Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Максимов Алексей Борисович
Должность: директор департамента по образовательной политике
Дата подписания: 30.10.2023 12:56:44
Уникальный программный ключ:
8db180d1a3f02ac9e60521a672742735e18b1d0

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Декан факультета

«Информационные технологии»

_____/Д.Г.Демидов/

«16» мая 2022

Рабочая программа дисциплины

«Основы программирования»

Направление подготовки:

09.03.03 Прикладная информатика

Образовательная программа (профиль):

«Информационные технологии управления бизнесом»

Год начала обучения:

2022

Уровень образования:

Бакалавриат

Квалификация (степень) выпускника:

Бакалавр

Форма обучения:

очная

Москва, 2022

Рабочая программа дисциплины «Основ программирования» составлена в соответствии с федеральным образовательным стандартом высшего образования – бакалавриата по направлению подготовки 09.03.-3 Прикладная информатика.

Заведующий кафедрой «Инфокогнитивные технологии»:



к.т.н., доцент /Е.А.Пухова /

Согласовано:

Руководитель образовательной программы:



_____ /М.В.Даньшина/

Программу составили:

старший преподаватель_____ / И.Н.Никишина/

_____ / _____ /

_____ / _____ /

_____ / _____ /

_____ / _____ /

1. Цели освоения дисциплины

К **основным целям** освоения дисциплины относится:

- получение знаний об основных конструкциях и идиомах языка программирования Python;
- умение на практике написать программу для выполнения поставленной аналитической задачи;
- закрепление получаемых в семестре знаний и навыков на практике;
- формирование взаимосвязей, получаемых в семестре знаний и навыков с изученными ранее и изучаемых параллельно с данной дисциплиной;
- подготовка студентов к деятельности в соответствии с квалификационной характеристикой бакалавра.

К **основным задачам** дисциплины относятся:

- овладение навыками и приемами программирования в рамках веб-технологии;
- изучение и освоение теоретического материала, как в процессе контактной, так и в ходе самостоятельной работы;
- выполнение предоставленных практических заданий различных форм, как в процессе контактной, так и в ходе самостоятельной работы;
- самостоятельная работа над тематикой дисциплины для формирования компетенций основной образовательной программы (далее, ООП).

2. Место дисциплины в структуре ОПОП

Дисциплина относится к числу учебных дисциплин части, формируемой участниками образовательных отношений.

Дисциплина взаимосвязана логически и содержательно-методически со следующими дисциплинами и практиками ОПОП:

- Алгоритмы и структуры данных;
- Прикладное программирование;
- Проектирование баз данных.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

В результате освоения дисциплины у обучающихся формируются следующие компетенции и должны быть достигнуты следующие результаты обучения как этап формирования соответствующих компетенций.

Код компетенции	Наименование компетенции	Индикаторы планируемых результатов обучения по дисциплине
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знать: возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации. ИПК-4.2. Уметь: разрабатывать алгоритмы для конкретных задач, учитывая различия в представлении информационных объектов современными браузерами; проводить работы по интеграции программных модулей с использованием языков программирования; тестировать написанную программу на наличие ошибок, интегрировать различные модули в одну информационную систему. ИПК-4.3. Владеть:

		навыками использования языка разметки гипертекста с языками программирования; навыками тестирования разработанной программы.

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплин (модулей), практик в соответствии с учебным планом и календарным графиком учебного процесса.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 4 зачетных единицы, т.е. 144 академических часов (из них 90 часов – самостоятельная работа студентов).

Разделы дисциплины изучаются на первом курсе в первом семестре, форма промежуточной аттестации –экзамен.

Содержание и темы лабораторных работ

ЛР-1	**Линейные программы**	2 ак. часа
Цель выполнения лабораторной работы: Дать студентам практический навык в подготовке простой программы и в записи математических выражений на языке программирования Python.		
Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.		
Порядок выполнения лабораторной работы: • Подготовка к выполнению к работе, в том числе: • изучение основ языка Python и механизмов его работы. • Составление и написание линейных алгоритмов. • Написание консольных приложений, использующих ввод/вывод. • Защита лабораторной работы.		
Контрольные вопросы: 1. Типы данных. Переменные. [ПК-1, ПК-2] 2. Числовые типы данных. [ПК-1, ПК-2] 3. Операции над числовыми типами данных. [ПК-1, ПК-2] 4. Строки. Строки unicod. [ПК-1, ПК-2] 5. Вывод данных. Ввод данных. [ПК-1, ПК-2]		
ЛР-2	**Разветвляющиеся вычислительные процессы**	2 ак. часа
Цель выполнения лабораторной работы: дать студентам практический навык в использовании условных операторов ветвления на языке программирования Python		
Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.		

Порядок выполнения лабораторной работы:
• Подготовка к выполнению к работе, в том числе:
  • изучение условных операторов Python;
  • изучение логических операторов Python;		

- Получение варианта задания.
- Разработка программы, согласно требованиям.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

6. Вывод данных. Ввод данных. [ПК-1, ПК-2]
7. Форматированный ввод/вывод. [ПК-1, ПК-2]
8. Списки. Выражения в списках. Использование списков, как стеков. Использование списков, как очередей. Операции сравнения для списков. [ПК-1, ПК-2]
9. Диапазоны. Кортежи. Отличие кортежей от словарей. Словари. [ПК-1, ПК-2]
10. Оператор if. Особенности операторов сравнения. [ПК-1, ПК-2]

ЛР-3	**Организация циклов**	2 ак. часа

Цель выполнения лабораторной работы: Дать студентам практический навык в использовании базовых конструкций структурного программирования - операторов цикла.

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
  - изучение операторов цикла в Python.
- Получение варианта задания.
- Разработка программы, согласно требованиям.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.

Контрольные вопросы:

11. Операторы цикла. Оператор for. Оператор while. Завершение цикла. Продолжение цикла. Оператор pass.
12. Определение функции. Пространство имен функции. Передача параметров. Ключи.

ЛР-4	**Одномерные массивы.**	2 ак. часа

Цель выполнения лабораторной работы: Дать студентам практический навык в написании программ обработки одномерных массивов: поиск максимумов и минимумов, сортировки.

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
  - изучение структур данных Python;
  - изучение встроенных функций работы с массивами;
  - изучение алгоритмов работы с массивами;
  - изучение различных алгоритмов сортировки массивов.
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Сборка единой программы, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

13. Списки. Выражения в списках. Использование списков, как стеков. Использование списков, как очередей. Операции сравнения для списков. . [ПК-1, ПК-2]
14. Диапазоны. Кортежи. Отличие кортежей от словарей. Словари. . [ПК-1, ПК-2]
15. Операторы цикла. Оператор for. Оператор while. Завершение цикла. Продолжение цикла. Оператор pass. [ПК-1, ПК-2]

ЛР-5	**Динамическое формирование контента и меню. Таблица умножения.**	2 ак. часа

Цель выполнения лабораторной работы: Дать студентам практический навык в написании программ обработки двумерных массивов с использованием функций..

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
  - изучение способов динамического формирования контента и пользовательских элементов управления.
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Сборка единой программы, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

16. Списки. Выражения в списках. Использование списков, как стеков. Использование списков, как очередей. Операции сравнения для списков. . [ПК-1, ПК-2]
17. Диапазоны. Кортежи. Отличие кортежей от словарей. Словари. . [ПК-1, ПК-2]
18. Операторы цикла. Оператор for. Оператор while. Завершение цикла. Продолжение цикла. Оператор pass.данные? [ПК-1, ПК-2]

ЛР-6	**Файлы**	2 ак. часа

Цель выполнения лабораторной работы: Дать студентам практический навык в написании программ, в которых выполняются операции с текстовыми файлами - чтение, запись.

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
  - повторение ранее пройденного материала;
  - изучение методов работы с файловой системой в языке Python.
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Сборка единой программы, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

19. Класс File. Открытие файла. Методы класса для File ввода-вывода. . [ПК-1, ПК-2]
20. Взаимодействие с файловой системой. Модуль path. Объекты и файловый ввод-вывод. [ПК-1, ПК-2]

ЛР-7	**GUI, классы, модуль tkinter.**	2 ак. часа

Цель выполнения лабораторной работы: Дать студентам практический навык в написании программ, использующих графический интерфейс пользователя (GUI). Ознакомить с такими понятиями, как класс, виджеты, массивы записей, а так же с атрибутами класса и методами передачи данных между классами..

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
    - Изучение методов построения графического пользовательского интерфейса;
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Сборка единой программы, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

21. Создание модулей. Поиск модулей. Компиляция модулей на Python. Стандартные модули Python. [ПК-1, ПК-2]
22. Использование функции dir(). [ПК-1, ПК-2]
23. Объявление класса. Управление атрибутами и методами класса. [ПК-1, ПК-2]
24. Объявление объектов. Множественное наследование. Заимствование свойств и методов у родительского класса. [ПК-1, ПК-2]

ЛР-8	**Программирование в графическом режиме.**	4 ак. часа

Цель выполнения лабораторной работы: Познакомить студентом на практике с написанием программ для формирования графических изображений с использованием модулей Python.

Результат: Алгоритм решения задачи, листинг решения задачи на языке Python, отчет по лабораторной работе

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
    - Изучение методов построения графического пользовательского интерфейса;
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Сборка единой программы, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Защита лабораторной работы.

Контрольные вопросы:

25. Создание модулей. Поиск модулей. Компиляция модулей на Python. Стандартные модули Python. [ПК-1, ПК-2]
26. Использование функции dir(). [ПК-1, ПК-2]
27. Объявление класса. Управление атрибутами и методами класса. [ПК-1, ПК-2]
28. Объявление объектов. Множественное наследование. Заимствование свойств и методов у родительского класса. [ПК-1, ПК-2]

ЛР-9	**Подготовка технического задания для разработки программного продукта.**	12 ак. часов

Цель выполнения лабораторной работы: Познакомить студентов с приемами создания технической документации

Результат: Техническое задание на разработку программного продукта.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
    - изучение методов документирования программного обеспечения;
    - изучение различных методологий разработки.
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка технического задания по полученному варианту.
- Защита лабораторной работы.

Контрольные вопросы:

29. назначение объекта; [ПК-1, ПК-2]
30. технические характеристики; [ПК-1, ПК-2]
31. технико-экономические требования; [ПК-1, ПК-2]
32. показатели качества; [ПК-1, ПК-2]
33. предписание по выполнению стадий создания технологической, конструкторской, программной и иной документации; [ПК-1, ПК-2]
34. специальные требования. [ПК-1, ПК-2]

ЛР-10	**Выполнение проекта.**	24 ак. часов

Цель выполнения лабораторной работы: Получить практические навыки самостоятельной реализации программного продукта

Результат: Программный продукт, реализованный в соответствии с техническим заданием.

Порядок выполнения лабораторной работы:

- Подготовка к выполнению к работе, в том числе:
  - изучение принципов командной разработки программного обеспечения;
  - изучение систем контроля версий.
- Получение варианта задания.
- Декомпозиция задания на отдельные подзадачи.
- Разработка соответствующих подзадачам пользовательских функций.
- Создание проекта, реализующей поставленную задачу.
- Тестирование программы, в том числе на устойчивость работы при некорректных данных.
- Подготовка технической документации по проекту.
- Подготовка презентационных материалов по проекту.
- Защита проекта.

Контрольные вопросы:

35. Критерии качества программного средства. Определение качества ПО в стандарте ISO 9126. Многоуровневая модель качества ПО. Оценочные характеристики качества программного продукта. [ПК-1, ПК-2]
36. Жизненный цикл программного продукта, фазы жизненного цикла. Этапы классического жизненного цикла, их содержание. [ПК-1, ПК-2]
37. Фаза разработки, этапы процесса разработки. Стратегии конструирования ПО: линейная, инкрементная, эволюционная [ПК-1, ПК-2]
38. Стандарт ISO/IEC 1220795: основные определения – система, модель жизненного цикла, квалификационные требования. Основные процессы, их содержание, работы и задачи процесса разработки. [ПК-1, ПК-2]
39. Стандарт ISO/IEC 15504 (SPICE): оценка возможностей разработчика. Связь этого стандарта с моделью зрелости предприятия SEI CMM. [ПК-1, ПК-2]
40. Прогностические модели процесса разработки: каскадная, RAD, спиральная. [ПК-1, ПК-2]
41. Адаптивные модели процесса разработки: экстремальное программирование, Scrum. [ПК-1, ПК-2]
42. Руководство программным проектом. Предварительные оценки проекта. Системный анализ и анализ требований. Анализ рисков. Планирование процесса разработки. Типовая структура распределения работ. [ПК-1, ПК-2]
43. Контроль процесса разработки. Размерно и функциональноориентированные метрики. Метрические характеристики объектно-ориентированных систем. [ПК-1, ПК-2]
44. Структурный и объектно-ориентированный подходы к разработке ПО. Их сравнительный анализ. Сущность объектного подхода к разработке программных средств. [ПК-1, ПК-2]

45. Анализ предметной области: цели и задачи. Модели предметной области. Формальные определения. Классификация моделей. Методология IDEF0, синтаксис IDEF0моделей. [ПК-1, ПК-2]
46. Диаграммы потоков данных (DFD диаграммы) и диаграммы потоков работ (IDEF3диаграммы), их использование при моделировании предметной области. [ПК-1, ПК-2]
47. Объектно-ориентированный анализ предметной области. Методика определения границ системы и ключевых абстракций. Пример проведения анализа. Функциональные и нефункциональные требования к системе. [ПК-1, ПК-2]
48. Функциональные требования к системе. Способ их представления в виде UML диаграммы. Пример диаграммы с использованием отношений «расширяет» и «включает». Понятие прецедента и сценария. [ПК-1, ПК-2]
49. Концептуальная модель системы: концептуальные классы, системные события и системные операции. Способ их представления в виде UML диаграмм. Пример концептуального описания прецедента. [ПК-1, ПК-2]
50. Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия. [ПК-1, ПК-2]
51. Проектирование программных средств. Цели и задачи этапа проектирования. Понятие модели проектирования, ее отличия от концептуальной модели. Стадии проектирования, их краткая характеристика [ПК-1, ПК-2]
52. Задачи, решаемые на стадии эскизного проектирования. Понятие архитектуры ПС. Проблема выбора архитектуры. Влияние архитектуры на качественные характеристики ПС. [ПК-1, ПК-2]
53. Понятие модуля и модульного программирования. Преимущества модульного подхода к разработке ПО. Модули как средство физического структурирования ПО. Свойства модулей. [ПК-1, ПК-2]
54. Задачи, решаемые на стадии детального проектирования. Цели и задачи проектирования пользовательского интерфейса. [ПК-1, ПК-2]
55. Понятие шаблона. Классификация шаблонов. Стандарт описания шаблонов. [ПК-1, ПК-2]
56. Идентификация методов программных классов. Диаграммы классов, способы отображения отношений ассоциации и зависимости. Пример диаграммы классов. [ПК-1, ПК-2]
57. Тестирование и отладка программного средства. Стадии тестирования и их характеристика. Основные принципы тестирования. Тесты и тестовые наборы. Понятие тестового покрытия. [ПК-1, ПК-2]
58. Отладочное тестирование. Соотношение структурного и функционального подходов. Примеры реализации. [ПК-1, ПК-2]
59. Интеграционное тестирование. Виды интеграционного тестирования. Критерии полноты тестовых наборов. Регрессионное тестирование. Критерии завершения отладочного тестирования. [ПК-1, ПК-2]
60. Системное тестирование. Виды системного тестирования. Критерии полноты тестовых наборов. [ПК-1, ПК-2]
61. Особенности объектно-ориентированного тестирования. Расширение области применения тестирования. Критерии тестирования моделей. Тестирование классов. Тестирование кластеров и потоковое тестирование. [ПК-1, ПК-2]
62. Понятие автоматизированного тестирования. Автотесты. Достоинства и недостатки автоматизированного тестирования. Средства автоматизированного тестирования.
63. Система Git, ее архитектура. Репозиторий, его структура, правки. Команды Git для работы с репозиториями. Основы рабочего процесса с Git. Конфликты и способы их разрешения. [ПК-1, ПК-2]

64. Понятие сборки, манифест сборки. Сборка приложения, системы автоматизации сборки. [ПК-1, ПК-2]
65. Документирование процесса разработки. Типы документов управления. [ПК-1, ПК-2]
66. Документирование программного продукта. Документация сопровождения, ее назначение и состав. Пользовательская документация, ее назначение и состав. [ПК-1, ПК-2]

5. Образовательные технологии

Методика преподавания дисциплины и реализация компетентностного подхода в изложении и восприятии материала предусматривает использование следующих активных и интерактивных форм проведения групповых, индивидуальных, аудиторных занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков у обучающихся:

- выполнение лабораторных работ в лабораториях вуза;
- индивидуальные и групповые консультации студентов преподавателем, в том числе в виде защиты выполненных заданий в рамках самостоятельной работы;
- посещение профильных конференций и работа на мастер-классах экспертов и специалистов индустрии.

Самостоятельная внеаудиторная работа студентов состоит из:

- подготовки к выполнению и подготовки к защите лабораторных работ;
- чтения литературы и освоения дополнительного материала в рамках тематики дисциплины;
- подготовки к текущей аттестации;
- подготовки к промежуточной аттестации.

6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения используются следующие оценочные формы самостоятельной работы студентов, оценочные средства текущего контроля успеваемости и промежуточных аттестаций:

- выполнение лабораторных работ, экзамен.

6.1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Основы программирования»

Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

В результате освоения дисциплины формируются следующие компетенции:

Код компетенции	В результате освоения образовательной программы обучающийся должен обладать
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов

В процессе освоения образовательной программы данные компетенции, в том числе их отдельные компоненты, формируются поэтапно в ходе освоения обучающимися дисциплины в соответствии с учебным планом и календарным графиком учебного процесса.

6.1.1. Описание показателей и критериев оценивания компетенций, формируемых по итогам освоения дисциплины, описание шкал оценивания

Показателем оценивания компетенций на различных этапах их формирования является достижение обучающимися планируемых результатов обучения по дисциплине.

Показатель	Критерии оценивания

	2	3	4	5
ПК-3. Способен разрабатывать требования и проектировать программное обеспечение				
ИПК-3.1. Знать: возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО.	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточность знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях.	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.
ПК-4**.** Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов				
ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации. ИПК-4.2. Уметь: разрабатывать алгоритмы для конкретных задач, учитывая различия в представлении информационных	Обучающийся демонстрирует полное отсутствие или недостаточное соответствие материалу дисциплины знаний, указанных в индикаторах компетенций дисциплины.	Обучающийся демонстрирует неполное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Допускаются значительные ошибки, проявляется недостаточност	Обучающийся демонстрирует частичное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Но допускаются незначительные ошибки, неточности,	Обучающийся демонстрирует полное соответствие следующих знаний, указанных в индикаторах компетенций дисциплины. Свободно оперирует приобретенными знаниями.

объектов современными браузерами; проводить работы по интеграции программных модулей с использованием языков программирования; тестировать написанную программу на наличие ошибок, интегрировать различные модули в одну информационную систему. ИПК-4.3. Владеть: навыками использования языка разметки гипертекста с языками программирования; навыками тестирования разработанной программы.		ь знаний, по ряду показателей, обучающийся испытывает значительные затруднения при оперировании знаниями при их переносе на новые ситуации.	затруднения при аналитических операциях.	

Шкала оценивания результатов промежуточной аттестации и её описание:

Форма промежуточной аттестации: экзамен.

Промежуточная аттестация обучающихся в форме экзамена проводится по результатам выполнения всех видов учебной работы, предусмотренных учебным планом по данной дисциплине (модулю), при этом учитываются результаты текущего контроля успеваемости в течение семестра. Оценка степени достижения обучающимися планируемых результатов обучения по дисциплине (модулю) проводится преподавателем, ведущим занятия по дисциплине (модулю) методом экспертной оценки. По итогам промежуточной аттестации по

дисциплине (модулю) выставляется оценка «отлично», «хорошо», «удовлетворительно» или «неудовлетворительно».

К промежуточной аттестации допускаются только студенты, выполнившие все виды учебной работы, предусмотренные рабочей программой по дисциплине.

Шкала оценивания	Описание
Отлично	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 5. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, умений, навыков приведенным в таблицах показателей, оперирует приобретенными знаниями, умениями, навыками, применяет их в ситуациях повышенной сложности. При этом могут быть допущены незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
Хорошо	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 4. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует неполное, правильное соответствие знаний, умений, навыков приведенным в таблицах показателей, либо если при этом были допущены 2-3 несущественные ошибки.
Удовлетворительно	Среднее значение для всех формируемых на момент проведения аттестации уровней компетенций – 3. Выполнены все виды учебной работы, предусмотренные учебным планом. Студент демонстрирует соответствие знаний, в котором освещена основная, наиболее важная часть материала, но при этом допущена одна значительная ошибка или неточность.
Неудовлетворительно	Не достигнуто пороговое значение хотя бы для одного уровня формируемых на момент проведения аттестации компетенций. Не выполнен один или более видов учебной работы, предусмотренных учебным планом. Студент демонстрирует неполное соответствие знаний, умений, навыков приведенным в таблицах показателей, допускаются значительные ошибки, проявляется отсутствие знаний, умений, навыков по ряду показателей, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

1.Введение в программирование на PHP5 [Электронный ресурс] Флойд К. С. Интернет-Университет Информационных Технологий 2007 г. http://www.knigafund.ru/books/176606

2.Основы программирования на PHP [Электронный ресурс] Савельева Н. В. Интернет-Университет Информационных Технологий 2005 г. http://www.knigafund.ru/books/178025

7.2. Дополнительная литература

1. Перспективные языки веб-разработки [Электронный ресурс] Богданов М. Р. Национальный Открытый Университет «ИНТУИТ» 2016 г. http://www.knigafund.ru/books/177945

2.Ваш первый сайт с использованием PHP-скриптов [Электронный ресурс]: учебное пособие Строганов А. С. Диалог-МИФИ 2015 г. http://www.knigafund.ru/books/198482

8. Материально-техническое обеспечение дисциплины

8.1 Требования к оборудованию и помещению для занятий

Лабораторные работы и самостоятельная работа студентов должны проводиться в специализированной аудитории, оснащенной современной оргтехникой и персональными компьютерами с программным обеспечением в соответствии с тематикой изучаемого материала. Число рабочих мест в аудитории должно быть достаточным для обеспечения индивидуальной работы студентов. Рабочее место преподавателя должно быть оснащены современным компьютером с подключенным к нему проектором на настенный экран, или иным аналогичным по функциональному назначению оборудованием.

8.2 Требования к программному обеспечению

Для выполнения лабораторных работ и самостоятельной работы необходимо следующее программное обеспечение:

1. Microsoft Windows.
2. Notepad++.
3. Anaconda Community.

4. Python Software Foundation Python.
5. Веб-браузер, Chrome.
6. Visual Studio 2019.
7. Visual Studio Code.
8. JetBrains PyCharm Community.
9. Git.
10. СУБД MySQL.
11. СУБД PostgresSQL.
12. СУБД SQLite
13. Microsoft Office 365.

Для проведения лекционных и практических занятий специального программного обеспечения для освоения дисциплины не требуется.

9. Методические рекомендации для самостоятельной работы студентов

Изучение дисциплины осуществляется в строгом соответствии с целевой установкой в тесной взаимосвязи учебным планом. Основой теоретической подготовки студентов являются аудиторные занятия, лабораторные работы.

В процессе самостоятельной работы студенты закрепляют и углубляют знания, полученные во время аудиторных занятий, дорабатывают конспекты и записи, готовятся к проведению и обрабатывают результаты лабораторных работ, готовятся к промежуточной аттестации, а также самостоятельно изучают отдельные темы учебной программы.

На занятиях студентов, в том числе предполагающих практическую деятельность, осуществляется закрепление полученных, в том числе и в процессе самостоятельной работы, знаний. Особое внимание обращается на развитие умений и навыков установления связи положений теории с профессиональной деятельностью будущего специалиста.

Самостоятельная работа осуществляется индивидуально. Контроль самостоятельной работы организуется в двух формах:

- самоконтроль и самооценка студента;

- контроль со стороны преподавателей (текущий и промежуточный).

Критериями оценки результатов самостоятельной работы студента являются:

- уровень освоения студентом учебного материала;
- умения студента использовать теоретические знания при выполнении практических задач;
- сформированность компетенций;
- оформление материала в соответствии с требованиями.

10. Методические рекомендации для преподавателя

1. При подготовке к занятиям следует предварительно проработать материал занятия, предусмотрев его подачу точно в отведенное для этого время занятия. Следует подготовить необходимые материалы – теоретические сведения, задачи и др. При проведении занятия следует контролировать подачу материала и решение заданий с учетом учебного времени, отведенного для занятия.

2. При проверке работ и отчетов следует учитывать не только правильность выполнения заданий, но и оптимальность выбранных методов решения, правильность выполнения всех его шагов.

Приложение 1

Структура и содержание дисциплины «Основы программирования»
по направлению подготовки 09.03.03 «Прикладная информатика»
(бакалавр)

№	Раздел	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов, и трудоемкость в часах					Виды самостоятельной работы студентов					Формы аттестации	
				Л	П/С	Лаб	СРС	КСР	К.Р.	К.П.	РГР	Реферат	К/р	Э	З
1	Лабораторная работа ЛР-1. *Линейные программы*	1	1			2	4								
2	Лабораторная работа ЛР-2. *Разветвляющиеся вычислительные процессы*	1	2			2	4								
3	Лабораторная работа ЛР-3. *Организация циклов.*	1	3			2	4								
4	Лабораторная работа ЛР-4. *Одномерные массивы*	1	4			2	4								
5	Лабораторная работа ЛР-5. *Двумерные массивы и функции.*	1	5			2	4								
6	Лабораторная работа ЛР-6. *Файлы.*	1	6-7			2	4								
7	Лабораторная работа ЛР-7. *GUI, классы, модуль Tkinter.*	1	8-9			2	8								
8	Лабораторная работа ЛР-8. *Программирование в графическом режиме*	1	10-11			8	8								
9	Лабораторная работа ЛР-9. *Подготовка технического задания для разработки программного продукта.*	1	12-13			16	18								

10	Лабораторная работа ЛР-10. *Выполнение проекта*		14-15			16	34								
	Форма аттестации		**16-17**							КП				**Э**	
	Всего часов по дисциплине					**54**	**90**								

Приложение 2

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

Направление подготовки 09.03.03 Прикладная информатика
профиль подготовки «Информационные технологии управления бизнесом»
Форма обучения: очная

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Основы программирования

Состав:

1. **Показатель уровня сформированности компетенций.**
2. **Перечень оценочных средств.**
3. **Экзаменационное задание.**
4. **Типовой экзаменационный билет.**
5. **Типовые практические задания.**

Москва, 2022 год

1. ПОКАЗАТЕЛЬ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

«Основы программирования»					
ФГОС ВО 09.03.03 «Прикладная информатика» профиль подготовки «**Информационные технологии управления бизнесом»**					
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие **общепрофессиональные компетенции**:					
Компетенции		**Перечень компонентов**	**Технология формирования компетенций**	**Форма оценочного средства****	**Степени уровней освоения компетенций**
Индекс	**Индекс**				
ПК-3	Способен разрабатывать требования и проектировать программное обеспечение	ИПК-3.1. Знать: возможности современных и перспективных средств разработки программных продуктов, технических средств; методологии разработки программного обеспечения и технологии программирования. ИПК-3.3. Владеть: современным инструментарием и средами разработки программного кода; современным инструментарием и средами проектирования программного кода, методами тестирования ПО	Лабораторные работы, самостоятельная работа	УО П КП Экзамен	**БАЗОВЫЙ УРОВЕНЬ:** способность выполнять полученное задание, применяя полученные знание и умения на практике, владеть соответствующими индикаторами компетенции при выполнении задания. **ПРОДВИНУТЫЙ УРОВЕНЬ:** способность выполнять полученное задание и решать самостоятельно сформированные задачи, применяя полученные знание и умения на практике. Уверенно владеть соответствующими индикаторами компетенции при выполнении задания, комбинировать их между собой и с индикаторами других компетенций для достижения проектных результатов.
ПК-4	Способен проводить работы по интеграции программных модулей и компонент и проверку работоспособности выпусков программных продуктов	ИПК-4.1. Знать: методы описания алгоритмов, основные абстрактные типы данных и их реализации. ИПК-4.2. Уметь: разрабатывать алгоритмы для конкретных задач,			

		учитывая различия в представлении информационных объектов современными браузерами; проводить работы по интеграции программных модулей с использованием языков программирования; тестировать написанную программу на наличие ошибок, интегрировать различные модули в одну информационную систему. ИПК-4.3. Владеть: навыками использования языка разметки гипертекста с языками программирования; навыками тестирования разработанной программы.			

**- Сокращения форм оценочных средств см. в приложении 2 к РП.

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ

№ ОС	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в ФОС
1	Устный опрос / собеседование, (УО)	Средство контроля, организованное как презентация обучающимся результатов выполнения Курсового проекта с демонстрацией наглядных материалов и ответов на вопросы педагогических работников (работника) на тему доклада, теме, проблеме и т.п.	Контрольные вопросы
2	Проект (П)	Конечный продукт, получаемый в результате планирования и выполнения комплекса учебных и исследовательских заданий. Позволяет оценить умения обучающихся самостоятельно конструировать свои знания в процессе решения практических задач и проблем, ориентироваться в информационном пространстве и уровень сформированности аналитических, исследовательских навыков, навыков практического и творческого мышления. Может выполняться в индивидуальном порядке или группой обучающихся.	Типовое практическое задание

3. ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ

Экзаменационное задание выполняется студентом индивидуально, по итогам изучения дисциплины или ее части. При этом достижение порогового результата работы над экзаменационным заданием соответствует описанному в п. 3 данного документа этапу освоения соответствующих компетенций на базовом или продвинутом уровне.

Базовый уровень: способность выполнять полученное задание, применяя полученные знание и умения на практике, владеть соответствующими индикаторами компетенции при выполнении задания.

Продвинутый уровень: способность выполнять полученное задание и решать самостоятельно сформированные задачи, применяя полученные знание и умения на практике. Уверенно владеть соответствующими индикаторами компетенции при выполнении задания, комбинировать их между собой и с индикаторами других компетенций для достижения проектных результатов.

Форма экзаменационного задания выбирается преподавателем и утверждается на заседании кафедры. Экзамен может проходить в следующих формах и с использованием следующих оценочных средств.

Форма	Представление оценочного средства в ФОС
Устная.	Банк контрольных вопросов, соответствующих отдельным темам дисциплины (см. п. 4 настоящего документа). Вопросы формируют экзаменационный билет (см. ниже), состоящий из теоретических вопросов и практических заданий (типовые практические задания представлены ниже). Билеты, включая вопросы и практические задания, формируются преподавателем и утверждаются на заседании кафедры. В них могут быть включены дополнительные контрольные вопросы и задания, не требующие у студентов наличия не формируемых данной дисциплиной компетенций или более высоких этапов сформированности формируемых. Для ответа на каждый вопрос и для решения любого практического задания студент должен находится на требуемом для данной дисциплине уровне сформированности всех соответствующих ей компетенций: каждый вопрос и задание проверяет уровень сформированности всех соответствующих данной дисциплине компетенций.
Письменная.	Оценочное средство полностью соответствует оценочным средствам устной формы задания.
Практико-ориентированная (формат WorldSkills).	Типовое задание практико-ориентированного экзамена. Задание практико-ориентированного формируется преподавателем на основе типового и Методических рекомендаций по разработке задания ПОЭ, утверждаются на заседании кафедры. Задание ПОЭ проверяет уровень сформированности всех соответствующих дисциплине компетенций.

4. ТИПОВОЙ ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»
направление подготовки 09.03.03 Прикладная информатика

ВОПРОСЫ:

1. Язык программирования Python. Взаимодействие с файловой системой. Модуль path. Объекты и файловый ввод-вывод.

2. Язык программирования Python. Списки. Выражения в списках. Использование списков, как стеков. Использование списков, как очередей. Операции сравнения для списков.

3. Понятие автоматизированного тестирования. Автотесты. Достоинства и недостатки автоматизированного тестирования. Средства автоматизированного тестирования.

4. Практическое задание Напишите программу, которая поможет слушателям начальной школы изучить сложение, умножение, вычитание. Используйте генератор псевдослучайных чисел для выработки двух положительных целых чисел в диапазоне 1...5. Используйте генератор псевдослучайных чисел для выбора одной из трех упомянутых операций. Программа должна выдавать вопрос типа «Сколько будет 2*3?» Затем учащийся вводит ответ. Ваша программа проверяет этот ответ. Если он правильный, пользователю нужно выдать сообщение «Очень хорошо!» и затем задать следующий вопрос. Если ответ неправильный, нужно выдать сообщение «Нет. Повторите, пожалуйста, снова.» и затем задать тот же самый вопрос повторно до получения правильного ответа. Предусмотреть систему начисления баллов и штрафов за правильные и не правильные ответы. Все результаты необходимо сохранить в файл. Предусмотреть графический и консольный вариант использования.

Утверждено: _____ / _____ / «__» _____ 20__г.

5. ТИПОВЫЕ ПРАКТИЧЕСКИЕ ЗАДАНИЯ

1. Напишите интерактивный учебник биологии. Он должен обеспечивать выдачу в случайном порядке пяти вопросов типа «Что такое курица (крокодил, береза, муравей, щука)» и принимать ответ типа «Курица это 1) рыба, 2) насекомое, 3) птица, 4) земноводное, 5) растение». После опроса поставьте испытуемому оценку.

2. Напишите интерпретатор. Предположим, что в файле data.txt находится последовательность слов и чисел типа 13 плюс 4 минус 2 умножить3 разделить 4 возвести 2 и т.д. Интерпретатор должен прочитать и распознать словесные команды и выполнить необходимые операции.

3. Паук находится на плоскости в точке с координатами x=50 и y=50. Каждую секунду он делает шаг влево, вправо, вниз или вверх с равной вероятностью. Смоделируйте движение паука с помощью генератора случайных чисел. Координаты x(t) и y(t) сохраните в одномерных массивах. Напечатайте траекторию паука в файл протокола в виде таблицы, которая содержит в клеточке с координатами x и y символ +, если паук там побывал, и символ, если он там не побывал.

4. Напишите программу, моделирующую многократный выбор двух костей домино из полного набора с помощью генератора псевдослучайных чисел. Всего имеется 28 костей, на которых написаны пары чисел 00, 10, ..., 65, 66. Смоделируйте с помощью этой программы бросание пары костей N=10000 раз и подсчитайте, сколько раз появилась пара, допускающая создание цепочки, например, 34 и 46.

5. Говорят, что целое число является совершенным числом, если его сомножители, включая 1 (но не само число) в сумме дают это число. Например, 6 – совершенное число, так как 6=1+2+3. Напишите функцию, которая определяет, является ли ее параметр n совершенным числом. Используйте эту функцию в программе, которая определяет и выводит в файл все совершенные числа в диапазоне от 1 до 1000.

Типовое задание в практико-ориентированном формате

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(МОСКОВСКИЙ ПОЛИТЕХ)

ЭКЗАМЕНАЦИОННОЕ ЗАДАНИЕ
по дисциплине
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»
направление подготовки 09.03.03 Прикладная информатика

ВРЕМЯ ВЫПОЛНЕНИЯ ЗАДАНИЯ

8 часов.

ЗАДАНИЕ ЭКЗАМЕНА

По счастливому стечению обстоятельств заказ на разработку модуля информационной системы для консалтинговой компании согласно готовому и утвержденному дизайну достался также вам. Т.к. дизайн разрабатывался именно вами, к тому же совсем недавно, то качество и пригодность исходных макетов для последующей разработки целиком и полностью лежит на вашей совести.

При анализе макета оказалось, что формировать каждый документ неудобно – поэтому было решено немного изменить алгоритм работы модуля: в меню остается пункт "Сформировать документы", при нажатии которого и заполнении формы формируются сразу все требуемые документы.

Ваша задача разработать приложение точно по макетам (с учетом указанного изменения), используя tkinter. При этом окна приложения, для которых не предоставлен макет, должны соответствовать общему стилю приложения. Также должны быть реализованы следующие функции:

- формирование счета;
- формирование счета-фактуры;
- формирование акта выполненных работ;
- формирование товарной накладной;
- формирование приходного ордера.

Особое внимание следует уделить точности вычислений цен и налогов – ведь в случае ошибки пользователи могут не только потерять деньги, но и заплатить гигантские штрафы от налоговой инспекции. Будьте очень внимательны!

СПРАВОЧНАЯ ИНФОРМАЦИЯ

СЧЕТ – документ, на основании которого покупатель оплачивает заказ. Поэтому в нем приводятся все платежные реквизиты продавца (поставщика):

- название, ИНН, КПП БИК и корреспондентский счет банка поставщика;

- название компании поставщика, ее ИНН, КПП, номер расчетного счета в банке.

Ниже обязательно указывается номер документа, дата составления, ИНН/КПП, название, адрес и телефон покупателя и продавца (поставщика). В счете указываются все товары, за которые производится оплата: номер по порядку, название товара, его количество, цена за единицу товара, сумма (произведение цены за единицу товара и его количества). Также указывается общая сумма по всем товарам цифрами и прописью (текстом). Указанные цены и суммы совпадают с тем, которые вводит пользователь при заполнении формы сделки. Однако в них уже входит НДС 20%, поэтому этот налог указывается отдельно (цена без НДС х 1.20 = указанная цена; налог НДС = указанная цена – цена без НДС). Документ подписывается только продавцом.

СЧЕТ-ФАКТУРА – документ для уплаты НДС. В нем явно указываются суммы без НДС, величина налога НДС (в процентах и в рублях), суммы с учетом НДС. Обратите внимание: при округлении учитывайте, что, если сумму без НДС сложить с налогом НДС должна получиться сумма с учетом НДСС (введенная пользователем). В счете-фактуре указывается меньше реквизитов, чем в счете. Документ подписывается только продавцом.

АКТ ВЫПОЛНЕННЫХ РАБОТ – документ, в которых стороны сделки утверждают, что и оплата, и поставка полностью произведены, претензии друг к другу отсутствуют. В нем указываются некоторые реквизиты поставщика и покупателя, перечень товаров (аналогично счету). Документ подписывается и продавцом, и покупателем.

ТОВАРНАЯ НАКЛАДНАЯ – документ, подтверждающий передачу товаров от поставщика к покупателю. В нем наиболее подробно описываются все товары с указанием их стоимости с НДС, без НДС и НДС отдельно. Обратите внимание – на каждой странице подводится промежуточный итог. В качестве допущения можно установить, что на каждой странице размещается не более двух товаров. Накладная подписывается продавцом, графы подписи покупателя остаются пустыми, т.к. неизвестно кто будет получать товар.

ПРИХОДНЫЙ ОРДЕР – документ, оформляемый при оплате наличными. Подписывается главным бухгалтером поставщика.

Образцы всех документов представлены в приложении.

УСЛОВИЯ ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения задания студент может использовать следующее программное обеспечение:

1. Microsoft Windows.
2. Notepad++.
3. Anaconda Community.
4. Python Software Foundation Python.
5. Веб-браузер, Chrome.
6. Visual Studio 2019.
7. Visual Studio Code.
8. JetBrains PyCharm Community.
9. Git.
10. СУБД MySQL.
11. СУБД PostgresSQL.
12. СУБД SQLite
13. Microsoft Office 365.

Не разрешается запускать и использовать другие программы. Не допускается использование Интернет, *flash*-накопителей, телефонов, ноутбуков, материалов на сервер. Разрешается использование лекций и печатных учебников по дизайну.

ДОСТУП К СПРАВОЧНИКАМ

Для получения справочной информации разрешается использовать следующий ресурс официальные справочные руководства используемых программных средств.

ВВОДНЫЕ ДАННЫЕ

- дамп базы данных организации;
- дизайн-макет форм приложения.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

- Работа выполняется на виртуальной машине, где установлено все необходимое программное обеспечение и размещены все необходимые данные.
- Для запуска приложения должен быть создан бинарный исполняемы файл для целевой операционной системы.
- Перед началом работы необходимо создать на рабочем столе папку (например, "WORK"), где будут размещаться промежуточные версии и данные.
- После выполнения работы приложение должно быть размещено в папке папке "*Рабочий стол/ФИО_группа*" (ФИО_группа – ваша фамилия, имя, отчество и номер группы на русском языке) - оценивается только видимый там результат.

КРИТЕРИИ ОЦЕНКИ ЗАДАНИЯ

№	Наименование критерия	Балл
	ФОРМА ВВОДА ДАННЫХ	**5**
1	Форма ввода данных создана	2
2	Осуществляется проверка на ввод некорректных данных	2
3	Формируется подтверждающее сохранение данных сообщение	1
	ТЕКСТОВАЯ ФОРМА ЧИСЕЛ	**2**
4	В формируемых документах присутствует корректная запись числа в текстовой форме	1
5	Падежи подписей к форме числа (рубль, рубля, рублей) корректны	1
	ФОРМИРОВАНИЕ СЧЕТА	**5**
6	При переходе по пункту меню и заполнении реквизитов документ формируется	1
7	Структура документа соответствует образцу	1
8	В документе отображены корректные реквизиты поставщика и покупателя, подписи	1
9	В документе корректно указаны названия, количество и суммы за товары	1
10	НДС подсчитан правильно	1

	ФОРМИРОВАНИЕ СЧЕТА-ФАКТУРЫ	**6**
11	При переходе по пункту меню и заполнении реквизитов документ формируется	1
12	Структура документа соответствует образцу	1
13	В документе отображены корректные реквизиты поставщика и покупателя, подписи	1
14	В документе корректно указаны названия, количество и суммы за товары	1
15	НДС каждого товара рассчитан правильно	1
16	Сумма НДС для каждого товара совпадает с общим НДС документа	1
	ФОРМИРОВАНИЕ АКТА ВЫПОЛНЕННЫХ РАБОТ	**6**
17	При переходе по пункту меню и заполнении реквизитов документ формируется	1
18	Структура документа соответствует образцу	1
19	В документе отображены корректные реквизиты поставщика и покупателя	1
20	В документе корректно указаны названия, количество и суммы за товары	1
21	Общая сумма НДС указана текстом	1
22	В документе отображены корректные подписи покупателя и продавца	1
	ФОРМИРОВАНИЕ НАКЛАДНОЙ	**10**
23	При переходе по пункту меню и заполнении реквизитов документ формируется	1
24	Структура документа соответствует образцу	1
25	Каждая страница документа формирует итог	1
26	Каждая страница документа печатается на отдельном листе	1
27	В документе при печати нет пустых страниц	1
28	На странице не отображаются данные с другой страницы	1
29	Форма записи цифр в колонке количества товара совпадает с образцом	1
30	Автоматически формируется количество листов в накладной	1
31	Количество наименований отображается в виде текста, а не числа	1
32	В документе отображены корректные подписи покупателя и продавца	1
	ФОРМИРОВАНИЕ ПРИХОДНОГО ОРДЕРА	**5**
33	При переходе по пункту меню и заполнении реквизитов документ формируется	1
34	Структура документа соответствует образцу	1
35	В документе отображены корректные реквизиты поставщика и покупателя, подписи	1
36	В документе корректно указаны количество и сумма	1
37	НДС подсчитан правильно	1
	СОГЛАСОВАННОСТЬ ДОКУМЕНТОВ	**1**
38	Все суммы во всех сформированных документах соответствуют друг другу (соответствующие НДС, цены, суммы и т.д. во всех документах одинаковые)	1

Результат работы оценивается согласно приведенным выше критериям, выполнение каждого из которых увеличивает результирующий баллом на указанное значение (доводится до сведения студентов только после экзамена).