Высокоуровневые методы программирования

Методические указания по выполнению курсовой работы

Язык программирования Python

**Содержание**

[1 Цель и задачи дисциплины 3](#_Toc57139331)

[2 Перечень планируемых результатов при выполнеии курсовой работы, соотнесенных с планируемыми результатами освоения образовательной программы 4](#_Toc57139332)

[3 Задания для курсовой работы 5](#_Toc57139333)

[4 Порядок выполнения курсовой работы 21](#_Toc57139334)

[5 Сроки выполнения отдельных этапов 22](#_Toc57139335)

[6 Содержание пояснительной записки 27](#_Toc57139336)

[7 Правила оформления пояснительной записки 29](#_Toc57139337)

[7.1 Оформление текстового и графического материала 29](#_Toc57139338)

[7.2 Оформление рисунков, схем алгоритмов, таблиц и формул 29](#_Toc57139339)

[7.3 Оформление текстов программ 31](#_Toc57139340)

[7.4 Оформление выводов 32](#_Toc57139341)

[7.5 Оформление списка литературы 32](#_Toc57139342)

[7.6 Оформление приложений 32](#_Toc57139343)

[8 КРИТЕРИИ ОЦЕНКИ ДЛЯ КУРСОВОЙ РАБОТЫ 33](#_Toc57139344)

[Приложение 34](#_Toc57139345)

# Цель и задачи дисциплины

**Цель:**

изучение теоретических и практических основ построения и организации функционирования программных комплексов, их программного обеспечения и способов эффективного применения современных высокоуровневых методов программирования

**Задачи:**

- получение теоретических и практических навыков решения прикладных информационных задач с использованием методик программирования высокого уровня

- разработка программных продуктов (структурной и функциональной схем программного обеспечения, структур данных, алгоритмов и реализующих их программ, стратегии тестирования и тестовых данных и т.п.);

# Перечень планируемых результатов при выполнеии курсовой работы, соотнесенных с планируемыми результатами освоения образовательной программы

Компетенции обучающегося, формируемые в результате освоения дисциплины и выполнения курсовой работы

Процесс изучения дисциплины направлен на формирование у обучающихся следующих компетенций (результатов освоения образовательной программы):

|  |  |
| --- | --- |
| Коды компетенций | Содержание компетенций |
| ПК-1 | Способность разработки прикладного программного обеспечения, автоматизации работы с базами данных и документами, программирования бизнес-логики приложений, интеграции разнородных данных |

Взаимосвязь планируемых результатов выполнения курсовой работы по дисциплине «Высокоуровневые методы программирования» с формируемыми компетенциями ОПОП

|  |  |
| --- | --- |
| Коды компетенций ОПОП | Индикаторы |
| ПК-1 | ПК 1.1. Знает технологии высокоуровневого программирования прикладного программного обеспечения |
| ПК 1.2. Умеет разрабатывать и тестировать прикладное программное обеспечение |
| ПК 1.3. Владеет навыками автоматизации решения типовых задач |

**3. Место дисциплины в структуре образовательной программы**

Дисциплина «Высокоуровневые методы программирования» относится к части, формируемой участниками образовательных отношений учебного плана ОПОП ВО по направлению подготовки 09.03.03 «Прикладная информатика». Общая трудоемкость (объем) дисциплины составляет 8 зач. ед. 288 часов.

# Задания для курсовой работы

Тема курсовой работы по дисциплине «Высокоуровневые методы программирования»:

**Разработка программного продукта для решения прикладных задач**

Задания подобраны таким образом, чтобы при выполнении работы студенты могли приобрести практические навыки разработки программных продуктов для решения прикладных задача среднего уровня сложности. Среда разработки PyCharm, язык программирования - Python. Рекомендуется использование наиболее распространенных библиотек и модулей стандартной библиотеки. Использование нестандартных библиотек возможно только после согласования с руководителем курсовой работы.

**Перечень заданий для курсовой работы**

Задание № 1

|  |  |
| --- | --- |
| Работа с наборами данных | БЕЗ ГРАФИЧЕСКОГО ИНТЕРФЕЙСА (~~GUI~~) |
| формулировка | Во внешнем файле resourse\_1.txt дан текст. Выведите все слова, встречающиеся в тексте, по одному на каждую строку, через пробел укажите количество повторений. Слова должны быть отсортированы по убыванию их количества появления в тексте, а при одинаковой частоте появления — в лексикографическом порядке. Вывод должен осуществляться в текстовый файл result\_1.txt. При необходимости можно продублировать вывод в консоль. |
| Методические указания | После того, как вы создадите словарь всех слов, необходимо отсортировать его по частоте встречаемости слова. Желаемого можно добиться, если создать список, элементами которого будут кортежи из двух элементов: частота встречаемости слова и само слово. Например, [(2, 'hi'), (1, 'what'), (3, 'is')]. Тогда стандартная сортировка будет сортировать список кортежей, при этом кортежи сравниваются по первому элементу, а если они равны — то по второму. Знаки препинания не должны учитываться.  **Программу сохранить под именем exercise\_1.py** |
| Входные данные | Преподаватель вводит текст в текстовый файл resourse\_1.txt (5-6 абзацев) и сохраняет его.  hi  hi  what is your name  my name is bond  james bond  my name is damme  van damme  claude van damme  jean claude van damme |
| Выходные данные | Выведите ответ на задание № 1 в текстовый файл result\_1.txt.  Damme 4  Is 3  Name 3  Van 3  Bond 2  claude 2  hi 2  my 2  james 1  jean 1  what 1 |

Задание № 2

|  |  |
| --- | --- |
| Разработка экспертной системы | С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ (GUI) |
| формулировка | Некоторый банк хочет внедрить систему управления счетами клиентов, поддерживающую следующие операции:   1. Пополнение счета клиента. 2. Снятие денег со счета. 3. Запрос остатка средств на счете. 4. Перевод денег между счетами клиентов. 5. Начисление процентов всем клиентам. |
| Методические указания | Необходимо реализовать такую систему. Первоначально у банка 1 клиент. Клиент(ы) банка идентифицируются именами (уникальная строка, не содержащая пробелов). Вам необходимо задать в качестве имени клиента – свою фамилию на английском языке с большой буквы. На вашу фамилию должен быть открыт счет с суммой равной вашему ID.  Ivanov 70121903  В отдельном поле должна быть предусмотрена возможность ввода простых команд, которые поддерживают следующие операции:   |  |  | | --- | --- | | DEPOSIT name sum | Зачислить сумму sum на счет клиента name. Если клиента нет, то он создается и на него заводится счет с указанной суммой. | | WITHDRAW name sum | Снять сумму sum со счета клиента name. Если клиента, то счет создается. Баланс при выполнении такой операции у вновь созданного клиента должен быть отрицательный. | | BALANCE name | Узнать остаток средств на счету клиента name. Для каждого запроса BALANCE программа должна вывести остаток на счету данного клиента. Если же у клиента с запрашиваемым именем не открыт счет в банке, выводится сообщение «NO CLIENT». Если пользователь не указал имя клиента – то выводится баланс всех существующих клиентов. | | TRANSFER name1 name2 sum | Перевести сумму sum со счета клиента name1 на счет клиента name2. Если у какого-либо клиента, то он заводится в системе и ему создается счет с переведенной суммой. | | INCOME p | Начислить всем клиентам, у которых открыты счета, p% от суммы счета. Проценты начисляются только клиентам с положительным остатком на счету, если у клиента остаток отрицательный, то его счет не меняется. После начисления процентов сумма на счету остается целой, то есть начисляется только целое число денежных единиц. Дробная часть начисленных процентов отбрасывается. |   ПРОГРАММА ДОЛЖНА ОБРАБАТЫВАТЬ ТЕКСТОВЫЕ КОМАНДЫ ИЗ ЛЕВОГО ПОЛЯ ТОЛЬКО ПОСЛЕ НАЖАТИЯ КНОПКИ «Calculate». То есть, пользователь СНАЧАЛА вводит желаемые команды, при этом каждая новая команда вводится с новой строки, а ПОТОМ нажимает на кнопку «Calculate». Результат должен быть выведен в поле справа.  Количество команд, которые может ввести пользователь за один раз – не более 20. При необходимости следует предусмотреть прокрутку в поле. Пользователю допускается вводить «пустые строки» - несколько раз нажимать на кнопку «enter». При нажатии клавиши ввода «enter», фокус не должен переходить на кнопку «Calculate».  Формат и внешний вид окна определяет разработчик. Для «очистки» левого и правого полей можно предусмотреть кнопку «Clear», при этом данные о ранее введенных клиентах не должны быть потеряны.  ВАЖНО:  1. Команды вводятся пользователем только большими буквами. Сама команда, имя клиента, суммы (числа) разделяются пробелами.  2. Предполагается, что пользователь такой системы грамотный и команды с аргументами вводит без ошибок в рамках их вышесформулированного синтаксиса.  3. Как только для несуществующего ранее клиента проводится операция **пополнения** (DEPOSIT), **снятия** (WITHDRAW) или **перевода денег** (TRANSFER), он вносится в систему, ему заводится счет с указанным балансом. Все дальнейшие операции проводятся только с этим счетом. Сумма на счету может быть как положительной, так и отрицательной, при этом всегда является целым числом.  **Программу сохранить под именем exercise\_2.py** |
| Входные данные | Преподаватель вводит текст в текстовый файл resourse\_2.txt (5-6 абзацев) и сохраняет его. |
| Выходные данные | Выведите ответ на задание № 1 в текстовый файл result\_2.txt.  Damme 4  Is 3  Name 3  Van 3  Bond 2  claude 2  hi 2  my 2  james 1  jean 1  what 1 |

Задание № 3

|  |  |
| --- | --- |
| Разработка аналитической системы | С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ (GUI) |
| формулировка | Разработать калькулятор со стандартным и расширенным функционалом.  Стандартный функционал   1. Арифметические действия + - \* /. 2. Возможность ввода отрицательного числа 3. Возведение в степень. 4. Извлечение квадратного корня. 5. Работа с памятью, состоящей из одной ячейки. 6. Должна быть кнопка сброса и кнопка «=» (равно).   Расширенный функционал   1. Наличие кнопки/меню перехода в расширенный режим 2. Возможность работы с несколькими ячейками памяти. Количество ячеек памяти выбирается согласно методическим указаниям. 3. Отображение последовательности математических операций и цифр в n-строчном «дисплее», с возможностью «прокрутки». Количество строк «дисплея» калькулятора выбирается согласно методическим указаниям. 4. Реализация «инженерных» функций расширенного режима. Конкретный перечень функций выбирается согласно методическим указаниям. |
| Методические указания | Необходимо разработать программу и GUI для реализации стандартных функций калькулятора. Примерный вид внешнего интерфейса представлен на рисунке.    Конкретная компоновка элементов интерфейса, внешний вил, цветовая палитра, максимальное количество отображаемых цифр, число отображаемых строк на «дисплее» калькулятора могут отличаться от представленного рисунка и должны согласовываться с руководителем курсовой работы.  Функционал «инженерного» режима работы калькулятора добавляется при нажатии на кнопку перехода в расширенный режим, либо выбора соответствующего пункта меню.  Расширенный режим работы определяется внешним видом «цифрового дисплея», количеством ячеек памяти (кнопки M+, M-, MS, MR, MS) и кнопками, отвечающими за дополнительные функции.  **Количество строк «цифрового дисплея»** должно определяться как последовательная сумма всех цифр ID студента. Суммирование отдельных цифр числа должно осуществляться до получения однозначного числа, состоящей из 1-й цифры. Данный процесс рекомендуется реализовать с помощью рекурсивной функции. Для данной функции должна быть предусмотрена специальная кнопка .  Например: обучающийся Иванов Иван Иванович, имеет ID 80121986  Сумма всех цифр равна:  8+0+1+2+1+9+8+6=35;  3+5=8  Следовательно, «цифровой дисплей» калькулятора должен иметь 8 строк.    Если после последовательного суммирования ID получилась цифра «1», например, для ID=82121986:  8+2+1+2+1+9+8+6=37;  3+7=10  1+0=1  В этом случае количество строк «цифрового дисплея» должно быть равно 10.  Таким образом, в зависимости от ID число «цифровых строк» дисплея калькулятора может варьироваться от 2-х до 10.  **Количество ячеек памяти** калькулятора должно определяться как последовательная сумма последних 3-х чисел ID. Суммирование отдельных цифр числа должно осуществляться до получения однозначного числа, состоящей из 1-й цифры. Данный процесс также рекомендуется реализовать с помощью рекурсивной функции.  Например: обучающийся Иванов Иван Иванович, имеет ID 80121986  Сумма последних трёх цифр равна:  9+8+6=23;  2+3=5  Следовательно, должно быть предусмотрено 5 ячеек памяти для работы с (M+, M-, MС, MR, MS).    Если после последовательного суммирования 3-х цифр ID получилась цифра «1», например, для ID=82121001:  0+0+1=1;  В этом случае количество ячеек памяти должно быть равно 2.  Таким образом, в зависимости от ID число ячеек памяти (и соответствующих кнопок, реализующий данный функционал) калькулятора может варьироваться от 2-х до 9.  **Дополнительные функции расширенного режима**, которые должны быть реализованы определяются в соответствие с таблицей по первой букве Фамилии обучающегося   |  |  |  | | --- | --- | --- | | Первая буква  фамилии | Наименование функции | Описание функции | | А | Pi, sin,tan,exp, n!, Frac | Число Пи, синус, тангенс, экспонента, факториал; отсекает целую часть, оставляет дробную | | Б | asin, acos, atg, log\_xy, n! | Арксинус, арккосинус, арктангенс, логарифм по основанию, факториал | | В | //, ctg, 10х, asin, acos | Целочисленное деление, котангенс, 10 в степени х, арксинус, арккосинус | | Г | F – E, y√x, lg10 | переключает ввод чисел в экспоненциальном представлении и обратно, y-ый корень числа x, где y обычно является положительным целым числом, десятичный логарифм | | Д | gcd(a, b); tanh, Ln, X3 | Возвращает наибольший общий делитель a и b; гиперболический тангенс, натуральный логарифм по основанию «e»; возвести в степень 3 | | Е | hypot(x, y), isqrt(), Sinh, Mod | функция вычисляет гипотенузу треугольника c катетами x и y; Возвращает целочисленный квадратный корень аргумента, округлённый вниз; гиперболический синус, вычислить остаток от деления одного числа на другое | | Ж | tanh, Ln, X3, Frac | гиперболический тангенс, натуральный логарифм по основанию «e», возвести в степень 3; отсекает целую часть, оставляет дробную | | З | sinh-1**,**  exp, asin, acos | обратный гиперболический синус, экспонента, арксинус, арккосинус | | И | Mod, tanh, asin, acos | вычислить остаток от деления одного числа на другое, гиперболический тангенс; арксинус, арккосинус | | К | Ln, X3, Dms, sin | натуральный логарифм по основанию «e», возвести в степень 3; переводит из десятичного вида в формат в градусы, минуты, секунды; синус | | Л | Х3, asin, acos | Возведение в куб, арксинус, арккосинус | | М | Sinh, Mod,  y√x, lg10 | гиперболический синус, вычислить остаток от деления одного числа на другое; y-ый корень числа x, где y обычно является положительным целым числом, десятичный логарифм | | Н | deg,  y√x, lg10, sin, cos | перевод угла в градусах, минутах и секундах в десятичные доли градуса; y-ый корень числа x, где y обычно является положительным целым числом, десятичный логарифм; синус, косинус | | О | Int, Pi, tanh, Ln, X3 | отображает целую часть десятичного числа, число Пи, выдает значение Pi для расчетов; гиперболический тангенс, натуральный логарифм по основанию «e», возвести в степень 3 | | П | Inv, sin, cos, tan | обратная функция для sin, cos, tan, переключает интерфейс на другие функции; синус; косинус; тангенс | | Р | Dms, sin, cos, tan | переводит из десятичного вида в формат в градусы, минуты, секунды; синус; косинус; тангенс | | С | Dms, 10^x, Pi, tanh, Ln | переводит из десятичного вида в формат в градусы, минуты, секунды; возведение десяти в произвольную степень, число Пи, гиперболический тангенс, натуральный логирифм | | Т | F – E, acos, atg, log\_xy, n! | переключает ввод чисел в экспоненциальном представлении и обратно; арккосинус, арктангенс, логорифм по основанию, факториал | | У | Pi, sin,tan,exp, asin, acos | Число Пи, синус, тангенс, экспонента, арксинус, арккосинус | | Ф | Mod, tanh, Ln, X3, exp, asin, acos | вычислить остаток от деления одного числа на другое, гиперболический тангенс, натуральный логарифм по основанию «e», возвести в степень 3, экспонента, арксинус, арккосинус | | Х | pow(a, b), n!, F – E, acos | функция выполняет возведение числа a в степень b и возвращает затем вещественный результат; вычисление факториала; переключает ввод чисел в экспоненциальном представлении и обратно; арккосинус | | Ц | Pi, sin,tan, floor() | Число Пи, синус, тангенс; Округляет число до ближайшего целого, но в меньшую сторону | | Ч | sin, cos, tan, Ln, X3 | синус; косинус; тангенс; натуральный логарифм по основанию «e», возвести в степень 3 | | Ш | Mod, tanh, F – E, acos | вычислить остаток от деления одного числа на другое, гиперболический тангенс, переключает ввод чисел в экспоненциальном представлении и обратно; арккосинус | | Щ | F – E, acos, tanh, Ln, X3 | переключает ввод чисел в экспоненциальном представлении и обратно; арккосинус; гиперболический тангенс, натуральный логарифм по основанию «e», возвести в степень 3 | | Э | tanh, Ln, X3, log\_xy, n! | гиперболический тангенс, натуральный логирифм, возведение в куб, логорифм по основанию, факториал | | Ю | X3, 10^x, Pi, tanh | возведение в куб, возведение десяти в произвольную степень, число Пи, гиперболический тангенс | | Я | tanh, asin, acos, F – E | гиперболический тангенс; арксинус, арккосинус, переключает ввод чисел в экспоненциальном представлении и обратно |   Примерный внешний вид, реализующий данный функционал может быть следующим:    ВАЖНО:  1. Программа должна использовать распространенные библиотеки. Если с согласия руководителя предполагается для реализации дополнительных функций и отрисовки интерфейса использование сторонних библиотек, то в пояснительной записке к курсовой работе должно быть представлено обоснование и подробное их описания.  2. При разработке программы следует использовать принципы объектно-ориентированного программирования.  3. Калькулятор должен работать и корректно выдавать результат.  4. Размер кнопок, цветовая палитра, расположение и т.п. определяется самостоятельно. Внешний вид калькулятора не оценивается, однако следует придерживаться традиционных компоновок.  5. При намеренном выполнении некорректных операций, например, извлечение квадратного корня из отрицательного числа, калькулятор должен на цифровой дисплей выдавать соответствующее сообщение об ошибке. Если же программу останавливает интерпретатор – то проверяемая функция считается не до конца отработанной.  **Программу «Калькулятор» сохранить под именем exercise\_3.py** |
| Входные данные | Преподаватель с помощью мыши вводит числа и выполняет арифметические операции.  Преподаватель с помощью мыши вводит числа и выбирает заявленные дополнительные функции.  Преподаватель вводит намеренно некорректные операции (попытка деления на 0) |
| Выходные данные | На цифровом дисплее должен отображаться результат, либо сообщение об ошибке при некорректной операции.  Проверяется количество строк в «цифровом дисплее» калькулятора и количество ячеек памяти. |

Задание № 4

|  |  |
| --- | --- |
| Задача о Ханойских башнях | С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ (GUI) |
| формулировка | Модифицированная задача о Ханойских башнях:  Существует 8 шпинделей, пронумерованых от 8 до 1 слева направо. На каждом шпинделе надеты диски, в количестве, равном соответствующей цифре из ID студента. Все диски имеют разные диаметры. Диаметр диска равен M \* 10 + N, где М – номер шпинделя, на котором надет диск, а N – это номер диска на шпинделе, считая сверху вниз.   1. Необходимо визуально изобразить предложенную задачу. Диски на шпинделях сделать случайных цветов. На каждом диске отображать цифру, равную его диаметру. Диаметр диска также показывать его фактическим размером в пикселях. 2. Необходимо вычислить, за какое минимальное количество итераций переместятся все диски на шпиндель номер 1 по следующим правилам:   а) За одну итерацию можно переместить не более одного диска  б) Диски можно класть только с большего на меньший  в) Со шпинделя номер 8 можно перекладывать диски только на шпиндели 7 и 6  г) Со шпинделя номер 1 можно перекладывать диски только на шпиндели номер 2 и 3  д) Со шпинделей от 2 по 7 можно перекладывать диски только на два соседних шпинделя.  3. Необходимо отобразить начальное и конечное расположение дисков на шпинделях, для этого под изображением Ханойских башен предусмотреть две кнопки «Начало» и «Окончание». При нажатии на нее, в надписи под схемой должен выводится текст «Итерация ХХ», где ХХ – номер итерации (либо 0, либо номер итоговой итерации, соответственно).  4. Необходимо графически отобразить четыре промежуточные итерации перекладывания дисков. Для этого:  а) общее количество итераций признаётся равным 100%,  б) ID студента делится на 4 двузначных числа, каждое из которых обозначает итерацию, соответствующую этому проценту выполнения общей задачи.  в) Под изображением Ханойских башень предусмотреть четыре поля для ввода цифр с процентами выполнения. По-умолчанию добавить туда числа из п. б)  г) Под каждым полем для ввода предусмотреть кнопку, при нажатии на которую схема Ханойской башни отображает расположение дисков на соответствующей итерации. Также в надписи под схемой должен выводится текст «Итерация ХХ», где ХХ – номер итерации  5. Дать возможность пользователю изменять проценты в полях для ввода цифр, и по нажатию соответствующей кнопки просматривать расположение дисков на данной итерации. |
| Методические указания | Так как задача алгоритмически достаточно проста, то основная часть работы над задачей студента сводится к правильной визуализации полученных результатов, а также оптимальному поиску промежуточных результатов. Схема ханойских башен должна выглядеть примерно таким образом:  ХБ  (В данном примере расположение дисков соответствует ID студента 70256421)  Так как размер диска должен соответствовать его номеру на шпинделе и номеру самого шпинделя, умноженного на 10, то как нетрудно догадаться, максимальный диаметр диска может быть 89. Поэтому для правильной визуализации без наложения дисков рекомеднуется выдерживать расстояние между шпинделями примерно в 100-120 пикселей (для окна формата 1280х1024).  Поскольку диски близких размеров будут отличаться всего на один пиксель, то для контроля на каждом диске необходимо проставить его диамер в виде цифры. Так как в конце задачи все диски будут находиться на первом шпинделе, а общее количество дисков теоретически может быть равно 72, то рекомендуется сделать толщину одного диска примерно равной 10-12 пикселям, для указанного окна.  Остальные элементы управления под схемой Ханойской башни рекомендуется выстраивать в следующем порядке:  На данном рисунке также видно, как следует разбивать ID студента для выведения промежуточных итогов. При нажатии на кнопки, нужно показать итерации, соответствующие 70%, 25%, 64% и 21% выполнения задачи.  **Уточнение**  Если по какому-либо проценту получается дробная итерация, то необходимо её визуализировать как промежуточный этап переноса диска. При этом диск изобразить в воздухе, между тем шпинделем, с которого он снят, и тем, на который он переносится. Номер итерации в таком случае отображать как дробный, с округлением до 3 цифр после нуля.  **Программу сохранить под именем exercise\_4.py** |
| Входные данные | Идентификатор студента.  Промежуточные проценты, вводимые преподавателем в соответсвующие поля над кнопками. |
| Выходные данные | На цифровом дисплее должно отображаться окно с начальным расположением дисков на шпинделях Ханойских башень. Шпиндели пронумерованы, на дисках также обозначены соответствующие диаметры. Под ней отображается шесть кнопок и четыре поля для ввода цифр. В нижней части экрана демонстрируется надпись «Итерация 0»  При нажатии на любую из шести имеющихся кнопок, либо при заполнении поля ввода другими данными и нажатии на кнопку, схема ханойских башень меняется, для отображения соответствующей итерации. Надпись в нижней части экрана также меняется. |

# Порядок выполнения курсовой работы

Тема курсовой работы является стандартной для всех обучающихся. В рамках курсовой работы обучающийся разрабатывает прикладное программное обеспечение для решения нескольких заданий. Формулировки заданий имеют общую и индивидуальную части. Индивидуализация заданий курсовой работы по дисциплине «Высокоуровневые методы программирования» реализуется на основе уникального номера ID студента и его ФИО. Заведующий кафедрой назначает научного руководителя. После консультаций с научным руководителем выбираются библиотеки и составляются примерные алгоритмы решения каждого задания. Необходимо отметить, что используя возможности различных библиотек для Python возможны разные подходы к решению заданий курсовой работы.

Помимо основной литературы также рекомендуется использование интернет-источников по тематике языка программирования python. Это, в первую очередь специализированные форумы, техническая документация, обзорные статьи по тематическим сайтам. Их изучение в контексте дисциплины «высокоуровневые методы программирования» служит расширению научно-технического кругозора, повышению качества и обоснованности использованных программных решений.

В процессе выполнения курсовой работы по согласованию с руководителем возможна конкретизация и уточнение формулировок исходных данных в задачах. Данные корректировки обязательно должны быть отражены в пояснительной записке курсового проекта.

**В процессе выполнения курсовой работы студенты должны:**

* Изучить задания на выполнение курсовой работы и разработку программного продукта;
* Выполнить анализ и составить примерную стратегию решения каждого задания;
* Выбрать необходимые библиотеки для реализации предметных областей программного продукта;
* Разработать алгоритмы и реализовать их в среде разработки с помощью выбранных библиотек
* Для отдельных заданий разработать графический интерфейс пользователя средствами библиотек Tkinter, wxPython, PyQT или PySimpleGUI;
* Выбрать стратегию тестирования и разработать тесты;
* Выполнить тестирование и отладку;
* Подробно описать решение каждого задания, форматы входных и выходных файлов.
* Сформировать пояснительную записку к курсовой работе, привести листинг программного кода для каждого задания.
* Сформировать архив исходных текстов программ, входных и результирующих файлов.

# Сроки выполнения отдельных этапов

Выполнение курсовой работы осуществляется в несколько этапов. В конце каждого этапа студент демонстрирует преподавателю результаты выполнения работы в виде фрагментов пояснительной записки и соответствующих текстовых или графических материалов, иллюстрирующих разработку программного приложения.

Основные этапы курсовой работы, объем их выполнения, оценка (по 100-бальной шкале) и представляемые преподавателю результаты, приведены в таблице 5.1.

**Таблица 5.1**

Этапы выполнения курсовой работы

|  |  |  |  |
| --- | --- | --- | --- |
| Этап | Содержание этапа | Оценка и  объем готовности работы | Представляемые результаты |
| 1 | - Изучение методических указаний и исходных данных к каждому заданию на выполнение курсовой работы;  - Анализ и составление примерной стратегии решения каждого задания  - Выбор необходимых библиотек для реализации предметных областей заданий | Объем работы  - 20 %  Оценка:  5 баллов | 1-я контрольная точка  Введение. Первая глава пояснительной записки курсовой работы.  Перечень библиотек, планируемых к использованию для выполнения заданий к курсовой работе. |
| 2 | Разработка алгоритмов и программной реализации каждого задания. | Объем работы  - 60 % | 2-я контрольная точка  Вторая глава пояснительной записки курсовой работы.  Интерфейс программного продукта, специальный раздел пояснительной записки: информационная модель, описание входных данных. |
|  | 1. Выполнение задания № 1 | Оценка:  10 баллов | Разработанный алгоритм и программная реализация на языке программирования.  ***Представляемые файлы к проверке задания № 1:***  *resourse\_1.txt – текстовый файл с входными данными*  *result\_1.txt – текстовый файл с результатами работы программы задания № 1*  *exercise\_1.py – файл с исходным программным кодом* |
|  | 2. Выполнение задания № 2: | Оценка:  10 баллов | Разработанный алгоритм и программная реализация на языке программирования и с использованием дополнительных библиотек.  ***Представляемые файлы к проверке задания № 2:***  *resourse\_2.txt – текстовый файл с входными данными*  *result\_2.txt – текстовый файл с результатами работы программы задания № 2*  *exercise\_2.py – файл с исходным программным кодом* |
|  | 3. Выполнение задания № 3:  - реализация стандартного функционала | Оценка:  10 баллов | Интерфейс программного продукта, специальный раздел пояснительной записки: проектирование стандартного функционала.  ***Представляемые файлы к проверке задания № 3.1:***  *exercise\_3.py – файл с исходным программным кодом* |
|  | - реализация расширенного функционала | Оценка:  30 баллов | Интерфейс программного продукта, специальный раздел пояснительной записки: проектирование расширенного функционала.  ***Представляемые файлы к проверке задания № 3.2:***  *exercise\_3.py – файл с исходным программным кодом* |
|  | - реализация дополнительных функций расширенного функционала | Оценка:  15 баллов | Интерфейс программного продукта, специальный раздел пояснительной записки: Индивидуальное задание реализации дополнительных функций.  ***Представляемые файлы к проверке задания № 3.3:***  *exercise\_3.py – файл с исходным программным кодом* |
|  | 4. Выполнение задания № 4:  - реализация модифицированной задачи о Ханойских башнях | Оценка:  15 баллов | Интерфейс программного продукта, специальный раздел пояснительной записки: Индивидуальное задание по расположению дисков в соответствие с ID студента.  ***Представляемые файлы к проверке задания № 4:***  *exercise\_4.py – файл с исходным программным кодом* |
| 3 | Тестирование и отладка программного продукта. Разработка технической документации по сопровождению программного продукта | Объем работы  - 90 %  Оценка:  5 баллов | 3-я контрольная точка  Третья глава пояснительной записки курсовой работы.  Готовый программный продукт, инструкция пользователю по работе с программным продуктом, специальный раздел пояснительной записки: тестирование и отладка. |
| 4 | Завершение оформления пояснительной записки | Объем работы  - 100% | 4-я контрольная точка  Полностью оформленная записка в электронном варианте. Архив исходников к каждому заданию. Выгрузка окончательного варианта в электронный университет |
| 5 | Защита курсовой работы |  | Программа, записка, приложения. |

**Порядок сдачи и защиты курсовой работы**

На защиту студент предоставляет:

- Архив в формате \*.zip или \*.rar файлов исходных кодов, входных и выходных тесктовых данных, имеющих название в соотвествие таблицей 5.1. Имя архивного файла должно содержать фамилию, инициалы студента и через знак подчеркивания его ID, например: «Иванов И.В\_ 70149127.zip»

- Пояснительную записку на 20-40 страницах, оформленную в текстовом редакторе. Имя файла пояснительной записки должно содержать фамилию, инициалы студента «Иванов И.В.docx».

Проверку курсовой работы руководитель осуществляет только в Электронном университете. **Окончательный вариант архива исходных кодов «Иванов И.В\_ 70149127.zip» и пояснительная записка «Иванов И.В.docx» к курсовой работе должны быть выгружены в Электронный университет**.

Программные решения и пояснительная записка могут быть предварительно выгружены студентом в статусе «черновик» в личный кабинет Электронного университета для проверки преподавателем.

Пояснительная записка может содержать следующие приложения:

- листинг - текст программного приложения;

- документацию в виде готовых выходных форм, указанную в заданиях.

В программном продукте проверяется:

* соответствие программы заданиям;
* работоспособность в различных режимах.

На защите студент коротко (3–5 мин.) докладывает об основных программных решениях, принятых в процессе разработки, и отвечает на вопросы преподавателя, и при необходимости демонстрирует работающий програмный код. Программный код должен работать корректно, используя интерпретатор Python 3.9 в среде разработке PyCharm Community.

Оценка за курсовую работу выставляется с учетом:

* качества выполненного программного продукта по каждому заданию, соотвествие требованиям Соглашения «PEP 8»
* работоспособности программного кода
* наличие подробных комментариев в коде
* правильности оформления записки
* результатов защиты

Оценивание курсовой работы осуществляется по балльно-рейтинговой системе, установленной для всех форм учебной деятельности студента.

В случае несогласия студента с оценкой, защита курсовой работы выполняется при комиссии, состоящей не менее чем из двух преподавателей.

# Содержание пояснительной записки

Пояснительная записка должна содержать обоснование выбора библиотек и алгоритмических решений, принятых обучающимся на каждом этапе разработки. Решения должны приниматься исходя из особенностей разрабатываемого задания и специфики исходных данных. Не должно быть обоснований типа «удобнее», «целесообразнее» и т. п. Необходимо пояснить, чем удобнее, почему целесообразно использование той или иной библиотеки. По возможности необходимо четко формулировать основания для принятия того или иного алгоритмического решения.

**Пример содержания пояснительной записки:**

Введение

1. Анализ заданий курсовой работы

1.1. Исходные данные к заданиям курсовой работы

1.2. Анализ методических указаний, входных и выходных данных к заданиям курсовой работы

1.2. Выбор и обоснование необходимых библиотек и среды разработки

1.3. Выводы по 1 главе

**2. Разработка программного продукта для решения прикладных задач.**

**2.1. Работа с наборами данных**

2.1.1. Построение алгоритма решения задания без графического интерфейса

2.1.2. Разработка программной реализации на языке программирования

2.1.3. Тестирование и отладка.

2.1.4. Формирование выходных файлов

**2.2. Разработка экспертной системы**

2.2.1. Построение алгоритма решения задания с графическим интерфейсом

2.2.2. Разработка программной реализации на языке программирования и с использованием дополнительных библиотек

2.2.2.1. Проектирование стандартного функционала

2.2.2.2. Проектирование расширенного функционала

2.2.2.3. Индивидуальное задание реализации дополнительных функций

2.2.3. Тестирование и отладка.

2.2.4. Формирование выходных файлов

**2.3. Разработка аналитической системы**

2.3.1. Построение алгоритма решения задания с графическим интерфейсом

2.3.2. Разработка программной реализации на языке программирования и с использованием дополнительных библиотек

2.3.3. Тестирование и отладка.

2.3.4. Формирование выходных файлов

**2.4. Разработка логико-аналитической системы**

2.4.1. Построение алгоритма решения задания «Ханойские башни»

2.4.2. Разработка программной реализации на языке программирования и с использованием дополнительных библиотек

2.4.3. Тестирование и отладка.

2.4.4. Формирование выходных файлов

2.5. Выводы по 2 главе

3. Разработка требований к техническим средствам реализации программного обеспечения для решения прикладных задач

Выводы

Список литературы

Приложения

Приложение А. Листинг текстов заданий

Приложение Б. Образцы GUI заданий

*Примечание* - В зависимости от выбранной технологии и особенностей используемых библиотек могут иметь место отклонения от рекомендуемой структуры записки. Возможно добавление новых разделов, если этого требует логика изложения. Все интересные материалы, выходящие за пределы рекомендуемого объема записки, выносятся в приложение. Объем приложения не лимитируется.

# Правила оформления пояснительной записки

## Оформление текстового и графического материала

Пояснительная записка оформляется в текстовом редакторе. Графический материал можно оформлять на слайдах презентации в соответствующем офисном пакете. Поля на листе определяются в соответствии с общими требованиями. При использовании текстовых редакторов для оформления записки параметры страницы заказываются в зависимости от устройства печати. При ручном оформлении выбираются из соображений удобства.

Нумерация страниц – сквозная. Номер проставляется сверху справа арабской цифрой. Страницами являются листы с текстами, рисунками и текстами приложения.

Первая страница – титульный лист расчетно-пояснительной записки. Номер страницы на титульном листе не проставляется. Образец титульного листа представлен в Приложении Б.

Вторая страница – Аннотация

Третья страница – оглавление, отражающее содержание изложенного материала. Ни аннотация, ни оглавление в содержании не упоминаются.

Затем следуют разделы записки в порядке, определенном заданием на курсовую работу.

Записка завершается списком литературы.

Далее могут следовать приложения, содержащие материал, не вошедший в записку по причине ее ограниченного размера, но интересный для более глубокого понимания назначения и возможностей разработки. Пояснительная записка может содержать одно и более приложений.

Наименование разделов и подразделов пишутся строчными буквами, кроме первой прописной. Расстояние между заголовками и текстом, а также между заголовками раздела и подразделов должно быть равно:

при выполнении документа машинописным способом – 1,5 интервалам;

при выполнении рукописным способом –10 мм.

Разделы и подразделы нумеруются арабскими цифрами с точкой. Разделы должны иметь порядковые номера 1, 2, и т. д. Номер подраздела включает номер раздела и порядковый номер подраздела, входящего в данный раздел, разделенные точкой. Например: 2.1., 3.5.

Перечисления надо нумеровать арабскими цифрами со скобкой; Например: 2), 3) и т. д. – с абзацного отступа. Допускается выделять перечисление простановкой дефиса перед пунктом текста или символом, его заменяющим, в текстовых редакторах.

## Оформление рисунков, схем алгоритмов, таблиц и формул

Иллюстрации (графики, схемы, диаграммы) могут быть приведены как в основном тексте, так и в приложении. Все иллюстрации именуются рисунками. Все рисунки, таблицы и формулы нумеруются арабскими цифрами последовательно (сквозная нумерация). В приложении – в пределах приложения. Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. Чертежи, графики, диаграммы, схемы должны соответствовать требованиям ЕСКД.

Рисунки, за исключением рисунков приложений следует нумеровать арабскими цифрами сквозной нумерацией.

Допускается нумеровать рисунки в пределах раздела. В этом случае номер рисунка состоит из номера раздела и порядкового номера рисунка, разделенных точкой.

Если рисунок один, то он обозначается «Рисунок 1». Рисунок может иметь наименование и пояснительные данные (подрисуночный текст). Слово «рисунок» и его наименование располагают посередине строки. Если есть подрисуночный текст, то слово «рисунок» и его наименование помещают после пояснительных данных.

Например: Рисунок 1 - Форма окна основного меню.

На все рисунки, таблицы и формулы в записке должны быть ссылки в виде:

«(рисунок 1)» или «форма окна основного меню приведена на рисунке 1».

Слово «рисунок» и его наименование располагается посредине строки.

Рисунки и таблицы должны размещаться сразу после той страницы, на которой, в тексте записки, она упоминается в первый раз. Если позволяет место, рисунок (таблица) может размещаться в тексте на той же странице, где на него дается первая ссылка.

Если рисунок занимает более одной страницы, на всех страницах, кроме первой, проставляется номер рисунка и слово «Продолжение». Например: Рисунок 1 – Продолжение.

Рисунки следует размещать так, чтобы их можно было рассматривать без поворота записки. Если такое размещение невозможно, рисунки следует располагать так, чтобы для рассматривания надо было повернуть записку по часовой стрелке. В этом случае верхним краем является левый край страницы. Расположение и размеры полей сохраняются в соответствии с установленными.

Схемы алгоритмов должны быть выполнены в соответствии со стандартом ЕСПД. Толщина сплошной линии при вычерчивании схем алгоритмов должна быть в пределах от 0,6 до 1,5 мм. Надписи на схемах должны быть выполнены чертежным шрифтом. Высота букв и цифр должна быть менее 3,5 мм.

Название таблицы, при его наличии, должно отражать ее содержание, быть точным, кратким. Название следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире. Заголовок, кроме первой буквы, выполняется строчными буквами. В аббревиатурах используются только заглавные буквы. Например: ПЭВМ.

При переносе части таблицы название помещается только над первой частью таблицы, нижнюю горизонтальную черту, ограничивающую таблицу, не проводят.

Ссылки на таблицы в тексте пояснительной записки должны быть в виде слова «таблица» с указанием ее номера. Например, «Результаты тестов приведены в таблице 4».

Таблицу с большим количеством строк допускается переносить на другой лист (страницу). При переносе слово «Таблица» и ее номер указывается один раз справа над первой частью таблицы, а над другими частями пишут слово «Продолжение» и указывают номер таблицы, например: «Продолжение таблицы 1». При этом заголовок помещают только над ее первой частью.

Таблицы, за исключением таблиц приложений следует нумеровать арабскими цифрами сквозной нумерацией.

Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделенных точкой.

Уравнения и формулы следует выделять из текста в отдельную строку. Ниже и выше каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Пояснения значений символов и числовых коэффициентов следует проводить непосредственно под формулой в той же последовательности, в которой они даны в формуле. Формулы следует нумеровать порядковой нумерацией в пределах записки арабскими цифрами. Номер формулы ставится в крайне правом положении на строке в круглых скобках на уровне формулы. Например:

z:=sin(x)+ln(y); (12)

Ссылка на номер формулы дается в скобках. Например: «расчет значений производится по формуле (12)».

**Оформление примечаний**.

Слово «Примечание» следует печатать с прописной буквы с абзаца и не подчеркивать. Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается с прописной буквы и примечание не номеруется. Несколько примечаний нумеруются по порядку арабскими цифрами без проставления точки. Примечание к таблице помещают в конце таблицы над линией, обозначающей окончание таблицы.

Пример одного примечания:

Примечание - \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пример нескольких примечаний:

Примечания

1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Оформление текстов программ

Тексты программ должны оформляться в соответствии с «Соглашением PEP 8». **В текстах должны быть комментарии:**

* после заголовка программы или подпрограммы приводится общая информация: назначение, входные данные, результаты, метод решения; данные о программисте, дата написания, версия;
* при объявлении данных - назначение переменных;
* в начале и в конце определенной функционально законченной части программы;
* для пояснения логических частей программы (ветвлений, циклов).

Наименование функция и процедур должны отражать их назначение. Логическая структура программы должна быть отражена в ее тексте с помощью:

* пустых строк между текстами блоков отдельных ее функционально законченных частей;
* сдвигами текста в строке при написании:
* заголовков вложенных циклов;
* тела цикла после его заголовка;
* альтернатив разветвлений процесса обработки данных.

## Оформление выводов

Выводы должны содержать краткие итоги и рекомендации по результатам проведенной работы, отвечать на вопросы, поставленные во введении. Рекомендуемый объем одна-две страницы.

## Оформление списка литературы

Список литературы должен включать все использованные источники. Сведения о книгах (монографиях, учебниках, пособиях, справочниках и т.д.) должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство, год издания. При наличии трех и более авторов допускается указывать фамилию и инициалы только первого из них со словами «и др.». Наименование места издания надо приводить полностью в именительном падеже: допускается сокращение названия только двух городов: Москва (М.) и Санкт-Петербург (С-П.).

Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, наименование статьи, наименование издания (журнала), наименование серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и номера страниц, на которых помещена статья.

При ссылке на источник из интернета надо указывать полный электронный адрес доступа к источнику информации.

## Оформление приложений

Каждое приложение должно начинаться с новой страницы с указанием наверху посередине страницы слова «ПРИЛОЖЕНИЕ» прописными буквами и иметь тематический заголовок, который записывается симметрично относительно текста с прописной буквы отдельной строкой. Приложения обозначаются заглавными буквами русского алфавита, начиная с А.

Страницы приложений не нумеруются.

# КРИТЕРИИ ОЦЕНКИ ДЛЯ КУРСОВОЙ РАБОТЫ

Оценка «**отлично**» ставится, если выполнены все требования к написанию и защите курсовой работы. Программный код во всех заданиях работает корректно. Итоговый балл >=85.

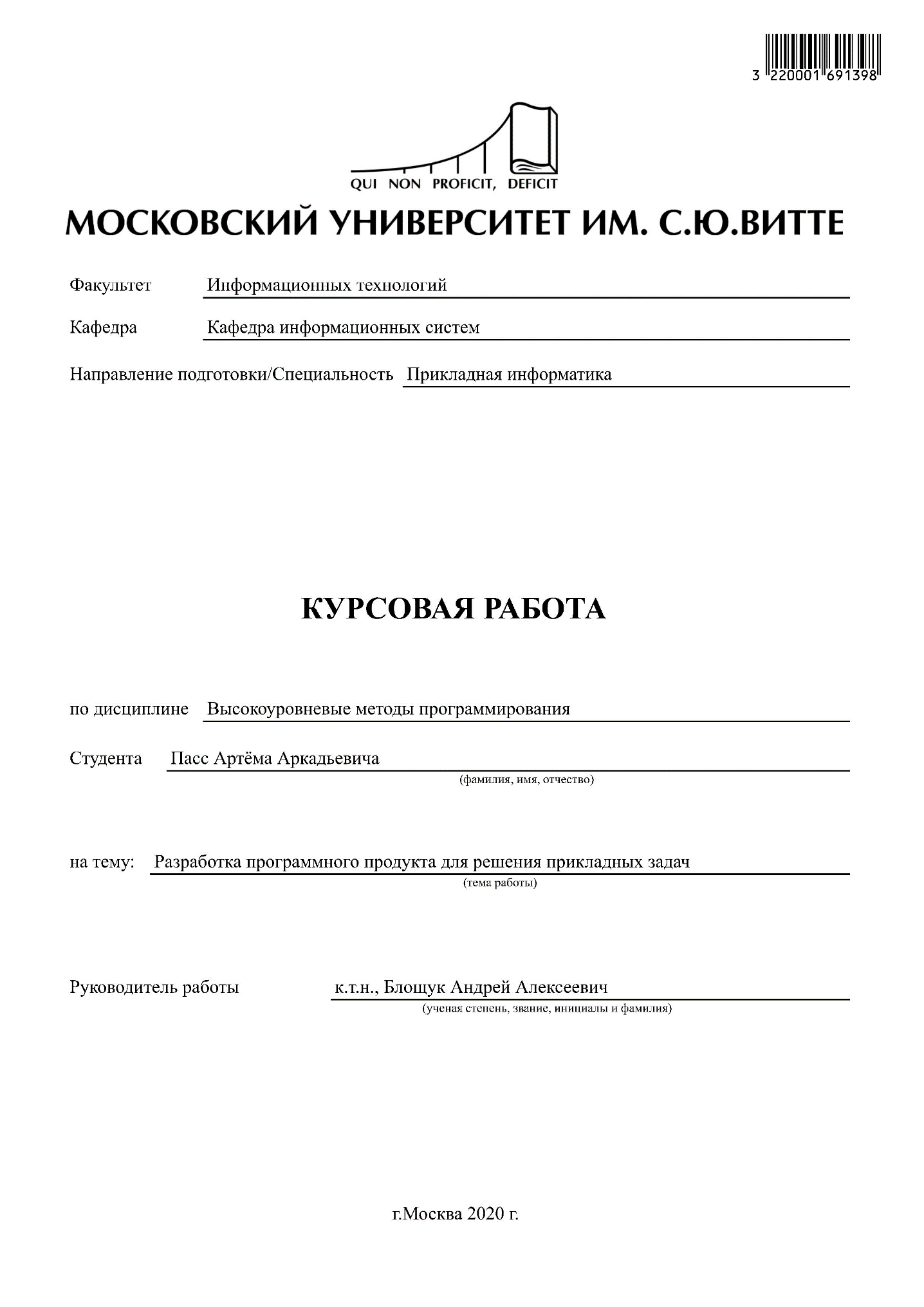
Оценка «**хорошо**» – основные требования к КР и ее защите выполнены, но при этом допущены недочёты. В частности, имеются неточности в входных и выходных данных; не выдержаны требования «Соглашения PEP 8»; имеются упущения в оформлении; дополнительные вопросы при защите даны неполные ответы. Программный код во всех заданиях работает корректно. Итоговый балл от 70 до 84.

Оценка «**удовлетворительно**» – имеются существенные отступления от требований к КР. Допущены фактические ошибки в понимании заданий курсовой работы. Выполнены не все задания. Программный код работает корректно как минимум в 3-х заданиях. Итоговый балл от 51 до 69.

Оценка «**неудовлетворительно**» – задания не выполнены, обнаруживается существенное непонимание постановки задач. Программный код работает не корректно, интерпретатор выдает ошибки. Итоговый балл – менее 50 баллов.

# Приложение

Образец титульного листа



*Прим. титульный лист генерируется по запросу в центр поддержки. Подписи на титульном листе не ставятся.*