы данных | 35 | Способность использовать встроенные структуры данных | | |
| | У2 | Способность создавать списки, кортежи и словари с помощью языка программирования Python | | |
| | У3 | Способность оперировать методами множеств | | |
| Тема 1.6. Работа с файлами | 32 | Способность работать с файлами в рамках программы | | |
| | У1 | Способность читать и записывать данные в файл | | |
| | У2 | Способность работать с бинарными файлами | | |
| Тема 1.7. Работа над ошибками | 35 | Способность обрабатывать ошибки при написании программы | | |
| | У2 | Способность создавать конструкции try-except-finally | | |
| | У3 | Способность обработки исключений | | |
| Тема 1.8. Регулярные выражения | 33 | Способность использовать регулярные выражения при написании программ | | |
| | У1 | Способность использовать функции модуля re | | |
| | У4 | Способность создавать флаги для регулярных выражений | | |
| Тема 1.9. Элементы функционального программирования | 33 | Способность создавать функции | | |
| | У1 | Способность использовать функции и lambda-функции | | |
| | У2 | Способность каррировать функции, использовать вложенные функции | | |
| Тема 1.10. Объектно-ориентированное | 34 | Способность проектировать программу в соответствии с ООП | | |
| | У1 | Способность использовать классы и объекты | | |

| | | | | |
|--|----|--|--|--|
| программирование | У2 | Способность использовать принципы ООП при написании программ | | |
| Тема 1.11. Практическая работа | 32 | Создание консольной игры с помощью ООП | | |
| Тема 1.12. Объектно-ориентированный дизайн | 32 | Способность создавать программы по принципам SOLID | | |
| | У1 | Способность обеспечивать взаимосвязи между классами внутри программы | | |
| | У2 | Способность работать с классами и объектами | | |
| Тема 1.13. Практическая работа | 32 | Способность следовать принципам ООП | | |
| | У1 | Способность работать в абстрактными классами | | |
| Тема 1.14. Сложность алгоритмов | 32 | Способность применять асимптотический анализ | | |
| | У1 | Способность применять линейный и бинарный поиск | | |
| | У2 | Способность обрабатывать предельное поведение функций | | |
| Тема 1.15. Рекурсия | 32 | Способность применять рекурсивные алгоритмы | | |
| | У1 | Способность подсчета факториала | | |
| | У2 | Способность использовать сортировку слиянием | | |
| Тема 1.16. Связный список | 32 | Способность применять связные списки | | |
| | У1 | Способность использования связных списков в рамках программы | | |
| Тема 1.17. Стек и очереди | 32 | Способность использовать стек и его операции | | |
| | У1 | Способность реализовывать очередь | | |
| | У2 | Способность реализовывать операции очереди | | |
| Тема 1.18. Практическая работа | У4 | Способность писать программы с использованием различных тип данных | | |
| Тема 1.19. Практическая работа | У3 | Способность писать программы с использованием стека и очереди | | |

4 Описание процедуры оценивания

Результаты обучения по дисциплине, уровень сформированности компетенций оцениваются по четырём бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Текущая аттестация по дисциплине проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная

дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по дисциплине результатов обучения, обеспечивающих результаты освоения образовательной программы в целом.

Критерии оценивания письменной работы

(оценочные средства: практическая работа).

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы. Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии выставления оценки студенту на зачете/ экзамене

(оценочные средства: практическая работа)

| Оценка по промежуточной аттестации | Характеристика качества сформированности компетенций |
|------------------------------------|--|
| «зачтено» / «отлично» | Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач. |
| «зачтено» / «хорошо» | Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, |

| | |
|---|---|
| | затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации. |
| «зачтено» / «удовлетворительно» | Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации. |
| «не зачтено» / «неудовлетворительно» | Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы. |

5. Примеры оценочных средств для проведения текущей аттестации

5.1 Примеры заданий для выполнения практических работ:

Тема 1.1. Введение в дисциплины.

Ввести с клавиатуры два числа. Каждое из введенных чисел присвоить переменным.

Необходимо поменять значения переменных так, чтобы значение первой переменной оказалось во второй, а второй - в первой.

Тема 1.2. Основные элементы программирования

Попробуйте написать программу для подготовки к контрольной точке. Игрок находится в замке. Перед ним 3 двери, за одной из них приведение. Предложение для игрока: Какую дверь ты открываешь: 1, 2 или 3? Игрок выбирает дверь. Если за дверью никого нет, игрок получает очко и переходит к пункту 3. Если за дверью приведение, игра заканчивается. В конце игры показываем игроку количество очков.

Тема 1.3. Практическая работа

Написать программу определения размера стипендии. Пользователь вводит 3 натуральных числа: общее количество оценок, количество пятерок, количество четверок. Если все пятерки – стипендия 6000 руб., если одна четверка – 4500 руб., если две четверки – 3750 руб., если нет троек – 3000 руб., иначе – нет стипендии.

Тема 1.4. Как писать программы

Написать игру "Виселица на поле чудес" для консоли. У вас есть слово, например, «олень». Необходимо отрисовать для игрока какие-либо символы, например, \u25A0 по количеству букв в выбранном слове. Необходимо установить счётчик «жизни» в какое-либо значение, например, 3. Может быть количество «жизни» будет равно длине слова. Предложить игроку ввести букву или всё слово целиком. Если буква правильная, то слово из символов перерисовывается с видимой буквой (или буквами). Если буква неправильная, то у игрока отнимается одна «жизнь». Если игрок ввёл слово и это слово правильно, либо это последняя правильная буква, либо у игрока закончились «жизни», то игра заканчивается.

Тема 1.5. Сложные типы данных.

Усовершенствовать игру «Виселица на поле чудес. Теперь у вас есть список слов с их описанием, из которого случайным образом выбирается слово для игры. Теперь игроку надо показать не только символы, но и описание. Предмет одежды, который носят на ногах. Предложите игроку после каждого тура сыграть ещё или отказаться. Пусть игра идёт до тех пор, пока не закончатся слова в списке

Тема 1.6. Работа с файлами

Документ article.txt содержит следующий текст: Вечерело, Жужжали мухи, Светил фонарик, Кипела вода в чайнике, Венера зажглась на небе, Деревья шумели, Тучи

разошлись, Листва зеленела. Требуется реализовать функцию `longest_words(file)`, которая выводит слово, имеющее максимальную длину (или список слов, если таковых несколько).

Тема 1.7. Работа над ошибками

Написать программу, которая запрашивает у пользователя имя файла, зачитывает этот файл и считает сумму всех чисел в файле. Отследите все возможные ошибки, которые могут произойти в коде. Ваша программа должна работать корректно и не падать ни при каких условиях. Обратите внимание, что именно это и является главной задачей данной практической работы, а не то, что конкретно происходит с результатом чтения файла.

Тема 1.8. Регулярные выражения

На железнодорожном вокзале ведётся журналирование рейсов. Каждый рейс в файле журнала занимает одну строку в формате: Рейс номер-поезда прибыл/отправился из/в город в время. В файле журнала может присутствовать и другая информация. Необходимо зачитать файл журнала. Выбрать необходимую информацию. Представить информацию в виде: [время] - Поезд № номер-поезда из/в город

Тема 1.9. Элементы функционального программирования

Напишите прототип программы последовательно (расширяя программу) добавляя необходимый функционал: чтение с клавиатуры и вывод в терминал, чтение из файла и вывод в терминал, чтение с клавиатуры и отправка данных по сети, чтение из файла и отправка данных по сети, чтение с клавиатуры и запись в файл, получение данных из сети и вывод в терминал, получение данных из сети и запись в файл. Точкой входа в программу должна быть одна функция `main`.

Тема 1.10. Объектно-ориентированное программирование

Создайте класс `User` и его наследника класс `SuperUser`, которые описывают пользователя и супер-пользователя. В классе `User` необходимо описать: конструктор, который принимает в качестве параметров значения для атрибутов `name`, `login` и `password`, свойства для изменения и получения значений атрибутов, метод `show_info`, который печатает в произвольном формате значения атрибутов `name` и `login`, атрибут класса `count` для хранения количества созданных экземпляров класса `User`.

Необходимые условия, которые надо учесть: атрибут `name` доступен и для чтения, и для изменения, атрибут `login` доступен только для чтения, атрибут `password` доступен только для изменения.

В классе `SuperUser` необходимо описать: конструктор, который принимает в качестве параметров значения для атрибутов `name`, `login`, `password` и `role`, свойство для изменения и получения значения атрибута `role`, метод `show_info`, который печатает в произвольном

формате значения атрибутов `name`, `login` и `role`, атрибут класса `count` для хранения количества созданных экземпляров класса `SuperUser`.

Тема 1.11. Практическая работа

Описать схватку двух бойцов. Игра должна быть реализована в рамках объектно-ориентированного программирования.

Тема 1.12. Объектно-ориентированный дизайн

Напишите программу с классом `Math`. Создайте два атрибута — `a` и `b`. Напишите методы `addition` — сложение, `multiplication` — умножение, `division` — деление, `subtraction` — вычитание. При передаче в методы параметров `a` и `b` с ними нужно производить соответствующие действия и печатать ответ.

Тема 1.13. Практическая работа

Напишите программу с классом `Student`, в котором есть три атрибута: `name`, `groupNumber` и `age`. По умолчанию `name = Ivan`, `age = 18`, `groupNumber = 10A`. Необходимо создать пять методов: `getName`, `getAge`, `getGroupNumber`, `setNameAge`, `setGroupNumber`. Метод `getName` нужен для получения данных об имени конкретного студента, метод `getAge` нужен для получения данных о возрасте конкретного студента, метод `setGroupNumber` нужен для получения данных о номере группы конкретного студента. Метод `setNameAge` позволяет изменить данные атрибутов, установленных по умолчанию, метод `setGroupNumber` позволяет изменить номер группы, установленный по умолчанию. В программе необходимо создать пять экземпляров класса `Student`, установить им разные имена, возраст и номер группы.

Тема 1.14. Сложность алгоритмов

Попробуйте самостоятельно описать алгоритм бинарного поиска на Python.

Тема 1.15. Рекурсия

Напишите рекурсивную функцию `sum_sub(list)`, которая будет принимать список целых чисел. Эта функция будет суммировать все нечётные числа и вычитать все чётные числа. В конце она будет возвращать получившееся значение.

Тема 1.16. Связный список

Традиционно списки выводятся на печать в скобках, с запятыми между элементами: `[1, 2, 3]`. Добавьте в класс `LinkedList` метод `print_list` так, чтобы он возвращал список в таком формате. Добавьте в класс `LinkedList` метод `last`, который вернет последний узел в списке или `None` в случае, когда список пуст. Протестируйте работу метода для пустого и непустого списков.

Тема 1.17. Стек и очереди

В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трёх и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться.

Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас «уничтожено». Непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Тема 1.18. Практическая работа

К тупику со стороны Пути 1 подъехал поезд. Необходимо сделать так, чтобы вагоны поезда попали на Путь 2 по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика). Условия: вводятся числа через пробел – номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика, вагоны пронумерованы натуральными числами от 1, каждое из которых встречается ровно один раз без. «Дырок» в нумерации нет (то есть, не может быть четыре вагона с номерами 1245). При этом известно, в каком порядке изначально идут вагоны поезда.

Тема 1.19. Практическая работа

Создайте и опишите класс `Interpreter`: конструктор класса должен принимать код в виде строки, метод `evaluate` единственный публичный метод, который доступен пользователю интерпретатора. Он должен вернуть результат выражения.