

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Новосибирский национальный исследовательский
государственный университет» (Новосибирский государственный университет, НГУ)

Экономический факультет



Согласовано
Декан ЭФ
Богомолова Т.Ю.

подпись
«19» 10 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

Направление подготовки: 38.03.05 Бизнес-информатика
Направленность (профиль): Бизнес-информатика

Форма обучения: очная

Разработчики:

А.Г. Минак

Зав. кафедрой ПММЭиП
д.э.н., профессор Мкртчян Г.М.

Новосибирск, 2020

Содержание

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	3
2. Место дисциплины в структуре образовательной программы	5
3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося	5
4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий.....	6
5. Перечень учебной литературы	10
6. Перечень учебно-методических материалов по самостоятельной работе обучающихся	11
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	11
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине	11
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине	11
10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине.....	12

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Цель дисциплины:

1. Подготовить студентов к аналитической, проектной и научно-исследовательской профессиональной деятельности;
2. Развить умения и навыки студентов по использованию принципов алгоритмизации и современных методов обработки информации для решения профессиональных задач.

Основные задачи дисциплины:

1. Подготовить студентов к выполнению следующих профессиональных задач:
 - Анализ и оценка применения ИС и ИКТ для управления бизнесом;
 - Поиск, сбор, обработка, анализ и систематизация информации в экономике, управлении и ИКТ;
2. Сформировать у студентов навыки написания компьютерных программ различными стилями и умения использовать высокоуровневые способы структурирования программного кода.

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
ОК-5 Способностью к коммуникации в устной и письменной формах на русском и иностранном языках для решения задач межличностного и межкультурного взаимодействия	правила оформления программного кода PEP8	правильно оформлять программный код, используя комментарии и строки документирования, в том числе на английском языке;	
ОК-6 Способностью работать в коллективе, толерантно воспринимая социальные, этнические, конфессиональные и культурные различия	принципы взаимодействия с системами контроля версий на примере GitHub;	использовать систему PyDoc;	подходами коллективной разработки в промышленном программировании;
ОК-7 Способность к самоорганизации и самообразованию		использовать средства управления атрибутами, такие как свойства и дескрипторы, декораторы	навыками написания сложных программ, используя документацию по языку Python и другие справочные материалы

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
		функций и классов, а также метаклассы	
ОПК-1 Способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	правила видимости программных элементов в языке Python; приемы передачи аргументов функций и методов;	организовать наборы инструкций и функций в виде крупных компонентов – модулей;	навыками структурирования программ на языке Python
ОПК-3 Способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях	общую синтаксическую модель языка Python; основные встроенные типы объектов, такие как числа, строки, списки, кортежи, словари, множества;	использовать базовые коллекции языка Python (списки, кортежи, словари, множества) для решения прикладных задач; писать программы различной сложности; использовать способы упаковки программного кода многократного использования и предотвращения появления избыточного кода;	навыками написания программного кода различными стилями;
ПК-16 Умение разрабатывать контент и ИТ-сервисы предприятия и Интернет-ресурсов	инструментарий структурирования программирования для многократного использования кода; способы декомпозиции задачи и декларативного описания объектов и процессов на уровне	писать программный код на уровне функционального, объектно-ориентированного и модульного программирования, используя системы контроля версий для выполнения	навыками событийно-управляемого программирования; объектно-ориентированным стилем программирования;

Результаты освоения образовательной программы (компетенции)	В результате изучения дисциплины обучающиеся должны:		
	знать	уметь	владеть
	классов, их атрибутов и методов	коллективных разработок; разрабатывать пользовательские интерфейсы; писать объектно-ориентированный программный код;	
ПК-18 Способность использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования	классические алгоритмы обработки данных.	использовать инструменты отладки и тестирования; использовать классические алгоритмы обработки данных и применять абстрактные типы данных;	навыками построения и реализации алгоритмов для решения практических задач;

2. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование» является обязательной, преподается в 1-4 набора 2019 года в 1-3 семестре).

Дисциплина связана понятийным аппаратом с школьным курсом математики и информатики и фактическим материалом с такими курсами, как теоретические основы информатики, математический анализ, линейная алгебра.

Дисциплины (практики), для изучения которых необходимо освоение дисциплины Программирование: эконометрия, Базы данных, Основы программирования и конфигурирования в корпоративных информационных системах, Программные интерфейсы и визуализация данных, Основы разработки мобильных приложений, выполнение ВКР.

3. Трудоемкость дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Набор 2020 года:

Трудоемкость дисциплины – 13 з.е. (468 ч)

Форма промежуточной аттестации: 1 семестр - дифференцированный зачет, 3-4 экзамен

Вид деятельности	Семестр	Семестр	Семестр	Семестр
	1	2	3	4
Контактная работа, часов, в том числе:	38	76	76	40
лекции	16	32	32	16
практические занятия	16	32	32	16

групповая работа с преподавателем	4	8	8	4
контактная работа при аттестации	2	2	2	2
консультации перед экзаменом	-	2	2	2
Самостоятельная работа, часов, в том числе:	34	104	68	32
самостоятельная работа во время занятий	28	86	50	20
самостоятельная работа во время промежуточной аттестации	6	18	18	12
Всего, часов	72	180	144	72

Набор 2019 года:

Трудоемкость дисциплины – 12 з.е. (432 ч)

Форма промежуточной аттестации: 1-3 семестр экзамен

Вид деятельности	Семестр	Семестр	Семестр
	1	2	3
Контактная работа, часов, в том числе:	76	76	76
лекции	32	32	32
практические занятия	32	32	32
групповая работа с преподавателем	8	8	8
контактная работа при аттестации	2	2	2
консультации перед экзаменом	2	2	2
Самостоятельная работа, часов, в том числе:	68	68	68
самостоятельная работа во время занятий	50	50	50
самостоятельная работа во время промежуточной аттестации	18	18	18
Всего, часов	144	144	144

Набор 2017-2018 года:

Трудоемкость дисциплины – 13 з.е. (468 ч)

Форма промежуточной аттестации: 1-3 семестр - экзамен, 4 семестр – защита курсовой работы - дифференцированный зачет

Вид деятельности	Семестр	Семестр	Семестр	Семестр
	1	2	3	4
Контактная работа, часов, в том числе:	76	76	76	9
лекции	32	32	32	
практические занятия	32	32	32	
групповая работа с преподавателем	8	8	8	
индивидуальная работа с преподавателем				7
контактная работа при аттестации	2	2	2	2
консультации перед экзаменом	2	2	2	
Самостоятельная работа, часов, в том числе:	68	68	68	27
самостоятельная работа во время занятий	50	50	50	21
самостоятельная работа во время промежуточной аттестации	18	18	18	6
Всего, часов	144	144	144	36

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

№ п/п	Раздел дисциплины	Семестр	Виды учебной работы, в часах		
			лекции	практические занятия	самостоятельная работа во время занятий
1.	Начало программирования.	1	8	8	14
2.	Управляющие инструкции: ветвление	1	4	4	8
3.	Управляющие инструкции: циклы	1	4	4	6
	<i>Итого за семестр</i>	1	16	16	28
4.	Строки.	2	4	4	10
5.	Файлы и исключения.	2	4	4	10
6.	Функции как механизмы абстракции.	2	4	4	10
7.	Списки и кортежи.	2	4	4	10
8.	Множества и словари.	2	4	4	10
9.	Рекурсия.	2	4	4	10
10.	Работа с файловой системой.	2	4	4	12
11.	Классы и объекты.	2	4	4	14
	<i>Итого за семестр</i>	2	32	32	86
12.	Графические интерфейсы (GUI).	3	8	8	12
13.	Работа с медиа объектами.	3	2	2	6
14.	Функциональное программирование..	3	4	4	4
15.	Декораторы.	3	4	4	6
16.	Генераторы.	3	2	2	4
17.	Объектно-ориентированное программирование. Классы и экземпляры	3	4	4	6
18.	Методы классов.	3	4	4	6
19.	Наследование. Композиция классов.	3	4	4	6
	<i>Итого за семестр</i>	3	32	32	50
20.	Классы исключений и их обработка. Генерация исключений.	4	2	2	2
21.	Магические методы классов. Итераторы. Контекстные менеджеры	4	2	2	2
22.	Механизмы работы классов. Дискрипторы. Метаклассы.	4	2	2	2
23.	Отладка и тестирование.	4	2	2	4
24.	Многопоточное программирование. Процессы и потоки.	4	2	2	2
25.	Работа с сетью. Сокеты.	4	2	2	2
26.	Асинхронное программирование.	4	4	4	6
	<i>Итого за семестр</i>	4	16	16	20
	<i>Итого за курс</i>		96	96	184

Содержание дисциплины Программирование:

Содержание разделов	
1.	Начало программирования. Программирование. Языки программирования. Язык Python. Задачи анализа данных. IDLE. Интерактивный режим. Структура программы. Строки. Комментарии. Переменные и оператор присваивания. Типы данных. Строковые литералы. Символьные последовательности. Конкатенация строк. Строки документирования. Типы числовых данных. Целые. Длинные целые. Числа с плавающей точкой. Символы таблицы ASCII. Выражения. Арифметические выражения. Преобразование типов. Использование функций и модулей. Вызовы функций. Параметры и возвращаемые значения. Математический модуль. Основной модуль
2.	Управляющие инструкции: ветвление Форматный вывод. Условные инструкции. Логический тип. Сравнения и логические выражения. Односторонние и многострочные условные инструкции. Блочное представление кода. Уровни вложенности программного кода. Разделители блоков. Правила оформления отступов. Логические операторы и составные логические выражения.
3.	Управляющие инструкции: циклы Выполнение оператора заданное количество раз. Итераторы. Диапазоны. Циклы с параметром. Ошибки определения диапазонов. Определение шага изменения управляющей переменной цикла. Условные итерации. Цикл с условием. Структура и поведение циклов. Принудительное завершение исполнения циклов. Случайные числа.
4.	Строки. Структура строк. Доступ к символам и подстрокам в строках. Срезы. Разрезание на подстроки. Шифрование данных. Строки и системы счисления. Преобразования в системах счисления. Строковые методы.
5.	Файлы и исключения. Текстовые файлы. Текстовые файлы и их формат. Запись текста и чисел в файл. Чтение данных из файла. Доступ и управление файлами и каталогами на диске. Контекстные менеджеры with/as. Обработка исключений. Инструкции try/except/else. Инструкции try/finally. Инструкции raise, assert. Вложенные обработки исключений. Идиомы исключений.
6.	Функции как механизмы абстракции. Процедурная декомпозиция. Устранение избыточности. Скрытие сложности. Решение задач с нисходящим проектированием. Текстовый анализ. Генератор предложения. Управление пространством имен программы. Время жизни переменных. Локальные и глобальные переменные. Области видимости. Аргументы и разделяемые ссылки. Значение по умолчанию в параметрах.
7.	Списки и кортежи. Литералы списков. Базовые операции со списками. Списочные выражения. Индексы. Срезы. Матрицы. Методы работы со списками. Обход части последовательности. Изменение списков. Генерирование индексов и элементов. Литералы кортежей. Базовые операции с кортежами. Присваивание кортежа в цикле. Сравнительный анализ списков и кортежей.
8.	Множества и словари. Литералы множеств. Неупорядоченные последовательности: Множества. Отношение между множествами и словарями. Литералы словарей. Базовые операции со словарями. Методы работы со словарями. Особенности использования словарей. Использование словарей в качестве записей. Ссылки и копии объектов. Сравнение, равенство, истина. Сравнение словарей. Объект None. Конструирование словаря. Реализации списка множеств и словарей. Анализ сложности реализаций списка множеств и словарей. Хеширование.

	Хеширование в реализации словарей. Хеширование в реализации множеств. Упорядоченные множества и словари.
9.	Рекурсия. Итеративные и рекурсивные подходы. Прямая рекурсия. Косвенная рекурсия. Классические рекурсивные алгоритмы. Трассировка рекурсивной функции. Использование рекурсивных определений. Бесконечная рекурсия. Преимущества рекурсии.
10.	Работа с файловой системой. os и os.path пути и имена. Операции с файловой системой. Применение рекурсивных алгоритмов.
11.	Классы и объекты. Объекты. Классы. Атрибуты. Методы. Наследование. Полиморфизм. Инкапсуляция. Множество экземпляров. Определения метода. Метод <code>__init__</code> и переменные экземпляра. Метод <code>__str__</code> . Время жизни объектов. Правила для определения простого класса. Примеры моделирования данных. Дерево классов. Рациональные числа. Арифметика рационального числа и перегрузка. Сравнения и метод <code>__cmp__</code> . Равенство и метод <code>__eq__</code> . Структурирование классов с наследованием и полиморфизмом. Иерархии наследования. Полиморфные методы. Абстрактные классы. Многократное использование программного кода.
12.	Графические интерфейсы (GUI). Событийно-управляемое программирование. Разработка простых программ с использованием GUI-интерфейса. Метки. Изображения. Кнопки и события ответа Поля ввода текста. Диалоговые окна. События по нажатию мыши. События клавиатуры.
13.	Работа с медиа объектами. Создание анимированных объектов. Работа со звуком и музыкой.
14.	Функциональное программирование. Передача функций в качестве параметров. Функции внутри других функций. Функции-обертки. Функции <code>map</code> , <code>filter</code> . Анонимные функции. Примеры lambda-выражений. Модуль <code>functools</code> . Использование метода <code>partial</code> . Работа со списочными выражениями.
15.	Декораторы. Назначение декораторов. Синтаксический сахар. Сложные примеры декораторов. Использование нескольких декораторов.
16.	Генераторы. Назначение генераторов. Механизм работы генератора. Особенности использования инструкции <code>yield</code> . Примеры применения генераторов. Передача параметров генератору.
17.	Объектно-ориентированное программирование. Классы и экземпляры Классы, объекты, экземпляры. Правила описания класса. Инициализация. Методы строкового представления. Работа с атрибутами объектов. Атрибуты классов. Строки документирования классов. Конструкторы экземпляров класса.
18.	Методы классов. Объявления и вызовы методов. Методы класса <code>@classmethod</code> . Статические методы <code>@staticmethod</code> . Декораторы методов. <code>.setter</code> , <code>.getter</code> , <code>.deleter</code> . Практика описания классов.

19.	Наследование. Композиция классов. Принципы наследования. Функция super(). Классы-наследники. Примеры композиции классов.
20.	Классы исключений и их обработка. Генерация исключений. Виды исключений. Класс BaseException. Примеры AttributeError, IndexError, KeyError, TypeError. Иерархия классов исключений. Генерация исключений. Примеры исключений.
21.	Магические методы классов. Итераторы. Контекстные менеджеры. Назначение магических методов. Примеры использования и переопределения магических методов. Классы-декораторы. Итераторы. Контекстные менеджеры. Примеры использования итераторов и контекстных менеджеров.
22.	Механизмы работы классов. Дискрипторы. Метаклассы. Назначение дискрипторов. Примеры использования дискрипторов. Механизмы работы классов. Метоклассы.
23.	Отладка и тестирование. Отладка. Pdb. Тестирование. Unittest. Назначение и примеры использования unittest.
24.	Многопоточное программирование. Процессы и потоки. Процессы и его характеристики. Создание процессов и потоков. Синхронизация потоков. Глобальная блокировка интерпретатора.
25.	Работа с сетью. Сокеты. Сокеты. Клиент-сервер. Таймауты и обработка сетевых ошибок. Обработка нескольких соединений.
26.	Асинхронное программирование. Исполнение кода в одном потоке. Механизмы использования генераторов и итераторов. Карутины. Asyncio.

5. Перечень учебной литературы

5.1 Основная литература

1. Шелудько, В.М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули / В.М. Шелудько ; Министерство науки и высшего образования РФ, Южный федеральный университет, Институт компьютерных технологий и информационной безопасности. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500060> (дата обращения: 12.11.2020). – Библиогр. в кн. – ISBN 978-5-9275-2648-2. – Текст : электронный.

2. Шелудько, В.М. Основы программирования на языке высокого уровня Python : учебное пособие / В.М. Шелудько ; Министерство науки и высшего образования РФ, Южный федеральный университет, Инженерно-технологическая академия. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500056> (дата обращения: 12.11.2020). – Библиогр. в кн. – ISBN 978-5-9275-2649-9. – Текст : электронный.

3. Буйначев, С.К. Основы программирования на языке Python : учебное пособие / С.К. Буйначев, Н.Ю. Боклаг ; Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. – Екатеринбург : Издательство Уральского университета, 2014. – 92 с. : табл., ил. – Режим доступа: по подписке. – URL:

<https://biblioclub.ru/index.php?page=book&id=275962> (дата обращения: 12.11.2020). – Библиогр. в кн. – ISBN 978-5-7996-1198-9. – Текст : электронный.

5.2 Дополнительная литература

4. Magnus Lie Hetland “Python Algorithms: Mastering Basic Algorithms in the Python Language” / Apress 2010 Online ISBN 978-1-4302-3238-4 URL: <https://link.springer.com/book/10.1007/978-1-4302-3238-4> (дата обращения: 12.11.2020)

5. Irv Kalb “Learn to Program with Python” / Learn to Program with Python 2016 Online ISBN Online ISBN URL: <https://link.springer.com/book/10.1007/978-1-4842-2172-3> (дата обращения: 12.11.2020)

6. Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2020. — 161 с. — (Высшее образование). — ISBN 978-5-534-10971-9. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454100> (дата обращения: 12.11.2020).

6. Перечень учебно-методических материалов по самостоятельной работе обучающихся

7. Густокашин М. Основы программирования на Python / М: Высшая школа экономики, 2017 (URL: <https://www.coursera.org/learn/python-osnovy-programmirovaniya>)

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Освоение дисциплины используются следующие ресурсы:

- электронная информационно-образовательная среда НГУ (ЭИОС);
- образовательные интернет-порталы;
- информационно-телекоммуникационная сеть Интернет.

Взаимодействие обучающегося с преподавателем (синхронное и (или) асинхронное) осуществляется через личный кабинет студента в ЭИОС, электронную почту.

7.1 Современные профессиональные базы данных:

Базы данных не используются

7.2. Информационные справочные системы

Информационные справочные системы не используются

Полезные зарубежные электронные ресурсы в открытом доступе:

—сайт, созданный для изучения различных языков программирования.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

интерпретатор Python 3.7, среда разработки JetBrains PyCharm Professional 2018, Windows и Microsoft Office

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации дисциплины Программирование используются специальные помещения:

1. Учебные аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации;

2. Помещения для самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации большой аудитории.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду НГУ.

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».

10. Оценочные средства для проведения текущего контроля и промежуточной аттестации по дисциплине

Перечень результатов обучения по дисциплине Программирование и индикаторов их достижения представлен в виде знаний, умений и владений в разделе 1.

10.1 Порядок проведения текущего контроля и промежуточной аттестации по дисциплине

Текущий контроль успеваемости:

После окончания темы (или нескольких тем) студентам предоставляется возможность выполнения индивидуальных практических работ (ИПР), контрольных работ, кейсов, тестов.

Итоговая оценка за семестр рассчитывается на основе полученной суммы баллов, и баллов, набранных при заключительном контроле знаний на зачете.

В таблицах приводится количество баллов, которое можно набрать в учебном семестре:

Текущий контроль					Зачет (дифференцированный) Экзамен	Итого
Индивидуальные практические работы	Контрольные работы	Кейсы	Тест	Итого		
30	15	10	5	60	40	100

Промежуточная аттестация:

Формой промежуточной аттестации по дисциплине является дифференцированный зачет 1-семестр, экзамен 3-4 семестры, которые проводятся в форме решения задач, в виде написания программ. Программы проходят автоматическую проверку, по результату которой формируются баллы за промежуточную аттестацию. Максимальное количество баллов на экзамене и зачете – 40. Таким образом, максимальное количество баллов, которые можно набрать по курсу – 100 баллов.

Критерии и шкалы оценивания заданий из оценочных средств

Баллы, набранные за выполнение заданий текущего контроля и промежуточной аттестации, конвертируются в оценку по дисциплине следующим образом:

Итоговая сумма набранных баллов	Оценка
≤ 40	неудовлетворительно
от 40,1 до 60	удовлетворительно
от 60,1 до 80	хорошо
от 80,1 до 100	отлично

Описание критериев и шкал оценивания индикаторов достижения результатов обучения по дисциплине

Таблица 10.1

Код компетенции	Результат обучения по дисциплине	Оценочное средство
ОК-5	Знание правил оформления программного кода PEP8	ИПР Кейсы Зачет/Экзамен
	Умение правильно оформлять программный код, используя комментарии и строки документирования, в том числе на английском языке;	ИПР Контрольные работы Кейсы Зачет/Экзамен
ОК-6	Знание принципов взаимодействия с системами контроля версий на примере GitHub;	ИПР Кейсы Зачет/Экзамен
	Умение использовать систему PyDoc;	ИПР Контрольные работы Кейсы Зачет/Экзамен
	Владение подходами коллективной разработки в промышленном программировании;	Кейсы
ОК-7	Умение использовать средства управления атрибутами, такие как свойства и дескрипторы, декораторы функций и классов, а также метаклассы	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Владение навыками написания сложных программ, используя документацию по языку Python и другие справочные материалы	ИПР Контрольные работы Кейсы Зачет/Экзамен
ОПК-1	Знание правил видимости программных элементов в языке Python; приемов передачи аргументов функций и методов;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Умение организовать наборы инструкций и функций в виде крупных компонентов – модулей;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Владение навыками структурирования программ на языке Python	ИПР Контрольные работы Кейсы

		Тест Зачет/Экзамен
ОПК-3	Знание общей синтаксической модели языка Python; основных встроенных типов объектов, таких как числа, строки, списки, кортежи, словари, множества;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Умение использовать базовые коллекции языка Python (списки, кортежи, словари, множества) для решения прикладных задач; писать программы различной сложности; использовать способы упаковки программного кода многократного использования и предотвращения появления избыточного кода;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Владение навыками написания программного кода различными стилями;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
ПК-16	Знание инструментария структурирования программирования для многократного использования кода; способов декомпозиции задачи и декларативного описания объектов и процессов на уровне классов, их атрибутов и методов	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Умение: писать программный код на уровне функционального, объектно-ориентированного и модульного программирования, используя системы контроля версий для выполнения коллективных разработок; разрабатывать пользовательские интерфейсы; писать объектно-ориентированный программный код;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Владение навыками событийно-управляемого программирования; объектно-ориентированным стилем программирования;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
ПК-18	Знание классических алгоритмов обработки данных.	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Умение использовать инструменты отладки и тестирования; использовать классические алгоритмы обработки данных и применять абстрактные типы данных;	ИПР Контрольные работы Кейсы Тест Зачет/Экзамен
	Владение навыками построения и реализации алгоритмов для решения практических задач;	ИПР Контрольные работы Кейсы Зачет/Экзамен

Таблица 10.2

Уровень сформированности компетенций	Баллы, оценка	Полнота знаний	Наличие умений	Наличие навыков
Высокий	80 – 100, «отлично»	Демонстрирует высокий уровень знаний в объеме, соответствующем программе курса	Демонстрирует высокий уровень умений без недочетов	Демонстрирует навыки на высоком уровне без ошибок
Средний	60 – 79, «хорошо»	Знает достаточно в базовом объеме, но допускает погрешности	Демонстрирует умения на базовом уровне, допуская недочёты	Демонстрирует базовые навыки с некоторыми недочетами
Минимально допустимый	40 – 59, «удовлетворительно»	Демонстрирует частичные знания без грубых ошибок	Демонстрирует частичные умения без грубых ошибок	Демонстрирует частичное владение без грубых ошибок
Компетенция не сформирована	0 – 39, «неудовлетворительно»	Знания не продемонстрированы, допускает грубые ошибки	Не продемонстрированы основные умения, допускает грубые ошибки	Не продемонстрированы базовые навыки, допускает грубые ошибки

Типовые контрольные задания и иные материалы, необходимые для оценки результатов обучения

Примеры задач для контрольных работ

Перечень тем индивидуальных практических работ(ИПР):

1. Простейшие программы
2. Ветвление
3. Циклы
4. Строки
5. Файлы и исключения
6. Функции
7. Списки и кортежи
8. Множества и словари
9. Рекурсия
10. Классы и объекты.
11. Работа с медиа объектами.

Перечень заданий в форме тематического исследования (кейс) (КТИ(к)):

1. Калькулятор налога на прибыль
2. Прогрессивный налог
3. Инвестиционный отчет
4. Анализ текста
5. Парсинг web-страниц

6. Тесселяция
7. Генерация предложений
8. Визуализация данных
9. Робот-программа (бот)
10. Фракталы
11. Файловая система
12. Игра «Жизнь»
13. Пользовательский интерфейс к симулятору банкомата
14. Симулятор банкомата
15. Моделирование обслуживания потока клиентов
16. Система обмена сообщениями

Перечень контрольных работ(КР):

1. Начало программирования
2. Ветвление
3. Циклы
4. Строки
5. Текстовые файлы и исключения
6. Функции
7. Списки и кортежи
8. Множества и словари
9. Рекурсия
10. Классы и объекты

Пример индивидуальной практической работы:

Практическая работа № 8. МНОЖЕСТВА И СЛОВАРИ

1. В строке задается множество натуральных чисел, разделенных пробелом. Проверьте принадлежит ли число, указанное во второй строке к множеству повторяющихся чисел.

Входные данные	Выходные данные
1 2 3 2 1 7 3 5 2	YES
1 2 3 2 1 7 3 5 7	NO

2. В первой строке указывается натуральное число N - количество студентов. В последующих N строках указываются факультативные курсы, которые выбрал каждый студент. Курсы указываются через пробел. Определите сколько курсов выбрали все студенты.

Входные данные	Выходные данные
5 История Биология Программирование Физика Физика История Программирование География Программирование История Программирование Физика История Биология Математика Программирование История Физика	2

3. То, что нравится Сладкоежкину, записано через пробел в первой строке. В следующей строке указано натуральное число N - количество друзей Сладкоежкина. В следующих N строках находятся предпочтения каждого из его друзей. Определите сколько наименований продуктов нравится только Сладкоежкину.

Входные данные	Выходные данные
Абрикос Авокадо Айва Ананас Апельсин Бананы Виноград Гранат Грейпфрут Груша Дыня Инжир Киви Лангсат Лимон Манго Орех Папайя Персик Яблоко Слива Тыква Финик 3 Гранат Груша Дыня Инжир Киви Лангсат Манго Папайя Персик Яблоко Слива Тыква Финик Абрикос Авокадо Киви Лангсат Манго Папайя Персик Айва Апельсин Бананы Виноград Гранат Груша Дыня Инжир Киви Лангсат Манго Папайя	4

4. В двух строках представлены элементы множеств, разделенные пробелами. Определить принадлежит ли значение, расположенное в третьей строке пересечению этих множеств.

Входные данные	Выходные данные
2 8 3 1 7 4 1 3 10 14 28 13 7	NO

6 11 84 0 33 12 5 66 11 22 102 12 65 12	YES
---	-----

5. Для заданного натурального числа n определить все простые числа меньше n . Для решения задачи использовать алгоритм решета Эратосфена и множества.

Входные данные	Выходные данные
22	2 3 5 7 11 13 17 19

6. Напишите программу для расшифровки равенства $XOD+XOD+XOD=MAT$, где буквам должны соответствовать цифры.

Ответы не должны содержать пробелов. Программа должна выводить все варианты расшифровки от наименьшего значения XOD до наибольшего.

Начало вариантов следующее:

$$108+108+108=324$$

$$109+109+109=327$$

$$126+126+126=378$$

$$129+129+129=387$$

...

7. В строке задается множество натуральных чисел, разделенных пробелом. Получите все перестановки этого множества в лексикографическом порядке.

Подсказка: Найдите и используйте соответствующий метод из модулей Python.

Входные данные	Выходные данные
2 1 3	1 2 3 1 3 2 2 1 3 2 3 1 3 1 2 3 2 1

8. В строке задается множество натуральных чисел, разделенных пробелом. Получите список всех подмножеств для заданного множества.

Входные данные	Выходные данные
1 2 3	{}, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3}

9. В первой строке задается множество натуральных чисел, разделенных пробелом. Их количество N . Во второй строке вводится натурально число K ($K \leq N$). Получите список всех K -элементных подмножеств для заданного множества.

Входные данные	Выходные данные
----------------	-----------------

1 2 3 4 5 3	{{1, 2, 3}, {1, 2, 4}, {1, 2, 5}, {2, 3, 4}, {2, 3 5}, {2, 4, 5}, {3, 4, 5}}
----------------	--

10. Дана строка. Вывести слова в порядке уменьшения их частоты вхождения в эту строку.

Входные данные	Выходные данные
пример слово слово бочка март пример март пример слово слово слово	слово пример март бочка

11. Русско-английский словарь задан N парами слов. В первой строке вводится натуральное число N - количество пар слов словаря, далее N строк - пары слов, в последней строке фраза, которую необходимо перевести. Если слова в словаре отсутствуют, то слово должно остаться без перевода.

Входные данные	Выходные данные
10 дождь rain я I этом this в in хороший good городе city погода weather собака dog могу can много many в этом городе всегда хорошая погода	in this city всегда хорошая weather

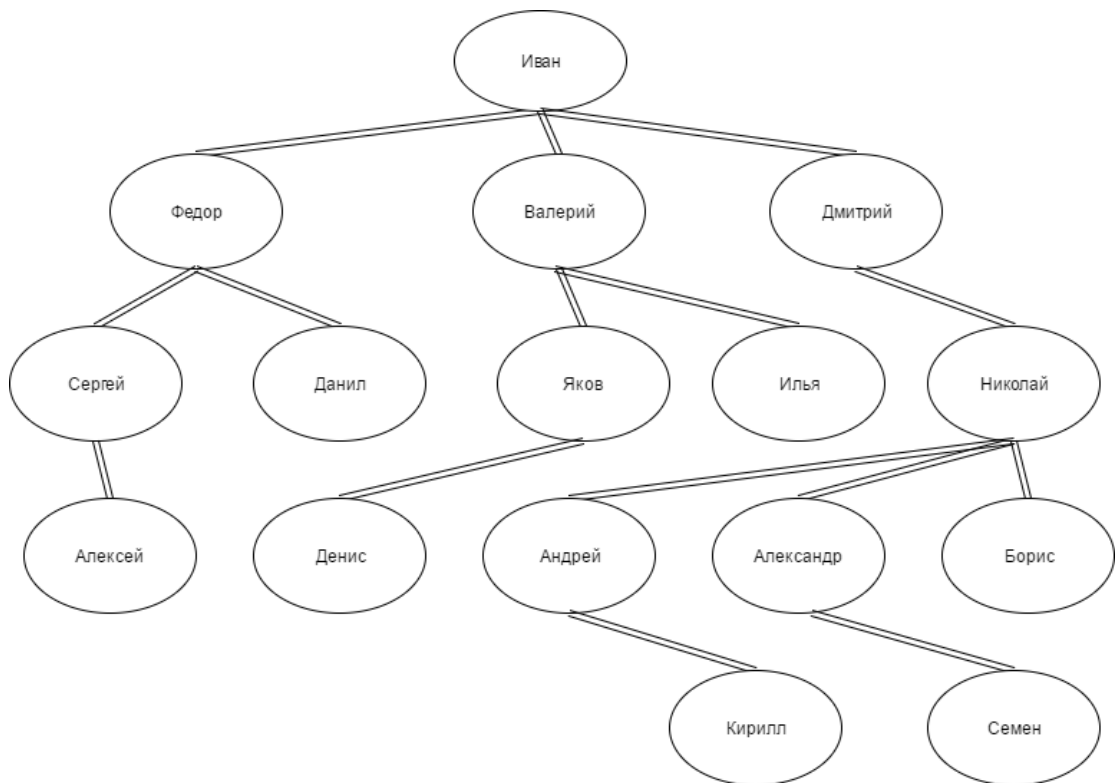
12. В первой строке вводится количество элементов словаря - натуральное число N. Далее, вводятся N строк содержащие пары слов антонимов через пробел. Определить антоним слова, указанного в последней строке. В случае если слово отсутствует в словаре, то оно должно выводиться без изменений.

Входные данные	Выходные данные
10 активность инертность безволие воля белеть чернеть верх низ война мир доброта злоба земля небо звонкий глухой огромный маленький полезно вредно верх	низ

13. В первой строке вводится количество элементов словаря - натуральное число N. Далее, вводятся N строк. Слова в каждой строке разделены пробелом. Первое слово в каждой строке - это форма предмета, следующие слова - предметы. Определить форму предмета, указанного в последней строке.

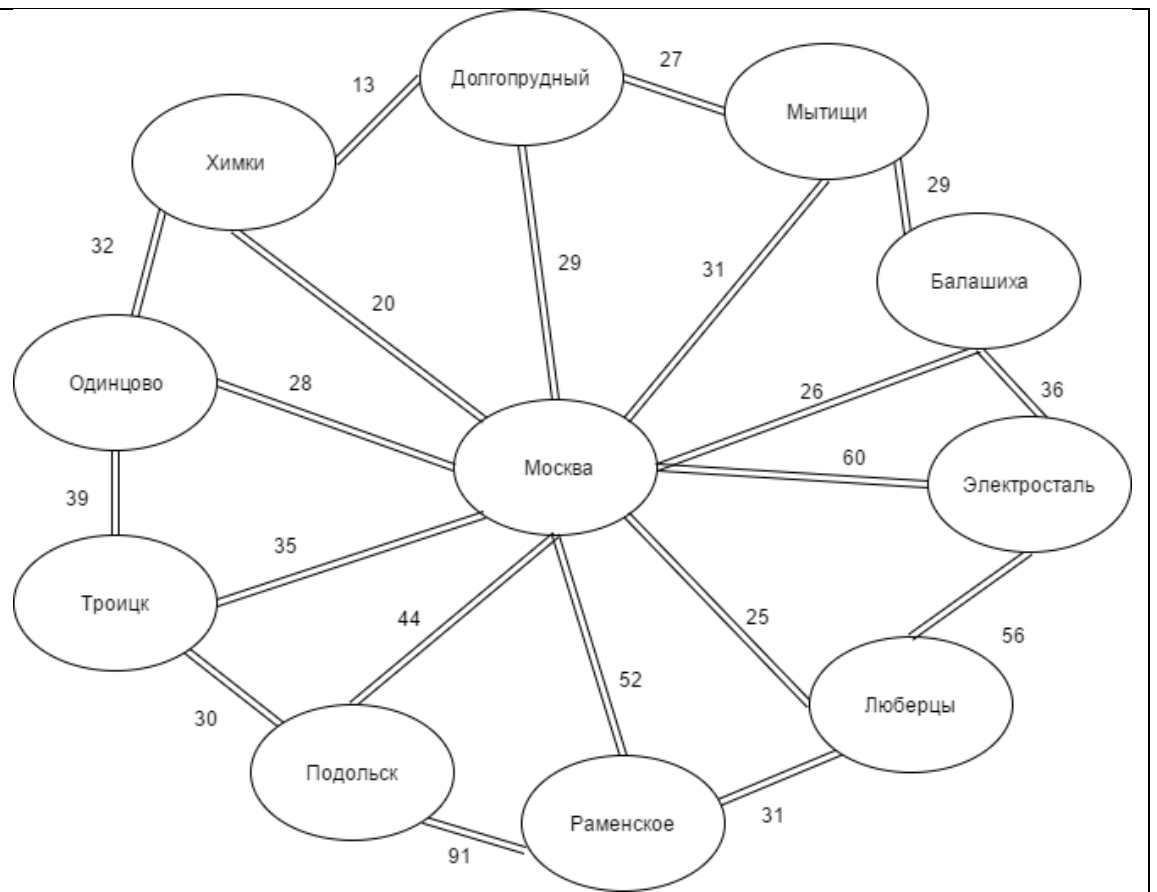
Входные данные	Выходные данные
5 круглый Солнце апельсин мяч вишня квадратный окно аквариум платок овальный яйцо дыня слива треугольный пирамида ель шатер прямоугольный книга дверь открытка конверт телевизор дыня	овальный

14. Генеалогическое древо представлено в следующем виде. В первой строке одно натуральное число N - количество родительских отношений. Далее следует N строк. В каждой строке указываются два имени через пробел. Первое - имя родителя, второе - имя потомка. Напишите рекурсивную функцию, которая определяет количество потомков (внуков, правнуков и т.д.) для заданного имени, указанного в последней строке. Для наглядности приведен рисунок.



Входные данные	Выходные данные
15 Иван Федор Иван Валерий Иван Дмитрий Федор Сергей Федор Данил Валерий Яков Валерий Илья Дмитрий Николай Сергей Алексей Яков Денис Николай Андрей Николай Александр Николай Борис Андрей Кирилл Александр Семен Николай	5

15. Известно расстояние между населенными пунктами. В первой строке вводится натуральное число N - количество населенных пунктов, во второй строке - число M - количество дорог. Далее M строк, каждая из которых содержит два названия населенных пунктов и расстояние между ними. Напишите программу, определяющую кратчайшее расстояние между населенными пунктами, указанными в последней строке. Для наглядности приведен рисунок.



Входные данные	Выходные данные
11	
20	
Москва Долгопрудный 29	
Москва Мытищи 31	
Москва Балашиха 26	
Москва Электросталь 60	
Москва Люберцы 25	
Москва Раменское 52	
Москва Подольск 44	
Москва Троицк 35	
Москва Одинцово 28	
Москва Химки 20	45
Долгопрудный Мытищи 27	
Мытищи Балашиха 29	
Балашиха Электросталь 36	
Электросталь Люберцы 56	
Люберцы Раменское 31	
Раменское Подольск 91	
Подольск Троицк 30	
Троицк Одинцово 39	
Одинцово Химки 32	
Химки Долгопрудный 13	
Химки Люберцы	

Пример индивидуального мини-проекта:

Мини-проект № 6. ФУНКЦИИ

Вариант 1. Куча фасоли

Мы собираемся сыграть в игру под названием "куча фасоли". В начале игры мы имеем кучу фасоли (из 16 бобов). В игру играют два игрока. Каждый игрок может взять 1, 2 или 3 боба из этой кучи. Пропускать ход нельзя. Игрок, который берет последний из бобов - проигрывает. Напишите функцию хода игрока, которая имеет один аргумент - количество бобов в куче, и возвращает оставшееся количество бобов после хода игрока. В основной программе вызываются по очереди функции хода первого и второго игроков. Напишите функцию «умного игрока». Имена функций должны быть *player1* и *player2*. Эти функции должны быть одинаковыми. На семинарском занятии, будет происходить тестирование Вашей функции с другими участниками. Кто придумает алгоритм с лучшей стратегией?

Вариант 2. Секвенирование ДНК

Термин секвенирование ДНК относится к способам определения порядка аминокислотной последовательности в молекуле ДНК. В строке используются первые буквы: АТСГ. Строки ДНК - это миллионы таких символов. Для того чтобы определить, содержится ли некоторая последовательность в ДНК, нужно выполнить поиск подстроки в строке. Эта задача играет важную роль в восстановлении неизвестной строки ДНК и исследования струны ДНК. В языке Python существует метод `find`, имеющий параметры: подстрока, позиция начала поиска, позиция окончания поиска. Этот метод возвращает индекс первого вхождения подстроки в строке, либо -1 в случае, если подстрока не найдена. Не используя метод `find`, напишите функцию, которая ведет себя точно так же. Поскольку ваша функция не является строковым методом, то строка в которой ведется поиск так же должна быть параметром функции. Исследователям структур ДНК часто хочется найти все места, где встречается подстрока. Напишите так же другую функцию, которая возвращает строку, содержащую все индексы найденных вхождений подстроки в строке, разделенные запятыми. Если подстрока не найдена, то функция должна возвращать пустую строку.

Линейный поиск подстроки в строке является не эффективным. Рекомендуется использовать более эффективные алгоритмы, например, алгоритм поиска подстроки в строке Бойера-Мура.

Вариант 3. Орнамент

Орнамент - это графическое украшение из сочетания геометрических, растительных или животных элементов. Как правило, орнамент состоит из повторяющихся графических элементов.

Разработайте не менее трех функций, каждая из которых выводит какой-либо графический элемент. Используя эти функции, напишите программу, которая выводит орнамент из этих элементов. Обязательным условием выполнения данного задания является цветное изображение орнамента и использование заливки цветом.

Вариант 4. Проверка номера кредитной карты

Номера кредитных карт определены следующим шаблоном:

Номер должен иметь от 13 до 16 цифр, и номер должен начинаться с:

- 4 для карт Visa
- 5 для кредитных карт MasterCard
- 37 для карт American Express
- 6 для карт Discover
- 2 для карт национальной платежной системы "Мир"

В 1954 году Ганс Лун из IBM предложил алгоритм для проверки номеров кредитных карт.

Ниже приведен этот алгоритм:

1. Цифры проверяемой последовательности нумеруются справа налево, начиная с 1.
2. Цифры, оказавшиеся на нечётных местах, остаются без изменений.

3. Цифры, стоящие на чётных местах, умножаются на 2.
 4. Если в результате такого умножения возникает число больше 9, оно заменяется суммой цифр получившегося произведения — однозначным числом, то есть цифрой.
 5. Все полученные в результате преобразования цифры складываются. Если сумма кратна 10, то исходные данные верны.
- Напишите программу, которая определяет тип карты и проверяет является ли номер карты правильным. Выполните декомпозицию решения задачи и представьте решение отдельных подзадач в виде функций.

Пример тематического исследования (кейс):

Кейс № 15. МОДЕЛИРОВАНИЕ ОБСЛУЖИВАНИЯ ПОТОКА КЛИЕНТОВ

Описание предметной области

Экономическое моделирование - это воспроизведение экономических объектов и процессов в малых, экспериментальных формах, в искусственно созданных условиях (натурное моделирование). В экономике чаще используется математическое моделирование посредством описания экономических процессов математическими зависимостями. Моделирование служит предпосылкой и средством анализа экономики и протекающих в ней явлений и обоснования принимаемых решений, прогнозирования, планирования, управления экономическими процессами и объектами.

Задача

Требуется создать программу, автоматизирующую управление занятостью номеров гостиницы. Программа должна обрабатывать входной поток заявок.

Анализ

Небольшая гостиница содержит K номеров ($20 \leq K \leq 30$), различающихся по типу:

1. Одноместные;
2. Двухместные;
3. Полулюкс;
4. Люкс.

По степени комфортности номера различаются:

1. Стандарт;
2. Стандарт улучшенный;
3. Апартамент.

Каждый номер обладает максимальной вместимостью (по количеству человек). Также предусмотрено питание:

1. Без питания;
2. Завтрак;
3. Полупансион.

Д
а
н
п
р
и
м

- 1 *одноместный 1 стандарт*
- 2 *двухместный 2 стандарт_улучшенный*
- 3 *люкс 3 апартамент*
- 4 *полулюкс 3 стандарт_улучшенный*
- 5 *одноместный 1 стандарт*
- 6 *одноместный 1 стандарт_улучшенный*
- 7 *одноместный 1 апартамент*
- 8 *двухместный 2 апартамент*
- 9 *двухместный 2 стандарт_улучшенный*
- 10 *одноместный 1 стандарт*
- 11 *одноместный 1 стандарт_улучшенный*
- 12 *двухместный 2 стандарт*
- 13 *люкс 3 апартамент*
- 14 *люкс 5 апартамент*

- 15 люкс 6 апартамент
- 16 полулюкс 2 стандарт_улучшенный
- 17 полулюкс 1 апартамент
- 18 одноместный 1 стандарт
- 19 двухместный 2 стандарт_улучшенный
- 20 двухместный 2 стандарт
- 21 одноместный 1 стандарт_улучшенный
- 22 одноместный 1 стандарт_улучшенный
- 23 полулюкс 2 апартамент
- 24 полулюкс 2 стандарт_улучшенный

Стоимость номера должна рассчитываться исходя из данных, представленных в виде словарей.

Приведена стоимость номера в сутки из расчета на одного человека:

Тип номера	Стоимость, руб.
Одноместный	2900,00
Двухместный	300,00
Полулюкс	3200,00
Люкс	4100,00

Коэффициенты, повышающие стоимость проживания в номере:

Степень комфортности	Коэффициент
Стандарт	1,0
Стандарт улучшенный	1,2
Апартамент	1,5

При бронировании назначается тип питания, информация о котором указана также в отдельном словаре:

Питание	Стоимость на одного человека в сутки, в руб.
Без питания	0,00
Завтрак	280,00
Полупансион	1000,00

Требуется создать программу, автоматизирующую управление занятостью номеров гостиницы. Программа обрабатывает входной поток заявок.

Информация о бронировании поступает из файла booking.txt

Каждая строка файла содержит следующую информацию:

<дата бронирования> <Фамилия> <Имя> <Отчество> <количество человек> <дата въезда>
<количество суток> <максимальная сумма на одного человека, которую готов потратить клиент за сутки>

Пример файла booking.txt

```
01.03.2018 Жиренкова Надежда Евдокимовна 1 01.03.2018 3 4400
01.03.2018 Мясников Виссарион Яковович 1 01.03.2018 1 3500
01.03.2018 Бузинская Альбина Кирилловна 1 03.03.2018 1 2200
01.03.2018 Канадов Самуил Севастьянович 2 01.03.2018 1 6600
01.03.2018 Грядкин Порфирий Кондратиевич 4 03.03.2018 2 8100
01.03.2018 Нефёдова Марфа Потаповна 3 03.03.2018 1 2200
01.03.2018 Архипов Никон Афанасиевич 1 01.03.2018 2 3400
01.03.2018 Ежова Лада Феликсовна 1 01.03.2018 2 5800
01.03.2018 Никитаев Евгений Иннокентиевич 1 01.03.2018 3 3400
01.03.2018 Шайн Лев Александрович 5 05.03.2018 3 46400
```

01.03.2018 Архаткин Дмитрий Юриевич 2 01.03.2018 3 2400
01.03.2018 Водолеев Николай Давыдович 1 01.03.2018 2 3000
01.03.2018 Сильвестров Евдоким Маркович 1 01.03.2018 2 3100
01.03.2018 Калугин Богдан Карлович 1 01.03.2018 2 3700
01.03.2018 Вершинин Борислав Евграфович 1 01.03.2018 3 3700
01.03.2018 Туржанская Марта Казимировна 1 01.03.2018 3 3700
01.03.2018 Аксёнов Лаврентий Семенович 3 03.03.2018 2 4600
01.03.2018 Капшукова Елена Валентиновна 1 01.03.2018 3 3600
01.03.2018 Грызлов Аскольд Никифорович 1 01.03.2018 2 3300
01.03.2018 Большой Эммануил Архипович 2 01.03.2018 4 6000
01.03.2018 Жернаков Родион Гордеевич 2 02.03.2018 2 5600
01.03.2018 Ярошевич Антонина Сократовна 2 02.03.2018 1 4300
01.03.2018 Юдицкий Осип Адамович 1 01.03.2018 1 3500
02.03.2018 Непеина Полина Адамовна 1 03.03.2018 1 3800
02.03.2018 Дудник Вячеслав Денисович 2 03.03.2018 2 3700
02.03.2018 Яснова Эвелина Кировна 2 03.03.2018 1 3200
02.03.2018 Хомичева Татьяна Афанасиевна 1 04.03.2018 3 4000
02.03.2018 Колосов Федот Ефремович 3 02.03.2018 3 2800
02.03.2018 Выгузова Рената Пахомовна 1 03.03.2018 1 5200
02.03.2018 Волгина Ника Мироновна 2 02.03.2018 3 5400
02.03.2018 Барановский Прохор Евгениевич 2 03.03.2018 3 3300
02.03.2018 Волков Феофан Феоктистович 3 03.03.2018 2 3000
02.03.2018 Заболотный Наум Федорович 2 02.03.2018 3 6100
02.03.2018 Свечин Карл Ильевич 2 03.03.2018 3 4200
02.03.2018 Маринина Ефросинья Самуиловна 1 03.03.2018 2 4200
02.03.2018 Увакин Мстислав Филимонович 1 03.03.2018 2 4000
02.03.2018 Ульянов Юлиан Эрнестович 3 02.03.2018 3 5200
02.03.2018 Чечурова Рада Вениаминовна 2 02.03.2018 3 6800
02.03.2018 Якова Анастасия Борисовна 1 03.03.2018 1 3200
02.03.2018 Бугайчук Денис Мартьянович 1 04.03.2018 1 6500
02.03.2018 Какошин Никифор Леонтиевич 5 03.03.2018 3 4100
03.03.2018 Малинин Архип Эмилевич 1 05.03.2018 4 4600
03.03.2018 Хомколов Елизар Остапович 1 05.03.2018 2 2500
03.03.2018 Максимушкина Ева Порфириевна 2 05.03.2018 3 4400
03.03.2018 Песков Георгий Алексеевич 4 03.03.2018 1 6000
03.03.2018 Сиянковская Дина Родионовна 1 05.03.2018 3 2200
03.03.2018 Яковченко Егор Чеславович 1 03.03.2018 1 3800
03.03.2018 Ушаков Андрей Ипатович 1 04.03.2018 3 3300
03.03.2018 Игнатов Лукьян Сидорович 1 05.03.2018 3 6500
03.03.2018 Ерастова Раиса Вацлавовна 1 03.03.2018 2 5700
03.03.2018 Якущенко Антон Ираклиевич 2 03.03.2018 1 2300
03.03.2018 Ивашечкина Кристина Сидоровна 3 04.03.2018 3 6100
03.03.2018 Голубчиков Ким Кондратович 1 03.03.2018 1 3500
03.03.2018 Гурковский Эмиль Онуфриевич 1 03.03.2018 3 3300
03.03.2018 Ямщикова Марина Гаврииловна 1 04.03.2018 2 3500
03.03.2018 Крутина Злата Ипполитовна 2 05.03.2018 2 3300
03.03.2018 Ярков Сидор Еремеевич 2 03.03.2018 2 5100
03.03.2018 Северин Ростислав Филиппович 3 04.03.2018 3 3800

03.03.2018 Быстров Иннокентий Макарович 2 04.03.2018 3 5500
03.03.2018 Кузьмов Олег Тимурович 2 03.03.2018 2 6500
03.03.2018 Киселев Яков Андриянович 1 04.03.2018 1 2700
04.03.2018 Ожогина Берта Соломоновна 1 05.03.2018 3 2300
04.03.2018 Янушевский Клавдий Елизарович 1 06.03.2018 1 5800
04.03.2018 Камалова Владислава Платоновна 3 05.03.2018 1 6800
04.03.2018 Сотов Якуб Потапович 2 05.03.2018 1 2300
04.03.2018 Малыхина Майя Владиленовна 1 04.03.2018 2 3500
04.03.2018 Супрунова Лидия Михайловна 1 06.03.2018 5 3100
04.03.2018 Ясюевич Константин Кириллович 1 06.03.2018 2 5200
04.03.2018 Эверлаков Потап Гаврилевич 1 06.03.2018 2 2500
04.03.2018 Курташкина Александра Никоновна 5 05.03.2018 3 5100
04.03.2018 Квасков Фома Богданович 2 06.03.2018 2 5600
04.03.2018 Носкова Зоя Клавдиевна 1 06.03.2018 3 3100
04.03.2018 Арефьев Давид Демьянович 2 06.03.2018 3 4700
04.03.2018 Загряжская Эмма Дмитриевна 2 05.03.2018 1 2200
04.03.2018 Эвентова Маргарита Геннадиевна 2 06.03.2018 2 4400
04.03.2018 Каширин Дементий Фомевич 2 04.03.2018 1 5600
04.03.2018 Торсунова Регина Вадимовна 4 04.03.2018 1 6400
04.03.2018 Каштанов Иосиф Левович 2 04.03.2018 3 5600
04.03.2018 Копылова Эмилия Ростиславовна 2 06.03.2018 2 2300
04.03.2018 Кузинков Геннадий Тарасович 1 05.03.2018 3 6000
04.03.2018 Харзина Доминика Ярославовна 2 05.03.2018 3 2400
05.03.2018 Трапезникова Ирина Тимофеевна 1 07.03.2018 2 2400
05.03.2018 Кондракова Бронислава Евгениевна 1 07.03.2018 2 6700
05.03.2018 Маслов Эдуард Артемиевич 1 05.03.2018 1 6800
05.03.2018 Рязанов Антип Сигизмундович 3 05.03.2018 3 4300
05.03.2018 Лекомцева Фаина Федотовна 2 05.03.2018 3 6600
05.03.2018 Вихров Вацлав Ростиславович 1 06.03.2018 2 5100
05.03.2018 Ассоров Артём Артемович 2 07.03.2018 2 5600
05.03.2018 Циолковский Модест Пахомович 1 06.03.2018 1 3300
05.03.2018 Николаевский Тимур Глебович 3 07.03.2018 3 5200
05.03.2018 Верица Захар Владиславович 1 05.03.2018 3 3200
05.03.2018 Шулёва Марианна Германовна 2 05.03.2018 1 4700
05.03.2018 Зимин Еремей Владимирович 1 06.03.2018 4 3600
05.03.2018 Васильев Владилен Михеевич 2 07.03.2018 3 6600
05.03.2018 Кацен Самсон Олегович 2 05.03.2018 1 5100
05.03.2018 Панкин Аркадий Аполлинариевич 1 07.03.2018 1 6500
05.03.2018 Ульянова Роза Владленовна 3 05.03.2018 1 3000
05.03.2018 Осин Тимофей Игоревич 2 05.03.2018 3 6100
05.03.2018 Апевалов Федор Аникитевич 2 06.03.2018 2 6500
05.03.2018 Бойков Игнатий Леонович 1 06.03.2018 2 5500
05.03.2018 Янушко Милена Тарасовна 1 07.03.2018 2 4400
06.03.2018 Долженко Фока Валерьянович 1 08.03.2018 1 2100
06.03.2018 Клычева Евгения Прокловна 5 08.03.2018 1 4200
06.03.2018 Акимова Янина Леонидовна 2 06.03.2018 2 2100
06.03.2018 Ширманов Демьян Савелиевич 1 06.03.2018 2 6700
06.03.2018 Анисимов Евстигней Эрнстович 2 07.03.2018 1 6400

06.03.2018 Тюрин Изяслав Модестович 2 07.03.2018 3 2800
06.03.2018 Тевосов Марк Титович 1 07.03.2018 2 2700
06.03.2018 Тальдаев Онисим Евлампиевич 4 08.03.2018 2 6000
06.03.2018 Колесников Агап Родионович 1 07.03.2018 2 3700
06.03.2018 Ермушина Зинаида Игнатъевна 2 06.03.2018 2 3400
06.03.2018 Чупова Ярослава Ульяновна 1 07.03.2018 2 2100
06.03.2018 Черкасов Артур Герасимович 2 07.03.2018 3 3700
06.03.2018 Якутина Евдокия Прохоровна 1 08.03.2018 3 3800
06.03.2018 Поварницын Соломон Андронович 1 06.03.2018 1 2600
06.03.2018 Бранта Стела Богдановна 2 08.03.2018 1 3300
06.03.2018 Беломестных Измаил Ипполитович 2 06.03.2018 1 4100
06.03.2018 Яковиченко Моисей Прохорович 3 08.03.2018 3 5800
06.03.2018 Яцков Анатолий Арсениевич 2 08.03.2018 2 2600
06.03.2018 Грызунов Фадей Данилевич 1 07.03.2018 3 2200
06.03.2018 Конаков Карп Иосифович 2 07.03.2018 3 5500
06.03.2018 Крючкова Инна Максимильяновна 2 06.03.2018 4 5300
06.03.2018 Суркова Каролина Иларионовна 3 06.03.2018 3 6000
07.03.2018 Лапухов Всеволод Никитевич 1 07.03.2018 1 6600
07.03.2018 Громько Вадим Венедиктович 1 07.03.2018 3 6800
07.03.2018 Галицкова Ариадна Павловна 1 09.03.2018 1 4000
07.03.2018 Койчев Платон Даниилович 2 07.03.2018 1 4500
07.03.2018 Соколов Евграф Изяславович 1 08.03.2018 1 2100
07.03.2018 Наполов Емельян Епифанович 1 08.03.2018 3 3600
07.03.2018 Митин Назар Тихонович 1 07.03.2018 2 6500
07.03.2018 Саянович Владлен Иванович 3 09.03.2018 3 4800
07.03.2018 Котенко Леонид Капитонович 2 09.03.2018 2 6500
07.03.2018 Савельев Святослав Феликсович 1 07.03.2018 1 3700
07.03.2018 Александрина Ксения Эрнестовна 3 08.03.2018 2 6500
07.03.2018 Севостьянов Мефодий Ерофеевич 2 09.03.2018 1 2300
07.03.2018 Мирсиянов Капитон Анатолиевич 2 08.03.2018 2 2300
07.03.2018 Жиглова Людмила Виталиевна 2 08.03.2018 3 4000
07.03.2018 Экель Инга Мефодиевна 2 09.03.2018 2 2600
07.03.2018 Железнов Всеслав Валериевич 1 09.03.2018 1 5300
07.03.2018 Золотухина Софья Назаровна 2 09.03.2018 3 3800
07.03.2018 Игонин Севастьян Елисеевич 5 08.03.2018 1 4200
07.03.2018 Капитанчук Михаил Самуилович 2 09.03.2018 1 4000
07.03.2018 Олейников Сократ Вадимович 1 09.03.2018 2 3400
07.03.2018 Чесноков Поликарп Ермолаевич 1 07.03.2018 2 2700
07.03.2018 Тимиряев Сигизмунд Самсонович 4 08.03.2018 3 8000
08.03.2018 Полевщикова Тамара Давидовна 2 10.03.2018 3 3000
08.03.2018 Раскатов Макар Панкратиевич 2 08.03.2018 3 6700
08.03.2018 Корниец Агния Елизаровна 1 10.03.2018 2 2300
08.03.2018 Букавицкий Василий Давидович 2 09.03.2018 1 4400
08.03.2018 Дробышева Ефросиния Лаврентиевна 2 08.03.2018 3 4500
08.03.2018 Анасенко Владлена Харитоновна 1 10.03.2018 2 2700
08.03.2018 Бабанова Юлия Рубеновна 1 10.03.2018 3 2600
08.03.2018 Грушецкий Владислав Зиновиевич 3 10.03.2018 1 6000
08.03.2018 Колвашев Агафон Миронович 1 09.03.2018 5 2300

08.03.2018 Гурина Мирослава Моисеевна 2 09.03.2018 1 6800
 08.03.2018 Горностаева Кира Игнatieвна 1 10.03.2018 3 5000
 09.03.2018 Тепляков Тихон Леонидович 1 10.03.2018 2 6000
 09.03.2018 Якимова Валентина Макаровна 2 09.03.2018 1 3300
 09.03.2018 Русаков Филимон Ульянович 3 11.03.2018 3 2300
 09.03.2018 Куделькин Ярослав Прокофиевич 2 09.03.2018 3 6200
 09.03.2018 Ясеневский Евсей Викентиевич 1 11.03.2018 2 5700
 09.03.2018 Шигаева Ираида Прокофьевна 2 09.03.2018 2 2400
 09.03.2018 Абросимов Панкратий Егорович 1 11.03.2018 2 2500
 09.03.2018 Семянин Прокофий Якубович 2 10.03.2018 3 4500
 09.03.2018 Янцева Нона Константиновна 3 10.03.2018 1 6800
 09.03.2018 Игнаткович Любава Георгиевна 1 09.03.2018 1 4600
 09.03.2018 Яфракова Светлана Тимуровна 2 11.03.2018 3 6800
 09.03.2018 Саламатова Нина Алексеевна 1 10.03.2018 3 5600
 09.03.2018 Вихрова Лариса Матвеевна 2 10.03.2018 2 4700
 09.03.2018 Валеева Яна Несторовна 2 10.03.2018 2 2000
 09.03.2018 Яблоков Данила Куприянович 2 11.03.2018 3 3200
 09.03.2018 Кулик Инесса Мечиславовна 2 09.03.2018 1 5000
 10.03.2018 Толбаева Алина Юлиевна 3 12.03.2018 4 5000
 10.03.2018 Карташëв Чеслав Сергеевич 1 12.03.2018 2 2400
 10.03.2018 Голубев Ульянов Игнatieвич 1 11.03.2018 3 6300
 10.03.2018 Икашев Валерьян Захарович 1 10.03.2018 1 6400
 10.03.2018 Сомов Кондратий Всеволодович 1 11.03.2018 1 4600
 10.03.2018 Витвинский Станислав Мечиславович 1 12.03.2018 2 2100
 10.03.2018 Тарасова Алла Юрьевна 5 12.03.2018 2 3100
 10.03.2018 Низовцев Алексей Федосиевич 1 12.03.2018 3 2700
 10.03.2018 Ситников Эрнст Проклович 1 12.03.2018 1 5300
 10.03.2018 Толбоев Кондрат Агапович 2 11.03.2018 1 4000
 10.03.2018 Костикова Диана Антоновна 2 10.03.2018 1 6000
 10.03.2018 Терезулов Лавр Фролович 1 10.03.2018 1 4400
 10.03.2018 Ямалтдинова Агафья Романовна 4 11.03.2018 3 4800
 10.03.2018 Щепкин Рюрик Никонович 1 10.03.2018 1 5500
 10.03.2018 Цехановецкая Лилия Евграфовна 1 12.03.2018 1 5200
 10.03.2018 Федченков Феоктист Эдуардович 1 10.03.2018 2 6100
 10.03.2018 Михалицина Вера Семеновна 3 10.03.2018 7 5100
 10.03.2018 Коротченко Семен Никанорович 2 11.03.2018 2 3400
 10.03.2018 Чупрова Василиса Трофимовна 2 12.03.2018 2 2600
 10.03.2018 Сëмина Арина Владиславовна 2 11.03.2018 1 5500
 10.03.2018 Чиркаш Нестор Михайлович 3 11.03.2018 3 2100
 10.03.2018 Клепахова Ангелина Архиповна 2 10.03.2018 2 3500
 10.03.2018 Иванников Владимир Климентович 2 10.03.2018 3 5400

Программа должна хранить информацию о фактической занятости всех номеров и о их занятости в ближайшие дни.

При бронировании номеров программа должна автоматически формировать сообщение о бронировании. Стратегия обработки заявок строится так, чтобы добиться максимальной занятости гостиницы с целью увеличения ее прибыли. Для этого программа «распоряжается» номерным фондом: в частности, при нехватке нужных номеров должны быть использованы пустующие номера большей комфортности (по меньшей цене, со

скидкой 30%). Например, при нехватке одноместных номеров можно поселить одного человека в двухместный номер (за 70% его стоимости), если это позволяет платежеспособность клиента. Кроме этого, программа назначает тип питания постояльцам, исходя из стратегии максимальной выгоды и платежеспособности клиентов.

Программа должна использовать структуру «Вариант размещения». Эта структура должен позволять производить работу с каждым вариантом размещения, и выбирать оптимальный вариант исходя из стратегии, описанной выше.

Цель моделирования – изучение стратегий обработки заявок на заселение. В ходе моделирования программа должна предоставлять отчет за каждый рабочий день, включающий:

- количество занятых номеров;
- количество свободных номеров;
- процент загрузки отдельных категорий номеров;
- процент загрузки гостиницы в целом;
- полученный доход за день;
- упущенный доход (в случаях отказа в бронировании или отказа клиента).

Примечание:

Рассмотрим одну из строк бронирования:

01.03.2018 Никитаев Евгений Иннокентиевич 1 02.03.2018 3 5400

Дата бронирования – 01.03.2018

Будет заселен один человек на три дня: 02.03.2018, 03.03.2018, 04.03.2018

Максимальный допустимый расход 5400 руб. на одного человека.

При всех свободных номерах программа должна забронировать номер 7 – одноместный апартамент и назначить полупансион питания на общую сумму: 5350 руб. в сутки.

При занятости одноместных номеров, программа должна попытаться забронировать номер 16 - двухместный полулюкс стандарт улучшенный без питания на общую сумму 5376 руб. в сутки.

При занятости и этого номера, производится оптимальный поиск других вариантов. Если не удастся найти ни один подходящий вариант – программа выводит сообщение в отказе от бронирования и считает упущенный доход 5400 руб. * 3 * 1 = 16200 руб.

ВАЖНО!

На любое сформированное предложение о бронировании номера, клиент может отказаться с вероятностью 25%, что так же будет считаться упущенным доходом.

Более чем шестиместные номера в гостинице не предусмотрены.

Заселение большего количества человек в номер, рассчитанный на меньшее количество человек – запрещено.

В моделировании рассматривается период в пределах не более одного календарного месяца.

Пример результатов моделирования

Результаты моделирования должны выводиться на экран. Ниже приведен пример моделирования за один день.

Поступила заявка на бронирование:

01.03.2018 Жиренкова Надежда Евдокимовна 1 01.03.2018 3 4400

Найден:

*номер 1 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. полупансион
стоимость 3900.00 руб./сутки*

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Мясников Виссарион Яковович 1 01.03.2018 1 3500

Найден:

*номер 6 одноместный стандарт_улучшенный рассчитан на 1 чел. фактически 1 чел.
без питания стоимость 3480.00 руб./сутки*

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Бузинская Альбина Кирилловна 1 03.03.2018 1 2200

Предложений по данному запросу нет. В бронировании отказано.

Поступила заявка на бронирование:

01.03.2018 Канадов Самуил Севастьянович 2 01.03.2018 1 6600

Найден:

номер 23 люкс апартамент рассчитан на 2 чел. фактически 2 чел. завтрак стоимость 12860.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Грядкин Порфирий Кондратиевич 4 03.03.2018 2 8100

Найден:

номер 15 люкс апартамент рассчитан на 6 чел. фактически 4 чел. полупансион стоимость 29830.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Нефёдова Марфа Потаповна 3 03.03.2018 1 2200

Предложений по данному запросу нет. В бронировании отказано.

Поступила заявка на бронирование:

01.03.2018 Архипов Никон Афанасиевич 1 01.03.2018 2 3400

Найден:

номер 5 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. завтрак стоимость 3180.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Ежова Лада Феликсовна 1 01.03.2018 2 5800

Найден:

номер 11 одноместный стандарт_улучшенный рассчитан на 1 чел. фактически 1 чел. полупансион стоимость 4480.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Никитаев Евгений Иннокентиевич 1 01.03.2018 3 3400

Найден:

номер 7 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. завтрак стоимость 3180.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Шайн Лев Александрович 5 05.03.2018 3 46400

Найден:

номер 14 люкс апартамент рассчитан на 5 чел. фактически 5 чел. полупансион стоимость 35750.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Архаткин Дмитрий Юриевич 2 01.03.2018 3 2400

Найден:

номер 8 двухместный стандарт рассчитан на 2 чел. фактически 2 чел. без питания
стоимость 4600.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Водолеев Николай Давыдович 1 01.03.2018 2 3000

Найден:

номер 10 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. без питания
стоимость 2900.00 руб./сутки

Клиент отказался от варианта.

Поступила заявка на бронирование:

01.03.2018 Сильвестров Евдоким Маркович 1 01.03.2018 2 3100

Найден:

номер 10 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. без питания стоимость 2900.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Калугин Богдан Карлович 1 01.03.2018 2 3700

Найден:

номер 21 одноместный стандарт_улучшенный рассчитан на 1 чел. фактически 1 чел. без питания стоимость 3480.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Вершинин Борислав Евграфович 1 01.03.2018 3 3700

Найден:

номер 22 одноместный стандарт_улучшенный рассчитан на 1 чел. фактически 1 чел. без питания стоимость 3480.00 руб./сутки

Клиент отказался от варианта.

Поступила заявка на бронирование:

01.03.2018 Туржанская Марта Казимировна 1 01.03.2018 3 3700

Найден:

*номер 22 одноместный стандарт_улучшенный рассчитан на 1 чел. фактически 1 чел.
без питания стоимость 3480.00 руб./сутки*

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Аксёнов Лаврентий Семенович 3 03.03.2018 2 4600

Найден:

*номер 4 полулюкс стандарт_улучшенный рассчитан на 3 чел. фактически 3 чел. заб
трак стоимость 12360.00 руб./сутки*

Клиент отказался от варианта.

Поступила заявка на бронирование:

01.03.2018 Капшукова Елена Валентиновна 1 01.03.2018 3 3600

Найден:

номер 18 одноместный стандарт рассчитан на 1 чел. фактически 1 чел. завтрак стоимость 3180.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Грызлов Аскольд Никифорович 1 01.03.2018 2 3300

Найден:

номер 12 двухместный стандарт рассчитан на 2 чел. фактически 1 чел. без питания стоимость 3220.00 руб./сутки

Клиент отказался от варианта.

Поступила заявка на бронирование:

01.03.2018 Большов Эммануил Архипович 2 01.03.2018 4 6000

Найден:

номер 16 полулюкс стандарт_улучшенный рассчитан на 2 чел. фактически 2 чел. по лупансион стоимость 9680.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Жернаков Родион Гордеевич 2 02.03.2018 2 5600

Найден:

номер 24 полулюкс стандарт_улучшенный рассчитан на 2 чел. фактически 2 чел. по лупансион стоимость 9680.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Ярошевич Антонина Сократовна 2 02.03.2018 1 4300

Найден:

номер 2 двухместный стандарт_улучшенный рассчитан на 2 чел. фактически 2 чел. полупансион стоимость 7520.00 руб./сутки

Клиент согласен. Номер забронирован.

Поступила заявка на бронирование:

01.03.2018 Юдицкий Осип Адамович 1 01.03.2018 1 3500

Найден:

номер 12 двухместный стандарт рассчитан на 2 чел. фактически 1 чел. завтрак стоимость 3500.00 руб./сутки

Клиент согласен. Номер забронирован.

=====
=====

Итого за 01.03.2018

Количество занятых номеров: 13

Количество свободных номеров: 11

Занятость по категориям:

Одноместных: 9 из 9

Двухместных: 2 из 6

Полулюкс: 1 из 4

Люкс: 1 из 5

Процент загрузки гостиницы: 54.17 %

Доход за день: 883250.00 руб.

Упущенный доход: 60100.00 руб.

Пример заданий контрольной работы:

Контрольная работа № 7. СПИСКИ И КОРТЕЖИ

Задание 1.

С клавиатуры, в одной строке вводятся 100 целых чисел через пробел. Программа должна из этих чисел, находить значение, которое наиболее близко к их среднему арифметическому. Если такое число не одно, то вывести наибольшее из них.

Входные данные:	Выходные данные:
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87 90 93 96 99 102 105 108 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162 165 168 171 174 177 180 183 186 189 192 195 198 201 204 207 210 213 216 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270 273 276 279 282 285 288 291 294 297 300	153

Задание 2.

N кеглей выставили в один ряд, занумеровав их слева направо числами от 1 до N . Затем по этому ряду бросили K шаров, при этом i -й шар сбил все кегли с номерами от l_i до r_i включительно. Определите, какие кегли остались стоять на месте.

Программа получает на вход количество кеглей N и количество бросков K . Далее идет K пар чисел l_i, r_i , при этом $1 \leq l_i \leq r_i \leq N$.

Программа должна вывести последовательность из N символов, где j -й символ есть "1", если j -я кегля осталась стоять, или ".", если j -я кегля была сбита.

Входные данные:	Выходные данные:
10 3 8 10 2 5 3 6	1.....1...

Задание 3.

С клавиатуры, в одной строке вводятся некоторое количество целых чисел, разделенных пробелами. Программа должна определять количество участков в этой последовательности, на которых элементы монотонно возрастают (каждое следующее число больше предыдущего).

Входные данные:	Выходные данные:
137 168 382 184 187 32 396 207 323 193 303 375 14 353 28 399 62 134 399 196 0 146 364 371 197 34 9 115 265 374 397 340 235 126 42 2 9 53 123 201 276 337 378 398 396 376 342 298 249 198 150 105 68 38 17 4 0 2 12 27 47 70 96 124 153 181 210 237 263 287 309 329 346 361 373 383 390 396 399 399 398 396 391 385 378 370 360 350 339 328 315 303 290 277 264 251 237 224 212 199	12

Задание 4.

Известно, что на доске 8×8 можно расставить 8 ферзей так, чтобы они не били друг друга. Вам дана расстановка 8 ферзей на доске, определите, есть ли среди них пара бьющих друг друга.

Программа получает на вход восемь пар чисел, каждое число от 1 до 8 — координаты 8 ферзей. Если ферзи не бьют друг друга, выведите слово NO, иначе выведите YES.

Входные данные:	Выходные данные:
1 7 2 4 3 2 4 8 5 6 6 1 7 3 8 5	NO

Пример экзаменационной работы:

ЭКЗАМЕНАЦИОННАЯ РАБОТА

Тестирование

1. Из фрагмента кода, представленного ниже, выберите ключи.

```
d = {"apple":40, "pear":45}
```

- a. "apple" и "pear"
- b. d = (40:"apple", 45:"pear")
- c. "apple", 40, 45, и "pear"
- d. 40 и 45

2. Каков результат работы предложенного ниже фрагмента программного кода?
def f (n):

```
    if n == 4:  
        return n  
    else:  
        return 2*f(n+1)
```

```
print(f(2))
```

- a. заикливание
- b. 4
- c. 8
- d. 16

3. Каков результат работы предложенного ниже фрагмента кода?

```
z=set('abc$de')  
print('a' in z)
```

- a. None
- b. нет правильного ответа
- c. False
- d. True

4. Каков результат работы предложенного ниже фрагмента программного кода?

```
def a (n) :  
    if n == 0:  
        return 0  
    elif n == 1 :  
        return 1  
    else :  
        return a (n- 1) + a(n- 2)
```

```
for i in range (0, 4) :  
    print (a(i) , end = "")
```

- a. 0 1 1 2 3

- b. заикливание
- c. 0 1 2 3
- d. 0 1 1 2

5. Каков результат работы предложенного ниже фрагмента программного кода?

```
def foo (fname, val) :  
    print (fname(val))  
foo(max, [1, 2, 3])  
foo(min, [1, 2, 3])
```

- a. ошибка
- b. 1 3
- c. 3 1
- d. нет правильного ответа

6. Каков результат работы предложенного ниже фрагмента программного кода?

```
>>>class demo():  
    def __repr__(self):  
        return '__repr__ built-in function called'  
    def __str__(self):  
        return '__str__ built-in function called'
```

```
>>>s=demo()  
>>>print(s)
```

- a. вызывается метод __repr__
- b. вызывается метод __str__
- c. ничего не выводится
- d. ошибка

7. Каков результат работы предложенного ниже фрагмента программного кода?

```
count={ }  
count[(1,2,4)] = 5  
count[(4,2,1)] = 7  
count[(1,2)] = 6  
count[(4,2,1)] = 2  
tot = 0  
for i in count:  
    tot=tot+count[i]  
print(len(count)+tot)
```

- a. 25
- b. 16
- c. 17
- d. нет правильного ответа

8. Каков результат работы предложенного ниже фрагмента программного кода?

```
class Foo:  
    def __init__(self):  
        self.x = 10  
        self.__y = 10  
    def display(self):  
        return self.__y
```

```
bar = Foo()  
print(bar.x)
```

- a. ошибка, у является защищенным атрибутом, а метод display() его возвращает
- b. ошибка, недопустимое имя атрибута __у
- c. 10
- d. ошибка, отсутствует метод, возвращающий self.x

9. Содержит ли предложенный ниже фрагмент программного кода ошибку?

```
class A:
    def __init__(self,b):
        self.b=b
```

```
    def display(self):
        print(self.b)
```

```
obj=A("Hello")
del obj
```

- a. да
- b. нет

10. Содержит ли приведенная ниже строка кода ошибку?

```
s={2, 3, 4, [5, 6]}
```

- a. нет
- b. да

11. Каков результат работы предложенного ниже фрагмента программного кода?

```
class A:
    def __init__(self):
        self.x = 0
```

```
class B(A):
    def __init__(self):
        A.__init__(self)
        self.y = 1
```

```
def main():
    b = B()
    print(b.x,b.y)
main()
```

- a. ошибка, неверно вызван метод
- b. 0 0
- c. 0 1
- d. ошибка, поскольку класс B наследует A, но атрибут x не унаследован

12. Каков результат работы предложенного ниже фрагмента программного кода?

```
a={ 1:"A",2:"B",3:"C"}
print(a.get(1,4))
```

- a. 1

- b. нет правильного ответа
- c. 4
- d. A

13. Дано $a = \{5,6,7\}$. Что произойдет, после выполнения $a.add(5)$?

- a. $a = \{5,5,6,7\}$
- b. $a = \{5,6,7\}$
- c. Ошибка, поскольку недопустимо использовать метод `add` для множеств
- d. Ошибка из-за того, что 5 уже существует в множестве

14. Каков результат работы предложенного ниже фрагмента программного кода?

```
def f1():  
    global x  
    x+=1  
    print(x)  
x=12  
print("x")
```

- a. ошибка
- b. x
- c. 13
- d. x
- e. 13

15. Какой метод перегружает оператор `==` ?

- a. нет правильного ответа
- b. `__eq__()`
- c. `__equ__()`
- d. `__isequal__()`

16. Разница между функциями `discard` и `remove` заключается в том, что:

- a. `discard` удаляет последний элемент множества, тогда как `remove` удаляет первый элемент множества
- b. `discard` выдает ошибку, если указанный элемент отсутствует в множестве, тогда как `remove` не выдает ошибку в случае отсутствия указанного элемента
- c. `remove` удаляет последний элемент множества, тогда как `discard` удаляет первый элемент
- d. `remove` выдает ошибку, если указанный элемент отсутствует в множестве, тогда как `discard` не выдает ошибку в случае отсутствия указанного элемента

17. Каков результат работы предложенного ниже фрагмента программного кода?

```
class A:  
    def __str__(self):  
        return '1'
```

```
class B(A):  
    def __init__(self):  
        super().__init__()
```

```
class C(B):  
    def __init__(self):
```

```
super().__init__()
```

```
def main():  
    obj1 = B()  
    obj2 = A()  
    obj3 = C()  
    print(obj1, obj2, obj3)  
main()
```

- a. 1 1 1
- b. вызов исключения
- c. 1 2 3
- d. '1' '1' '1'

18. Каков результат работы предложенного ниже фрагмента программного кода?

```
class A:  
    def __init__(self, x):  
        self.x = x  
    def count(self, x):  
        self.x = self.x + 1
```

```
class B(A):  
    def __init__(self, y=0):  
        A.__init__(self, 3)  
        self.y = y  
    def count(self):  
        self.y += 1
```

```
def main():  
    obj = B()  
    obj.count()  
    print(obj.x, obj.y)  
main()
```

- a. 3 0
- b. нет правильного ответа
- c. 0 1
- d. 3 1

19. Какое значение примет f после следующего присваивания?

```
f = frozenset(set([5, 6, 7]))
```

- a. {5,6,7}
- b. ошибка синтаксиса
- c. frozenset ({5,6,7})
- d. ошибка, невозможность преобразовать множество в frozenset

20. Каков результат работы предложенного ниже фрагмента программного кода?

```
a = {}  
a[1] = 1  
a['1'] = 2  
a[1]=a[1]+1  
count = 0
```

```
for i in a:
    count += a[i]
print(count)
```

- a. нет правильного ответа
- b. 4
- c. 1
- d. 2

Задачи

1. Дано множество натуральных чисел A и N множеств натуральных чисел B_1, B_2, \dots, B_N . Определите, для скольких множеств B_i , множество A содержит все элементы B_i , за исключением одного элемента. Программа должна вывести количество таких множеств.

Входные данные:

В первой строке - элементы множества A через пробел

Во второй строке - число N

В следующих N строках - элементы множеств B_i через пробел.

Выходные данные:

Одно число - количество множеств B_i , для которых A содержит все, кроме одного элемента.

Пример:

Входные данные	Выходные данные
1 2 3 4 5 6 3 1 2 3 2 3 7 8 9 2	1

Примечание: из трех множеств, только одно множество $\{2, 3, 7\}$ удовлетворяет условию.

2. На вход из стандартного потока ввода программа получает список словарей в формате JSON. Каждый словарь списка имеет строковые ключи и кортежи целых чисел в качестве значений. Программа должна вывести в стандартный поток вывода словарь, содержащий все ключи исходного списка словарей. Значения по ключам должны быть просуммированы. Ключи нового словаря должны иметь порядок их вхождения в исходном списке словарей.

Примечание:

Для получения объекта a из стандартного потока в формате JSON можно использовать:

```
import json
```

```
a = json.loads(input())
```

Входные данные:

Список словарей в JSON формате.

Выходные данные:

Словарь в формате JSON.

Пример:

Входные данные	Выходные данные
[{"key1": [0, 4, 1], "key2": [3, -8, 14, 5]}, {"key1": [1, 7], "key2": [9, -13]}, {"key2": [1, -1, 1], "key3": [3]}]	{'key1': 13, 'key2': 11, 'key3': 3}

3. Напишите рекурсивную функцию, определяющее второе минимальное значение, т.е. если последовательность упорядочить по не убыванию, то это должен быть второй элемент последовательности. Не допускается использование итеративных алгоритмов и стандартных функций, таких как `min()`, `sort()`, `sorted()`, использование более одного параметра функции. Функция не должна использовать вспомогательных функций.

Входные данные:

Список вводится из стандартного потока в формате JSON.

Функция на вход получает единственный параметр `x` - список целых чисел.

Выходные данные:

Функция возвращает одно целое число - значение второго минимума.

Программа выводит на экран это число.

В случае, если такое значение определить невозможно, программа должна выводить `None`.

Пример:

Входные данные	Выходные данные
[3, 8, 6, 4, 20, 12]	4

**Лист актуализации рабочей программы дисциплины
«Программирование»**

№	Характеристика внесенных изменений (с указанием пунктов документа)	Дата и № протокола Ученого совета экономического факультета	Подпись ответственного