МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Луганский государственный университет имени Владимира Даля»

Экономический факультет
Кафедра экономической кибернетики и прикладной статистики

УТВЕРЖДАЮ:
Декан экономического факультета
_____ Тхор Е.С.
(подпись)
«24» апреля 2023 года

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНЫХ ПРОДУКТОВ»

По направлению подготовки 38.03.05 Бизнес-информатика
Профиль: «Информационная бизнес-аналитика»

Луганск – 2023

Лист согласования РПУД

Рабочая программа учебной дисциплины «Технологии разработки программных продуктов» по направлению подготовки 38.03.05 Бизнес-информатика. – 48 с.

Рабочая программа учебной дисциплины «Технологии разработки программных продуктов» составлена с учетом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 38.03.05 Бизнес-информатика, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 29 июня 2020 года № 838.

СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):
к.т.н., доц. Велигура А.В.

Рабочая программа дисциплины утверждена на заседании кафедры экономической кибернетики и прикладной статистики «18» 04 2023 г., протокол № 26

Заведующий кафедрой экономической кибернетики
и прикладной статистики _____ А.В. Велигура

Переутверждена: «___» _____20___ г., протокол № _____

Согласована (для обеспечивающей кафедры):
Декан экономического факультета _____ Тхор Е.С.

Переутверждена: «___» _____20____ года, протокол № _____

Рекомендована на заседании учебно-методической комиссии экономического факультета «21» апреля 2023 г., протокол № 4.

Председатель учебно-методической
комиссии экономического факультета _____ Е.Н. Шаповалова

© Велигура А.В., 2023 год
© ФГБОУ ВО «ЛГУ им. В. Даля», 2023 год

Структура и содержание дисциплины

1. Цели и задачи дисциплины, ее место в учебном процессе

Цель изучения дисциплины – изучение и практическое освоение общих принципов и современных методов технологии программирования.

Задачи:

знать теоретические основы и современные информационные технологии анализа, проектирования и разработки программного обеспечения;

уметь проектировать и разрабатывать различные виды программного обеспечения на основе объектно-ориентированного подхода;

иметь опыт разработки программ средней сложности;

иметь представление о библиотеках классов и инструментальных средствах, применяемых при разработке программного обеспечения.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Технологии разработки программных продуктов» относится обязательной части блока дисциплин.

Необходимыми условиями для освоения дисциплины являются: навыки работы с компьютером как средством обработки информации, знания современных методов и средств информационных технологий, владение культурой мышления, навыками анализа и моделирования, владение навыками алгоритмизации и императивного программирования, владение навыками работы с системами программирования на этапах разработки и отладки программного кода.

Содержание дисциплины является логическим продолжением содержания дисциплин «Введение в специальность», «Бизнес-информатика», «Бизнес-информатика 2», «Дискретный анализ в экономике», «Логические основы систем управления в экономике», «Математика» и служит основой для освоения дисциплин «Вычислительные системы, сети, телекоммуникации», «Системы, основанные на знаниях», «Информационные системы и технологии в управленческой деятельности», «Интеллектуальный анализ данных», «Технология проектирования и администрирования баз данных», «Информационно-коммуникационные технологии в бизнесе» и учебной практики.

3. Требования к результатам освоения содержания дисциплины

Код и наименование компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов
ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере	ОПК-3.1. Способен определять этапы создания и использования продуктов	**Знать:** Методологии формулирования требований к продуктам, технологии внедрения и

информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации		использования продуктов в сфере информационно-коммуникационных технологий
		Уметь: осуществлять создание и использование продуктов в сфере информационно-коммуникационных технологий на всех этапах их жизненного цикла
		Владеть: навыками применения инструментальных средств создания и использования продуктов в сфере информационно-коммуникационных технологий
	ОПК-3.3. Способен разрабатывать алгоритмы и программы для практической реализации процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий	**Знать:** технологии разработки алгоритмов и программ для практической реализации процессов создания и использования продуктов
		Уметь: разрабатывать алгоритмы и программы для практической реализации процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий
		Владеть: навыками применения инструментальных средств для практической реализации процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (зач. ед.)		
	Очная форма	**Очно-заочная форма**	**Заочная форма**
Общая учебная нагрузка (всего)	**180 (5 зач. ед)**	**216 (6 зач. ед)**	**216 (6 зач. ед)**
Обязательная аудиторная учебная нагрузка (всего) в том числе:	**102**	**42**	**18**
Лекции	34	14	6
Семинарские занятия	-		
Практические занятия	34	14	6
Лабораторные работы	34	14	6
Курсовая работа (курсовой проект)	36	36	36
Другие формы и методы организации образовательного процесса (*расчетно-графические работы, групповые дискуссии, ролевые игры, тренинг, компьютерные симуляции, интерактивные лекции, семинары, анализ деловых ситуаций и т.п.)*			
Самостоятельная работа студента (всего)	**78**	**174**	**198**
Форма аттестации	зачет, курсовая	зачет, курсовая	зачет, курсовая

4.2. Содержание разделов дисциплины

Тема 1. ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Знакомство со средой разработки IDLE. Структура программы. Комментарии. Основные типы данных и операций в языке Python. Основные типы переменных. Преобразование типов переменных.

Тема 2. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ НА ЯЗЫКЕ PYTHON

Операторы ввода и вывода данных. Оператор присваивания. Встроенные функции и методы для работы с числами. Модуль Math. Математические функции. Операторы для работы с последовательностью. Приоритет выполнения операторов. Генерация случайных чисел. Модуль random.

Тема 3. ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ И ЦИКЛИЧЕСКИХ ПРОЦЕССОВ НА ЯЗЫКЕ PYTHON

Операции сравнения. Операторы условного перехода. Операторы цикла. Функции range() и enumerate(). Операторы перехода на следующую итерацию и прерывания цикла. Вложенные циклы.

Тема 4. ОПЕРАЦИИ НАД СТРОКАМИ В ЯЗЫКЕ PYTHON

Строки и двоичные данные. Создание строки. Специальные символы. Операции над строками. Форматирование строк. Метод format(). Функции и методы для работы со строками и символами. Регулярные выражения: синтаксис, поиск по шаблону.

Тема 5. СТРУКТУРЫ ДАННЫХ ЯЗЫКА PYTHON

Структуры данных языка Python. Списки. Создание списка. Операции над списками. Многомерные списки. Перебор элементов списка. Генераторы списков и выражения-генераторы. Функции для работы со списками. Добавление и удаление элементов списка. Поиск элемента в списке и получение сведений о значениях, входящих в список. переворачивание и перемешивание списка. Выбор элемента списка случайным образом. Сортировка списка. Заполнение списка числами. Преобразование списка в строку. Кортежи. Операции над кортежами. Множества. Операции над множествами. Диапазоны. Операции над диапазонами. Словари. Создание словаря. Операции над словарями. Перебор элементов словаря. Методы для работы над словарями. Генераторы словарей.

Тема 6. ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ В ЯЗЫКЕ PYTHON

Определение функции и ее вызов. Необязательные параметры функций и сопоставление по ключам. Переменное число параметров в функции. Анонимные функции. Функции генераторы. Декораторы функций. Глобальные и локальные переменные. Рекурсивные функции. Вложенные функции.

Тема 7. МОДУЛИ И ПАКЕТЫ ЯЗЫКА PYTHON

Модули. Понятие модуля. Подключение модуля: инструкции import и from. Пути поиска модулей. Повторная загрузка модулей. Пакеты. Понятие пакета. Работа с пакетами.

Тема 8. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Понятие класса, атрибута и метода. Определение класса и создание экземпляра класса. Конструкторы и деструкторы. Наследование. Множественное наследование. Понятие примесей и их использование. Специальные методы классов. Перегрузка операторов. Статические методы и методы классов. Абстрактные методы. Ограничение доступа к идентификаторам внутри класса. Свойства классов. Декораторы классов.

Тема 9. ИТЕРАТОРЫ, КОНТЕЙНЕРЫ И ПЕРЕЧИСЛЕНИЯ В ЯЗЫКЕ PYTHON

Итератор класса. Понятие итератора класса и его использование. Контейнеры. Контейнеры- последовательности. Контейнеры словари. Перечисления. Атрибуты и методы перечислений.

Тема 10. РАБОТА С ФАЙЛАМИ И КАТАЛОГАМИ В ЯЗЫКЕ PYTHON

Открытие файла. Методы для работы с файлами. Классы и модули используемые для доступа к файлам. Права доступа к файлам и каталогам. Функции для манипулирования файлами. Преобразование пути к файлам и каталогам. Перенаправление ввода/вывода. Сохранение объектов в файл. Функции для работы с каталогами.

4.3. Лекции

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
1	Введение в программирование на языке Python	2	1	1
2	Программирование линейных алгоритмов на языке Python	2	1	1
3	Программирование разветвляющихся и циклических процессов на языке Python	2	1	1
4	Операции над строками в языке Python	4	1	1
5	Структуры данных языка Python	4	1	1
6	Пользовательские функции в языке Python	4	1	1
7	Модули и пакеты языка Python	4	2	
8	Объектно-ориентированное программирование на языке Python	4	2	
9	Итераторы, контейнеры и перечисления в языке Python	4	2	
10	Работа с файлами и каталогами в языке Python	4	2	
Итого:		**34**	**14**	**6**

4.4. Практические (семинарские) занятия

№ п/п	Название темы	Объем часов		
		Очная форма	Очно-заочная форма	Заочная форма
1	Введение в программирование на языке Python	2	1	1
2	Программирование линейных алгоритмов на языке Python	2	1	1
3	Программирование разветвляющихся и циклических процессов на языке Python	2	1	1
4	Операции над строками в языке Python	4	1	1
5	Структуры данных языка Python	4	1	1
6	Пользовательские функции в языке Python	4	1	1
7	Модули и пакеты языка Python	4	2	
8	Объектно-ориентированное программирование на	4	2	

	языке Python			
9	Итераторы, контейнеры и перечисления в языке Python	4	2	
10	Работа с файлами и каталогами в языке Python	4	2	
Итого:		**34**	**14**	**6**

4.4. Лабораторные работы

№ п/п	Название темы	Объем часов		
		Очная форма	**Очно-заочная форма**	**Заочная форма**
1	Введение в программирование на языке Python	2	1	1
2	Программирование линейных алгоритмов на языке Python	2	1	1
3	Программирование разветвляющихся и циклических процессов на языке Python	2	1	1
4	Операции над строками в языке Python	4	1	1
5	Структуры данных языка Python	4	1	1
6	Пользовательские функции в языке Python	4	1	1
7	Модули и пакеты языка Python	4	2	
8	Объектно-ориентированное программирование на языке Python	4	2	
9	Итераторы, контейнеры и перечисления в языке Python	4	2	
10	Работа с файлами и каталогами в языке Python	4	2	
Итого:		**34**	**14**	**6**

4.6. Самостоятельная работа студентов

№ п/п	Название темы	Вид СРС	Объем часов		
			Очная форма	**Очно-заочная форма**	**Заочная форма**
1	Введение в программирование на языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	2	13	16
2	Программирование линейных алгоритмов на языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	13	16
3	Программирование разветвляющихся и циклических процессов на языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	13	16
4	Операции над строками в языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	13	16
5	Структуры данных языка Python	Подготовка к практическим занятиям и лабораторным	4	13	16

		работам, к текущему и промежуточному контролю знаний и умений.			
6	Пользовательские функции в языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	13	16
7	Модули и пакеты языка Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	14	16
8	Объектно-ориентированное программирование на языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	14	16
9	Итераторы, контейнеры и перечисления в языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений.	4	14	15
10	Работа с файлами и каталогами в языке Python	Подготовка к практическим занятиям и лабораторным работам, к текущему и промежуточному контролю знаний и умений	4	14	15
11	Курсовая работа		36	36	36
12	Зачет	Подготовка к зачету	4	4	4
Итого:			**78**	**174**	**198**

4.7. Курсовые работы/проекты.

В соответствии с учебным планом предусмотрена курсовая работа, направленная на закрепление и углубление практических и теоретических знаний, полученных в процессе изучения дисциплины. Курсовая работа является одной из форм текущего контроля самостоятельной работы студента, представляет собой самостоятельную разработку некоторой темы по изучаемой дисциплине.

Целью выполнения курсовой работы является систематизация, закрепление и углубление знаний и компетенций, приобретенных при изучении курса «Технологии разработки программных продуктов», формирование навыков построения блок-схем алгоритмов, разработки программ на языке высокого уровня и самостоятельного творческого решения профессиональных задач.

Задачи курсовой работы:

развитие умений создавать блок-схемы алгоритмов и программ для решения различного рода задач на языке высокого уровня;

овладение навыками исследовательской работы на основе вычислительного эксперимента с моделями процессов функционирования несложных объектов и систем;

формирование у студентов элементов творческого мышления;

развитие умения создавать законченные приложения на языке высокого уровня на примере создания прикладного программного продукта в рамках сферы деятельности выпускника по направлению «Бизнес-информатика».

При выполнении курсовой работы студент должен продемонстрировать способности к таким видам деятельности, как:

поиск требуемой информации по теме;

изучение и критический анализ полученных материалов;

систематизация и обобщение имеющейся информации;

самостоятельное определение путей решения поставленных задач;

оформление решения задач в виде пояснительной записки;

логическое обоснование и формулировка выводов, предложений и рекомендаций по результатам работы.

В результате выполнения курсовой работы студент должен

научиться:

создавать программу на языке высокого уровня в соответствии с основными этапами ее разработки;

строить схему алгоритма работы программы в соответствии с требованиями ГОСТ 19.701-90;

грамотно тестировать программу;

анализировать результаты работы программы и делать выводы.

Для выполнения курсовой работы студент получает от преподавателя индивидуальное задание – предметную область, в рамках которой выполняются все этапы работы.

Курсовая работа выполняется в соответствии с методическими указаниями и включает следующие этапы:

1. Разработка развернутого технического задания на программный продукт;

2. Анализ задания, выбор технологию проектирования и разработка проекта программного продукта;

3. Выбор структуры данных для решения задачи в рамках предметной области программного продукта;

4. Разработка интерфейса пользователя;

5. Выбор стратегии тестирования и разработка тестов;

6. Выбор языка и среды программирования;

7. Разработка алгоритма и его реализация в выбранной среде разработки;

8. Тестирование и отладка;

9. Разработка необходимой документации, указанной в техническом задании.

Каждый этап включает самостоятельную работу, выполняемую перед аудиторным занятием, и работу на аудиторном занятии. Результаты выполнения заданий оформляются с помощью текстового редактора. Некоторые задания предполагают моделирование и анализ с помощью инструментальных средств. Модели в виде диаграмм, созданных при помощи инструментальных средств автоматизированного проектирования программных продуктов, вставляются в отчет виде скриншотов или распечатываются непосредственно из программы.

При защите курсовой работы студент рассказывает о полученных результатах, дает пояснения, отвечает на вопросы преподавателя.

5. Образовательные технологии

Преподавание дисциплины ведется с применением следующих видов образовательных технологий*:*

Информационные технологии: использование электронных образовательных ресурсов (электронный конспект, размещенный на сайте кафедры ecpsdahl.ru) при подготовке к лекциям, практическим занятиям и лабораторным работам.

Мультимедийные технологии: презентации к лекциям.

6. Формы контроля освоения дисциплины

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем(ями), ведущими лабораторные работы по дисциплине в следующих формах*:*

собеседование (устный или письменный опрос);

контрольная работа;

тесты.

Фонды оценочных средств, включающие типовые задания, контрольные работы, тесты и методы контроля, позволяющие оценить результаты текущей и промежуточной аттестации обучающихся по данной дисциплине, помещаются в приложении к рабочей программе в соответствии с «Положением о фонде оценочных средств».

Форма аттестации по результатам освоения дисциплины проходит в форме письменного/устного зачета/экзамена (включает в себя ответ на теоретические вопросы и решение задач) либо в сочетании различных форм (компьютерного тестирования, решения задач и пр.). Студенты, выполнившие 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачетную книжку выставляются оценки по шкале, приведенной в таблице.

Шкала оценивания	Характеристика знания предмета и ответов	Зачеты
отлично (5)	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

7. Учебно-методическое и информационное обеспечение дисциплины:

а) основная литература:

1. Северенс Ч. Введение в программирование на Python: Учебная литература для ВУЗов Национальный Открытый Университет «ИНТУИТ», 2016 http://biblioclub.ru/index.php?page=book_red&id=429184

2. Прохоренок Н., Дронов В. Python 3. Самое необходимое.: БХВ-Петербург, 2016 https://bhv.ru/product/python-3-samoe-neobhodimoe/

3. Федоров Д.Ю. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ PYTHON. Учебное пособие для прикладного бакалавриата: Гриф УМО ВОМ.:Издательство Юрайт, 2018 https://biblio-online.ru/book/1EE056CF-F11A-4C18-8D33-40B703D49AC5

4. Златопольский Д.М. Основы программирования на языке Python: Учебники Издательство "ДМК Пресс", 2017 https://e.lanbook.com/book/97359

б) дополнительная литература:

1. Хахаев И. А. Практикум по алгоритмизации и программированию на Python: курс М.: Национальный Открытый Университет «ИНТУИТ», 2016//ЭБС «Университетская библиотека online» http://biblioclub.ru/index.php?page=book&id=429256

2. Тузовский А.Ф. Объектно-ориентированное программирование: Учебное пособие для прикладного бакалавриата М.: Издательство Юрайт, 2018 // ЭБС "Юрайт" https://urait.ru/bcode/451429

3. Л. Рамальо Python. К вершинам мастерства: ДМК Пресс, 2016//ЭБС издательства «Лань» https://e.lanbook.com/book/93273

в) методические указания:

1. Конспект лекций по дисциплине “Технология программирования и разработки программных продуктов” для студентов направления подготовки “38.03.05 – Бизнес-информатика. Информационная бизнес-аналитика” [Электронный ресурс] / сост. А.В. Велигура. – Луганск: ЛНУ им. В. Даля, 2018. – 300 с.
2. Методические указания к выполнению курсовой работы по дисциплине «Программирование» для студентов направления подготовки “38.03.05 – Бизнес-информатика” [Электронный ресурс] / сост. А.В. Велигура. – Луганск: ЛНУ им. В. Даля, 2018. – 29 с.
3. Методические указания к лабораторным работам по дисциплине «Технологии разработки программных продуктов» для студентов направления подготовки 38.03.05 – Бизнес-информатика [Электронный ресурс] / сост. А.В. Велигура. – Луганск: ЛНУ им. В. Даля, 2019. – 76 с.
4. Методические указания к выполнению курсовой работы по дисциплине «Технологии разработки программных продуктов» для студентов направления подготовки 38.03.05 – Бизнес-информатика [Электронный ресурс] / сост. А.В. Велигура. – Луганск: ЛНУ им. В. Даля, 2019. – 25 с.
5. Методические указания к самостоятельной работе по дисциплине «Технологии разработки программных продуктов» для студентов направления подготовки 38.03.05 – Бизнес-информатика [Электронный ресурс] / сост. А.В. Велигура. – Луганск: ЛНУ им. В. Даля, 2019. – 34 с.

г) Интернет-ресурсы:

1. Министерство образования и науки Российской Федерации – http:://минобрнауки.рф/
2. Федеральная служба по надзору в сфере образования и науки – http://obrnadzor.gov.ru/
3. Министерство образования и науки Луганской Народной Республики –https://minobr.su
4. Министерство промышленности и торговли Луганской Народной Республики – https://www.minpromlnr.su/main.php/

5. Министерство экономического развития Луганской Народной Республики – https://merlnr.su/

6. Министерство финансов Луганской Народной Республики – https://minfinlnr.su/

7. Народный совет Луганской Народной Республики – https://nslnr.su

8. Государственный комитет статистики Луганской Народной Республики – https://www.gkslnr.su/

9. Портал Федеральных государственных образовательных стандартов высшего образования – http://fgosvo.ru/

10. Федеральный портал «Российское образование» – http://www.edu.ru/

11. Информационная система «Единое окно доступа к образовательным ресурсам» – http://window.edu.ru/

12. Федеральный центр информационно-образовательных ресурсов – http://fcior.edu.ru/

13. Статистические сборники ВШЭ - https://www.hse.ru/org/hse/primarydata/

14. OpenOffice.org: Теория и практика - https://www.altlinux.org/Books:Openoffice

15. Базовый курс по OpenOffice - https://4creates.com/training/49-bazovyy-kurs-po-openoffice.html

Электронные библиотечные системы и ресурсы

16. Электронно-библиотечная система «Консультант студента» – http://www.studentlibrary.ru/

17. Электронно-библиотечная система «StudMed.ru» – https://www.studmed.ru/

Информационный ресурс библиотеки образовательной организации

18. Научная библиотека имени А. Н. Коняева – http://biblio.dahluniver.ru/

8. Материально-техническое и программное обеспечение дисциплины

Для проведения лекционных занятий требуется аудитория, оборудованная мультимедийным проектором с экраном. Для проведения лабораторных и практических занятий необходим компьютерный класс и презентационная техника (проектор, экран, компьютер/ноутбук)

Программное обеспечение:

Функциональное назначение	**Бесплатное программное обеспечение**	**Ссылки**
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Графический редактор	GIMP (GNU Image Manipulation Program)	http://www.gimp.org/ http://gimp.ru/viewpage.php?page_id=8 http://ru.wikipedia.org/wiki/GIMP
Редактор PDF	Adobe Acrobat Reader	https://get.adobe.com/ru/reader/
Аудиоплейер	VLC	http://www.videolan.org/vlc/

9. Оценочные средства по дисциплине

Паспорт
оценочных средств по учебной дисциплине
«Технологии разработки программных продуктов»

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине	Контролируемые темы учебной дисциплины, практики	Этапы формирования (семестр изучения)
1	ОПК-3	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1	Тема 1 Тема 2 Тема 3 Тема 4	3
			ОПК-3.3	Тема 5 Тема 6 Тема 7 Тема 8 Тема 9 Тема 10	3

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов	Контролируемые разделы (темы) учебной дисциплины (модуля), практики	Наименование оценочного средства[2]
1	ОПК-3	ОПК-3.1	Знать: Методологии формулирования требований к продуктам, технологии внедрения и использования продуктов в сфере информационно-коммуникационных технологий Уметь: осуществлять создание и использование продуктов в сфере информационно-коммуникационных технологий на всех этапах их жизненного цикла Владеть: навыками применения инструментальных средств создания и использования продуктов в сфере информационно-коммуникационных технологий	Тема 1 Тема 2 Тема 3 Тема 4	Собеседование (устный или письменный опрос), контрольная работа, тесты
		ОПК-3.3	Знать: технологии разработки алгоритмов и программ для практической реализации процессов создания и использования продуктов Уметь: разрабатывать алгоритмы и	Тема 5 Тема 6 Тема 7 Тема 8 Тема 9 Тема 10	Собеседование (устный или письменный опрос), контрольная работа, тесты

			программы для практической реализации процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий Владеть: навыками применения инструментальных средств для практической реализации процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий		

Оценочные средства по дисциплине «Технологии разработки программных продуктов»

Перечень вопросов (для проведения собеседования (устный или письменный опрос))

1. Компилируемые и интерпретируемые языки программирования. Примеры и сравнения.
2. Общие сведения о языке Python и особенности его стиля программирования.
3. Синтаксис и управляющие конструкции языка Python.
4. Переменные, значения и их типы. Присваивание значения.
5. Ввод значений с клавиатуры.
6. Встроенные операции и функции. Основные алгоритмические конструкции.
7. Условный оператор. Множественное ветвление. Условия равенства / неравенства.
8. Циклы и счетчики.
9. Определение функций. Параметры и аргументы. Вызовы функций. Оператор возврата return.
10. Определение класса. Методы класса.
11. Последовательности в Python. Операторы, общие для всех типов последовательностей.
12. Специальные операторы и функции для работы со списками.

13. Работа со словарями. Методы словарей.

14. Вложенные списки. Матрицы.

15. Основные стандартные модули и пакеты в Python и их импортирование.

16. Модуль Math.

17. Некоторые специализированные модули и приложения.

18. Библиотека символьной математики Sympy.

19. Создание собственных модулей и их импортирование.

20. Создание независимых exe-приложений в Python.

21. Ссылки в пакетах.

22. Пакеты и файловая система.

23. Класс File.

24. Открытие файла.

25. Методы класса для File ввода-вывода.

26. Взаимодействие с файловой системой.

27. Модуль path.

28. Объекты и файловый ввод-вывод.

29. Объявление класса

30. Управление атрибутами и методами класса

31. Объявление объектов

32. Множественное наследование

33. Заимствование свойств и методов у родительского класса

34. Приведите преимущества и недостатки библиотеки «NumPy»? Обоснуйте ответ.

35. Приведите преимущества и недостатки библиотеки «SciPy»? Обоснуйте ответ.

36. Какая библиотека содержит модули для линейной алгебры, оптимизации, интеграции и статистики? Обоснуйте ответ.

37. Приведите преимущества и недостатки библиотеки «Pandas»? Обоснуйте ответ.

38. Приведите преимущества и недостатки библиотеки «Matplotlib»? Обоснуйте ответ.

39. Приведите преимущества и недостатки библиотеки «Seaborn»? Обоснуйте ответ.

40. Приведите преимущества и недостатки библиотеки «Bokeh»? Обоснуйте ответ.

41. Приведите преимущества и недостатки библиотеки «Plotly»? Обоснуйте ответ.

42. Какая библиотека может поддерживать R, MATLAB, Perl, Julia, Arduino и REST? Обоснуйте ответ.

43. Приведите преимущества и недостатки библиотеки «SciKit-Learn»? Обоснуйте ответ.

44. Для чего предназначена библиотека «Theano»? Обоснуйте ответ.

45. У какой библиотеки существует тесная интеграция с NumPy? Обоснуйте ответ.

46 Для чего предназначена библиотека «TensorFlow»? Обоснуйте ответ.

47. Какие существуют библиотеки от Google? Обоснуйте ответ.

48. Для чего предназначена библиотека «Keras»? Обоснуйте ответ.

49. Какие возможности у библиотеки «Keras»? Обоснуйте ответ.

50. Для чего предназначена библиотека «NLTK»? Обоснуйте ответ.

51. Какие бывают применения функции map()? Обоснуйте ответ.

52. Какие бывают применения функции filter()? Обоснуйте ответ.

53. Какие бывают применения функции reduce()? Обоснуйте ответ.

54. Как происходит импорт модулей и их составляющих из пакета? Обоснуйте ответ.

55. Как решаются задачи с использованием внешних источников данных в виде текстовых и бинарных файлов? Обоснуйте ответ.

56. Как можно создать веб-приложения с использованием Python? Обоснуйте ответ.

57.Перечислите все компании использующие язык Python для разработки?

58. Можно ли программировать на Python в области компьютерного зрения? Обоснуйте ответ.

Критерии и шкала оценивания по оценочному средству собеседование (устный или письменный опрос)

Шкала оценивания (интервал баллов)	Критерий оценивания
5	собеседование (устный или письменный опрос) прошел на высоком уровне (студент в полном объеме осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
4	собеседование (устный или письменный опрос) прошел на среднем уровне (студент в целом осветил рассматриваемый вопрос, владеет профильным понятийным (категориальным) аппаратом и т.п.)
3	собеседование (устный или письменный опрос) на низком уровне (студент допустил существенные неточности, изложил материал с ошибками, не владеет в достаточной степени профильным категориальным аппаратом и т.п.)
2	собеседование (устный или письменный опрос) прошел на неудовлетворительном уровне или не представлен (студент не готов, не выполнил задание и т.п.)

Контрольная работа

Типовые контрольные задания для проведения текущего контроля

Задание 1.

1. Проработать теоретическое введение по данной теме.

2. Построение блок-схем основных алгоритмических конструкций.

3. Решение задач на построение блок-схем линейных и разветвляющихся алгоритмов.

4. Решение задач на построение блок-схем циклических и итерационных алгоритмов.

Задание 2.

1. Проработать теоретическое введение по данной теме.

2. Решение вычислительных задач. Вычисление сумм, произведений, итерационных формул. Переменные. Типы данных.

3. Программирование задач линейной структуры. Ввод и вывод данных. Преобразование типов данных.

4. Программирование задач разветвляющих алгоритмических структур.

Задание 3.

1. Проработать теоретическое введение по данной теме.

2. Программирование задач с использованием циклов с параметром.

3. Программирование задач при использовании циклов с неизвестным числом повторений.

4. Обработка данных строкового типа.

Задание 4.

1. Проработать теоретическое введение по данной теме.

2. Задачи обработки массивов — инициализация, поиск, сортировка.

3. Программирование задачи инициализации массива случайными числами. Программирование задачи простого поиска.

4. Программирование задач простых алгоритмов сортировки данных в массивах.

Задание 5.

1. Проработать теоретическое введение по данной теме.

2. Чтение и запись данных в файлах последовательного и прямого доступа.

3. Программирование задач на чтение и запись данных в текстовых файлах последовательного доступа.

4. Программирование задач на сортировку и поиск данных в типизированных файлах.

Задание 6.

1. Проработать теоретическое введение по данной теме.

2. Графическая система координат. Класс точки и массив точек.

3. Программирование задач на построение графических примитивов — отрезка, ломанной, многоугольников.

Задание 7.

1. Проработать теоретическое введение по данной теме.

2. Структура класса, поля и методы класса.

3. Программирование задач на создание структуры класса, процедур для методов класса.

4. Программирование задач на использование объектов класса пользователя.

Задание 8.

Заданы списки строк L E загруженные из текстовых файлов в компоненты Memo.

1. Описать процедуру, которая по списку L строит два новых списка: L1 - из четных длин строк и L2 из остальных элементов списка L.

2.Описать процедуру, которая вставляет в начало списка L, после 1-й строки список E.

3.Описать процедуру, которая вставляет в конец списка L перед последней строки список E.

4.Описать процедуру, которая вставляет новую строку из E после каждой строки длиннее 10 знаков списка L.

5.Описать процедуру, которая вставляет в список L новый элемент E1 перед каждым элементом.

6.Описать процедуру, которая вставляет в список L подстроку E1 перед первым вхождением подстроки E, если E входит в строку из L.

7. Описать процедуру, которая вставляет в непустой список L пару новых строк E1 и E2 перед его самой длиной строкой.

8. Описать процедуру, которая вставляет в непустой список L, строки которого пронумерованы по убыванию, новую строку E так чтобы по ее длине определить место, чтобы сохранилась упорядоченность.

9. Описать процедуру, которая удаляет из непустого списка L первые N строк.

10. Описать процедуру, которая удаляет из списка L строки четной длины, если такие есть.

11. Описать процедуру, которая удаляет из списка L все элементы равные E, если такие есть.

12. Описать процедуру, которая удаляет из непустого списка L последние N строк.

13. Описать процедуру, которая удаляет из списка L строки нечетной длины, если такие есть.10. Описать процедуру, которая удаляет из списка L строки четной длины, если такие есть.

14. Описать процедуру, которая заменяет из списка L строки четной длины, если такие есть последовательно на строки из списка E.

Задание 9.

Добавить поля в класс кольца для изменения его вида следующим образом:

1. добавляется хорда;

2. добавляется спица (радиус к центру);

3. добавляется маленькое кольцо на окружности;

4. добавляется маленькое кольцо в центре;
5. добавляется маленький треугольник на окружности;
6. добавляется маленький треугольник в центре;
7. добавляется маленький прямоугольник на окружности;
8. добавляется маленький прямоугольник в центре;
9. двойное кольцо с штриховкой внутри;
10. вместо кольца эллипс;
11. вместо кольца квадрат;
12. вместо кольца треугольник;
13. вместо кольца прямоугольник;
14. вместо кольца шестиугольник.

Задание 10.

Произвести: Программирование типовых алгоритмов для обработки числовых типов данных. Преобразование типов.

Задание 11.

Произвести: Программирование типовых алгоритмов для обработки массивов

Задание12.

Произвести: Программирование типовых алгоритмов для обработки коллекций.

Задание 13.

Произвести: Создание собственных классов в Java: свойства, методы, конструкторы

Задание 14.

Произвести: Статические методы в Java, перегрузка методов, рекурсия

Задание 15.

Произвести: Программирование типовых алгоритмов для реализации наследования и инкапсуляции.

Задание 16.

Произвести: Программирование типовых алгоритмов для реализации полиморфизма, переопределения, перегрузка методов.

Задание 17.

Произвести: Программирование типовых алгоритмов для реализации абстрактных классов и интерфейсов.

Задание 18. Составьте три варианта программы вычисления факториала с помощью операторов цикла while, for и с помощью рекурсии.

Задание 19. 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).

Задание 20. 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).

Задание 21. Составьте программу вычисления чисел Фибоначчи тремя способами с помощью цикла (for, while) и с помощью рекурсии.

Задание 22. Составьте программу перевода из десятичной системы счисления в двоичную, восьмеричную и шестнадцатеричную системы счисления.

Задание 23. 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)))
```

Задание 24. 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)
```

Задание 25. Какую задачу решает приведенный ниже оператор:

```
print(len(set(open('input.txt', 'r', encoding = 'utf8').read().split())))
```

Задание 26. 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)
```

Задание 27. Решите задачу формирования списка, являющегося пересечением двух

произвольно заданных в программе списков, введенных из файлов.

Задание 28. 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)
```

Задание 29. 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.

Задание 30. 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()

Задание 31. 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.

Задание 32. 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()
```

Задание 33. Напишите программу, которая переводить введенное пользователем десятичное числа в двоичную, восьмеричную, шестнадцатеричную системы счисления, и выведите полученные результаты.

Задание 34. 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.

Задание 35. 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]

Задание 36. 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

Задание 37. 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.

Задание 38. 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.

Задание 39. 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 104. 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 105. Гарантируется что, если задача имеет решение, то есть решение, которое содержит не более, чем 105 операций. Составить программу для решения данной задачи.

Задание 40. 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.

Задание 41. Дана фраза, в которой, кроме букв, имеются пробелы. Проверить, является ли она палиндромом без учета пробелов (например, фраза «АРГЕНТИНА МАНИТ НЕГРА» палиндромом является).

Задание 42. 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.

Задание 43. Дана строка, в которой без пробелов записано арифметическое выражение в виде суммы N натуральных чисел, например - «11+27+ … + 33».

Вычислить эту сумму.

Задание 44. 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.

Задание 45. 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.

Задание 46. 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.

Задание 47. 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.

Задание 48. 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.

Задание 49. 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 ($1 \le i \le 100$).

Выходные данные

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

Задание 50. 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: водятся два целых числа A и B.

Выходные данные: Найденные числа в порядке возрастания.

Критерии и шкала оценивания по оценочному средству контрольная работа

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Контрольная работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Контрольная работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Контрольная работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Контрольная работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Тесты

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:

а) Допустимо записывать одну инструкцию в нескольких строках, для чего можно ее заключить в пару круглых, квадратных или фигурных скобок;

б) Возможно записать несколько инструкций в одной строке, разделяя их точкой с запятой

в) Тело составной инструкции не может располагаться в той же строке, что и тело основной, если тело составной инструкции не содержит составных инструкций;

г) Комментарии к программе на языке Питон начинаются с знака #.

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

а)	print('Нет - короновирусу!')	б)	Арифметическое выражение
в)	print(2 + 3)	г)	Строка
д)	print('2' + '3')	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:

a. да;

b. 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.

10. Для решения какой задачи может быть использован следующая инструкция языка Питон 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.

11. 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.

12. 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.

13. 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.

14. 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.

15. 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.

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

а) да;

б) нет.

17. 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.

18. 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.

19. 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.

20. 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.

21. 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

22. 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.

23. 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.

24. 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.

25. 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.

26. Пусть задается последовательность натуральных чисел, заканчивающаяся нулем. Какой результат выдаст приведенный ниже 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.

27. 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.

28. 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».

29. 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».

30. 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=' ')
```

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.

31. 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.

31. 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).

32. Что выведет следующий код:

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

Приведите ответ в виде списка языка Питон.

33. 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.

34. 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

35. 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.

36. 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.

37. 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 itrable;

г) функция sorted может принимать в качестве параметра строку.

38. Какой список выведет на экран ПК команда:

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

39. 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.

40. 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.

41. 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 _____.

42. 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 _____.

43. 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.

44. 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

45. 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.

46. Существуют ли неизменяемые множества:

а) да;

б) нет.

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

48. 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.

49. 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.

50. Установите соответствие между операциями с множествами, которые создают новые множества и их описанием:

Операция	Описание
A | B	*Объединение множеств*
A & B	*Множество, элементы которого входят в A, но не входят в B*
A - B	*Элементы входят в A | B, но не входят в A & B*
A ^ B	*Пересечение множеств*

51. 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

52. Для операций A | B, A & B, A – B, A ^ 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.

53. 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.

54. 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.

55. 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).

56. 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).

57. Определитесь 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.

58. 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.

59. 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.

60. 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.

61. 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.

Критерии и шкала оценивания по оценочному средству тесты

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Тесты выполнены на высоком уровне (правильные ответы даны на 90-100% тестов)
4	Тесты выполнены на среднем уровне (правильные ответы даны на 75-89% тестов)
3	Тесты выполнены на низком уровне (правильные ответы даны на 50-74% тестов)
2	Тесты выполнены на неудовлетворительном уровне (правильные ответы даны менее чем на 50% тестов)

Курсовая работа

В процессе написания курсовой работы студент должен продемонстрировать умение работать с учебной и научной экономической литературой, навыки владения компьютерной техникой и пакетами прикладных программ моделирования бизнес-процессов, применять научную методологию в проектировании, разработке и тестировании программных продуктов.

По результатам курсовой работы оформляется отчет. Рекомендуется следующее содержание отчета:

- титульный лист;
- задание на курсовую работу;
- оглавление, включающее наименование всех разделов и пунктов с указанием номеров страниц;
- введение, в котором обосновывается актуальность темы, указываются цель и задачи исследований;
- теоретическую часть, в которой обосновывается выбранный метод решения или модель и полученные закономерности или содержатся описания примененных в работе алгоритмов, структур данных;
- исследовательскую часть, содержащую структуры и исходные данные, полученные результаты и их анализ;
- заключение с краткими выводами по результатам работы и предложениями по их использованию;
- список литературы;
− приложения.

Во введении дается обоснование актуальности работы, формулируются ее цель и задачи, а также определяются предмет и объект работы. Введение содержит краткий обзор содержания работы, описание используемых приемов и методов, обоснование практической значимости результатов.

Независимо от конкретности проблемы, каждая программа должна удовлетворять ряду требований, на которых защищается квалификация разработчика.

1. Устойчивость программы. Программа не должна терять работоспособности ни при каких, даже некорректных, действиях пользователя. Всякие действия, грозящие потерей информации, выполняются только после повторного подтверждения. Вводимая информация там, где возможно, подвергается логическому контролю.

2. Обеспечение целостности баз данных. При любых действиях пользователя базы не должны терять целостности (некорректность индексов, потеря ссылок в связях после удаления-добавления записей и т. д.).

3. Функциональная полнота. В рамках согласованного с преподавателем подмножества функций все они должны быть реализованы.

4. Терминологическая среда и интерфейс. В диалоговых средствах используются только термины, понятные пользователю, и не используются

термины разработчика («запись», «индексация» и т.д.). Появление служебных англоязычных сообщений СУБД недопустимо. Язык диалога — с соблюдением норм вежливости, цветовая гамма — по общепринятым рекомендациям.

5. Использование клавиатуры. На любом этапе нажатие любой клавиши (особенно функциональных) должно игнорироваться или вызывать предусмотренные действия (описанные в средствах помощи).

6. Порядок движения. Движение по дереву алгоритма «сверху вниз» сопровождается заголовками всех пройденных вершин; возврат возможен только на предыдущий уровень с сохранением введенной информации, выбранных пунктов меню и указателей записей.

7. Средства помощи и реклама. При запуске программы появляется рекламная заставка, отражающая суть и возможности программного средства, а также сведения об авторе.

8. Входные и выходные документы. Экранные формы для ввода и корректировки должны быть максимально «похожими» на привычные для пользователя документы; результаты работы не только отображаются на экране, но и выводятся в текстовый файл в привычной для пользователя форме с возможностью корректировки и распечатки.

9. Средства документации. Программы снабжаются внутренней документацией в виде комментариев, средств меню и диалога, средств помощи, а также внешней документацией в виде отчета, содержащего постановку задачи, структуру баз данных, инструкцию для пользователя с элементами рекламы, описанием алгоритма, экранных форм, примеры входных и выходных документов.

Темы курсовых работ

1. Книжный магазин.
2. Магазин DVD-дисков с предоставлением проката дисков.
3. Мелкооптовый магазин продовольственных товаров.
4. Магазин компьютерных комплектующих.
5. Компьютерный магазин.
6. Магазин аудиотехники.
7. Расчет с пользователями ЛВС за использование Интернет-ресурсов.
8. Ломбард.
9. Банковские договоры на депозит
10. Банковские договоры на кредитование.
11. Оптовый склад.
12. Турагенство. Планирование туров.
13. Экскурсионное обслуживание. Продажа и заказ билетов.
14. Прием коммунальных платежей.
15. Торговый дом. Заказ и отпуск товаров.

16. Косметический салон. Предварительная запись и расчет с клиентами.
17. Магазин автозапчастей. Предварительный заказ и расчет с клиентами.
18. Грузоперевозки. Предварительный заказ и планирование.
19. Отдел материально-технического снабжения предприятия.
20. Отдел сбыта готовой продукции предприятия.
21. Финансовый отдел предприятия.
22. Учет отработанного времени сотрудников.
23. Расчет заработной платы сотрудников бюджетной сферы.
24. Регистрация объектов, связанных с преступлениями.
25. Учет малолетних правонарушителей.
26. Регистрация дорожно-транспортных происшествий.
27. Регистрация разыскиваемых транспортных средств.
28. Расчет с клиентами юридической консультации.
29. Научные труды и методические разработки МО
30. Выпускники школы
31. Биржа труда
32. Участники конференции
33. Предварительная запись клиентов юридической консультации
34. Регистрация договоров купли-продажи недвижимости в нотариальной конторе.
35. Регистрация заявлений и жалоб граждан.
36. Регистрация нарушений авторских прав.

Критерии и шкала оценивания по оценочному средству курсовая работа

Шкала оценивания (интервал баллов)[2]	Критерий оценивания
5	Курсовая работа выполнена на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Курсовая работа выполнена на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Курсовая работа выполнена на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Курсовая работа выполнена на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (зачет)

Вопросы к зачету

1. Операторы ввода и вывода данных. Оператор присваивания.
2. Встроенные функции и методы для работы с числами.
3. Модуль Math. Математические функции.

4. Операторы для работы с последовательностью.
5. Приоритет выполнения операторов. Генерация случайных чисел. Модуль random.
6. Операции сравнения. Операторы условного перехода.
7. Операторы перехода на следующую итерацию и прерывания цикла.
8. Функции range() и enumerate().
9. Вложенные циклы.
10. Строки и двоичные данные. Создание строки.
11. Специальные символы. Операции над строками.
12. Форматирование строк. Метод format().
13. Функции и методы для работы со строками и символами.
14. Регулярные выражения: синтаксис, поиск по шаблону.
15. Списки. Создание списка. Операции над списками.
16. Многомерные списки. Перебор элементов списка.
17. Генераторы списков и выражения-генераторы.
18. Функции для работы со списками. Добавление и удаление элементов списка.
19. Добавление и удаление элементов списка.
20. Поиск элемента в списке и получение сведений о значениях, входящих в список. переворачивание и перемешивание списка.
21. Выбор элемента списка случайным образом. Сортировка списка.
22. Заполнение списка числами. Преобразование списка в строку.
23. Кортежи. Операции над кортежами.
24. Множества. Операции над множествами.
25. Диапазоны. Операции над диапазонами.
26. Словари. Создание словаря. Операции над словарями.
27. Методы для работы над словарями. Генераторы словарей.
28. Определение функции и ее вызов. Необязательные параметры функций и сопос-тавление по ключам.
29. Переменное число параметров в функции.
30. Анонимные функции.
31. Функции-генераторы.
32. Декораторы функций.
33. Глобальные и локальные переменные.
34. Рекурсивные функции. Вложенные функции.
35. Модули. Понятие модуля. Подключение модуля: инструкции import и from.
36. Пути поиска модулей. Повторная загрузка модулей.
37. Пакеты. Понятие пакета. Работа с пакетами.
38. Понятие класса, атрибута и метода.

39. Определение класса и создание экземпляра класса. Конструкторы и деструкторы.
40. Наследование. Множественное наследование.
41. Понятие примесей и их использование.
42. Специальные методы классов. Перегрузка операторов.
43. Статические методы и методы классов.
44. Абстрактные методы.
45. Ограничение доступа к идентификаторам внутри класса.
46. Свойства классов. Декораторы классов.
47. Итератор класса. Понятие итератора класса и его использование.
48. Понятие контейнера. Контейнеры-последовательности. Контейнеры-словари.
49. Перечисления. Атрибуты и методы перечислений.
50. Открытие файла. Методы для работы с файлами.
51. Классы и модули используемые для доступа к файлам.
52. Права доступа к файлам и каталогам.
53. Функции для манипулирования файлами.
54. Python. Преобразование пути к файлам и каталогам.
55. Перенаправление ввода/вывода.
56. Сохранение объектов в файл.
57. Функции для работы с каталогами.
58. Принципы построения графических пользовательских интерфейсов.
59. Основы технологии MVC.
60. Основные парадигмы программирования
61. Принципы разработки информационных систем и программных комплексов
62. Принципы процедурно-структурного программирования.
63. Основные принципы объектно-ориентированного программирования.

Критерии и шкала оценивания промежуточной аттестации «зачет»

Характеристика знания предмета и ответов	Зачеты
Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.	зачтено
Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.	
Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.	
Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.	не зачтено

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)