



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
«НАЦИОНАЛЬНЫЙ МИНЕРАЛЬНО-СЫРЬЕВОЙ УНИВЕРСИТЕТ «ГОРНЫЙ»

**Кафедра информационных систем и вычислительной техники**

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**к выполнению курсового проекта**

**по дисциплине**

**«Технологии программирования»**

(наименование по рабочему учебному плану)

**Специальность: 09.03.02 Информационные системы и технологии**

**САНКТ-ПЕТЕРБУРГ  
2022**

Составитель:

доцент В.А. Медведев

Научный редактор:

профессор И.В. Иванова

## ЦЕЛИ И ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ

Целями курсового проектирования является изучение современного подхода к программированию на основе объектно-ориентированной технологии, приобретение навыков разработки программ на языке Java.

Для достижения целей в процессе курсового проектирования должны быть решены следующие задачи:

- выбор (формулирование) темы задания и детализация поставки задачи;
- разработка проекта приложения (модель данных, управление, пользовательский интерфейс);
- абстрагирование, разработка классов и их иерархии;
- создание текста (кодирование) разработанного проекта на языке программирования;
- разработка тестовых примеров;
- тестирование и отладка программы;
- разработка отчетных документов в соответствии с действующими стандартами.

### 1. ЭТАПЫ КУРСОВОГО ПРОЕКТИРОВАНИЯ

**Начальным этапом** курсового проектирования является выбор (формулирование) темы проекта и уяснение задания (форму см. в Приложении1). Опыт показывает, что для того, чтобы минимизировать количество ошибок при разработке программы, необходимо вникнуть в содержание задачи путем уяснения её условия, четко представить себе основные функции, которые надо реализовать, исходные данные для работы программы, накладываемые ограничения и так далее. На основе этого составляется ориентировочный план работы.

**Этап разработки проекта приложения** состоит в декомпозиции общей задачи на части (модули, файлы) и описание их назначения и функций.

Одним из широко используемых шаблонов (pattern) проектирования систем является Model-View-Controller (MVC, «модель-представление-поведение», «модель-представление-контроллер», «модель-вид-контроллер»). Шаблон проектирования или паттерн - повторяющаяся архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

В рамках шаблона MVC модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Данная концепция проектирования часто используется для построения архитектурного каркаса при переходе к реализации в конкретной предметной области.

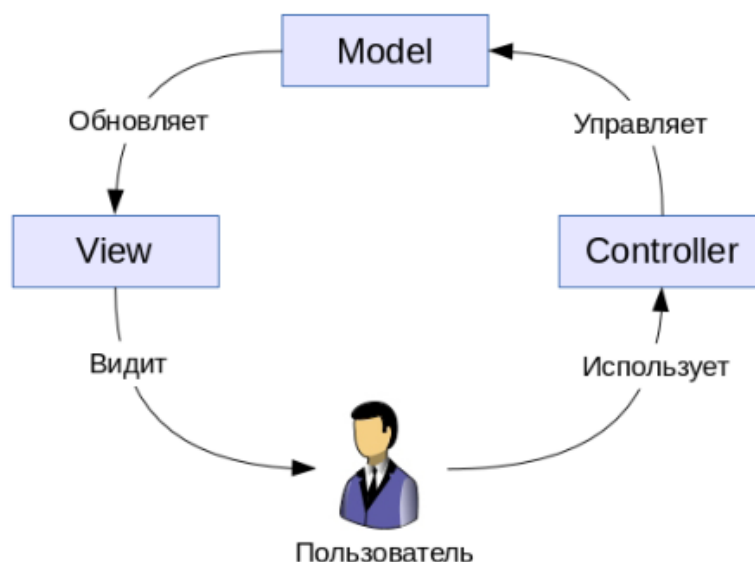


Рис. Схема MVC

Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

*Модель (Model).* Модель представляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.

*Представление, вид (View).* Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (диалоговое окно) с графическими элементами (поля, кнопки и др.).

*Контроллер (Controller).* Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Важно отметить, что как представление, так и контроллер зависят от модели. Однако модель не зависит ни от представления, ни от контроллера. Тем самым достигается назначение такого разделения, что имеется возможность строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели (таблицы, графики,..).

Результатом данного этапа должна быть схема структуры классов с описанием их назначения и состава (схематично).

**Этап написания программы.** Необходимо стремиться, чтобы программа получалась простая, наглядная и читаемая. Чем проще написана программа, чем лучше она оформлена, тем удобнее такая программа в изучении и эксплуатации. Удобочитаемость программы связана со стилем программирования. Поясним основные требования хорошего стиля программирования.

*Первое требование.* Программа должна содержать комментарии. Комментарии нужны как на стадиях проектирования и отладки программы, так и позже. Можно выделить комментарии вводные и пояснительные. Вводные комментарии должны помещаться в начале каждой программы или подпрограммы (функции) и пояснять их назначение. Пояснительные комментарии сопровождают наиболее трудно описываемые части программы. Средней нормой можно считать одну строку комментариев на десять строк программы.

*Второе требование* хорошего стиля программирования заключается в пропуске строк, что значительно повышает наглядность программы. Как в естественном языке мы пользуемся пропусками строк для отделения разделов, так и в программе с помощью пропуска строк можно разделять ее отдельные фрагменты. Например, пропуском одной строки можно отделять группу логически связанных операторов, пропуском двух строк – основные логически связанные фрагменты программы. Использование незаполненных строк облегчает поиск отдельных частей в программе.

*Третье требование* заключается в рациональном выборе имен переменных так, чтобы они наилучшим образом характеризовали те величины, которые они представляют. Все обозначения объектов в алгоритме и программе должны по возможности совпадать.

*Четвертое требование* касается размещения операторов. В большинстве языков программирования высокого уровня допускается размещение нескольких операторов на одной строке. Однако запись нескольких операторов на одной строке затрудняет чтение программы. Кроме того, по сообщению транслятора о синтаксической ошибке легче обнаружить эту ошибку при размещении в одной строке одного оператора.

*Пятое требование.* Для указания связи операторов, между ними делают одинаковые отступы от начала строки. Отступы, не оказывая влияния на логику программы, существенно улучшают ее читаемость. Правильно сделанные отступы хорошо подчеркивают структуру программы.

**Этап тестирования и отладки программы.** Этот этап является наиболее сложным этапом, так как общих и одновременно эффективных методов поиска ошибок не существует.

Все ошибки, которые возникают в процессе работы можно разделить на две большие группы: алгоритмические и программные. *Алгоритмические* ошибки приводят к неправильному или неэффективному решению задачи и обусловлены либо некорректной постановкой задачи, либо непониманием ее сути, либо неправильным выбором или применением математического аппарата. *Программные* ошибки возникают из-за недостаточного знания программистом языка и технологии программирования. Кроме алгоритмических и программных ошибок могут также возникать ошибки, вызываемые невнимательными действиями программиста в процессе набора программы.

При тестировании и отладке программы выполняются следующие виды работ:

а) Подготовка контрольных вариантов для проверки правильности решения поставленной задачи по программе. Контрольные варианты представляют собой наборы исходных данных и соответствующие им результаты решения задачи для наиболее характерных частных случаев выполнения программы. Контрольные варианты обычно рассчитывают вручную.

б) Планирование процесса отладки. Планирование состоит в выборе режима трансляции в интегрированной среде программирования, в определении контрольных точек для проверки ошибочных ситуаций, в формировании списка тех переменных, значения которых необходимо контролировать в этих точках в процессе выполнения программы или при других отладочных действиях.

в) Трансляция и исправление обнаруженных транслятором синтаксических ошибок. Исправление выявленных транслятором синтаксических ошибок представляет собой сложный этап отладки программы. Следует отметить, что на этапе трансляции выявляются не все ошибки. Ошибки логического характера транслятор обнаружить не может.

Из-за большого разнообразия синтаксических ошибок невозможно дать конкретные рекомендации по поводу способов их устранения. Однако можно указать следующий подход. Если в программе выявлено очень большое количество ошибок, следует просто исправить понятные и очевидные ошибки и повторить трансляцию. Некоторые сообщения о синтаксических ошибках являются на самом деле ложными, так как вызваны другими ошибками. Поэтому не стоит тратить много времени на то, чтобы разобраться с каждым выданным сообщением, если характер ошибки

не очевиден сразу. Ложных сообщений об ошибках особенно много тогда, когда в программе имеет место какое-то серьезное упущение, например, неправильное написание начала главной функции, отсутствие описания массива и так далее.

Так как вышерасположенные в программе ошибки влияют на синтаксис следующего за ними кода, их и надо исправлять в первую очередь. При этом после исправления каждой ошибки повторять трансляцию кода.

г) Выполнение программы и исправление логических ошибок. Программист приступает к этому виду работ только после устранения всех ошибок, обнаруженных на предыдущем этапе отладки программы. Выполнение программы производят для варианта, разработанного ранее в качестве контрольного. Если при этом получаются правильные результаты, то следует переходить к проверке программы на других вариантах. В случае если результаты не верны, производится обнаружение и исправление логических ошибок и повторная проверка программы по контрольным вариантам. После совпадения результатов выполнения программы со всеми контрольными вариантами отладка считается законченной.

**Этап анализа полученных результатов.** Это исследовательский этап, в рамках которого производится использование уже разработанной программы для получения искомых результатов, осуществляется анализ результатов решения задачи и в случае необходимости - уточнение математической модели. Программы, имеющие большое практическое значение, используются длительное время. Иногда даже в процессе эксплуатации программы могут исправляться, дорабатываться.

**Этап подготовки отчетных документов.** Заключительным этапом является формирование пояснительной записки.

## **2. СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

Пояснительная записка по своему содержанию должна соответствовать заданию на курсовую работу. В структурном плане пояснительная записка должна включать в себя следующие части:

- титульный лист;
- аннотацию;
- содержание (оглавление);
- основную часть;
- заключение;
- список использованной литературы;
- приложения.

**Титульный лист** выполняется стандартным образом (см. Приложение Б).

**В аннотации** кратко излагается содержание работы и выходные данные пояснительной записки (количество страниц, рисунков и др.). Объём аннотации - до одной страницы. Аннотация оформляется по правилам основной части, описание которой будет приведено ниже.

**Содержание (оглавление)** пояснительной записки включает номера и наименования разделов и подразделов основной части пояснительной записки с указанием номера листа, на котором размещается их начало.

**Основная часть** содержит материал, подлежащий обязательному представлению. Она включает следующие составляющие.

- 1) Замысел и состав проекта. Здесь указывается, в каком направлении планируется развивать тему проекта и до каких границ, а также – какие возможности (результаты) у разработанного приложения (программы). Состав проекта описывается на уровне файлов и классов .
- 2) Текст программы с комментариями.
- 3) Результаты работы программы (виды экрана и файлов).

**Заключение** должно содержать конкретные выводы по результатам работы.

Примерный перечень пунктов выводов имеет следующий вид:

- какая решена задача (учебная, техническая, тактическая или какая-то другая);
- вид задачи и метод её решения;
- структура проекта и особенности программы (состав проекта, как осуществлена проверка вводимых данных, какие основные средства Java использовались при реализации проекта и почему, как реализован дружественный интерфейс пользователя и др.)
- какие задачи оказались сложными и почему;
- техническая характеристика разработанной программы (язык программирования, версия, среда программирования, объём кода и т.д.);
- в какой области профессиональной деятельности может использоваться созданное приложение.

**Список использованной литературы.** В пояснительной записке должны быть сделаны ссылки на использованную литературу. Список использованной литературы должен содержать только те источники, которые использовались при разработке курсового проекта и на которые есть ссылки в тексте пояснительной записки.

**Приложения.** Материал, дополняющий текст документа, допускается помещать в приложениях. Приложениями могут быть, например, экранные формы ин-



терфейсов разработанной программы, графический материал, расчёты, образцы оформления, таблицы и т.д.

### **3. ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ**

#### **Общие требования**

Изложение текста и оформление пояснительной записки курсовой работы выполняются в соответствии с общеустановленными требованиями. Страницы текста и включенные в пояснительную записку иллюстрации и таблицы должны соответствовать формату А4. Текст печатается на одной стороне листа белой бумаги через полтора интервала. Цвет шрифта - черный. Размер шрифта (кегель) - не менее 12 пт. Обычная практика - кегль 14 пт в основном тексте. В [1] не определяется тип шрифта, но обычно - Times New Roman.

Текст следует печатать без автоматических переносов, соблюдая следующие размеры полей: правое - не менее 10 мм, верхнее и нижнее - не менее 20 мм, левое - не менее 30 мм. Межстрочный интервал – полуторный; абзацный отступ (табуляция) – 1,25 см, выравнивание – по ширине.

Страницы работы нумеруются арабскими цифрами (нумерация сквозная по всему тексту). Номер страницы ставится в центре нижней части листа без точки. Титульный лист и аннотация или реферат включаются в общую нумерацию, но номера на них не ставятся. Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц.

Качество напечатанного текста и оформления иллюстраций, таблиц должно удовлетворять требованию их четкого воспроизведения.

Опечатки, описки и графические неточности, обнаруженные в процессе подготовки пояснительной записки, допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного текста черными чернилами, пастой или тушью - рукописным способом.

Фамилии, названия организаций, фирм, названия программных продуктов должны приводиться на языке оригинала. Допускается транслитерировать имена собственные и приводить названия организаций в переводе на русский язык с добавлением (при первом упоминании) оригинального названия.

#### **Заголовки**

К структурным элементам пояснительной записки относятся "Аннотация", "Содержание", "Заключение", "Список использованных источников", "Приложение". Заголовки структурных элементов располагают в середине строки без точки в конце

и печатают прописными (заглавными) буквами, не подчеркивая. Каждый структурный элемент следует начинать с новой страницы.

Основную часть отчета пояснительной записки курсовой работы делят на разделы, подразделы и пункты. Пункты, при необходимости, могут делиться на подпункты.

Разделы, подразделы, пункты и подпункты нумеруются арабскими цифрами и записываются с абзацного отступа. Их обычно нумеруют в пределах всего текста, кроме приложений, следующим образом: разделы – 1, 2, 3 и т.д., подразделы – 1.1, 1.2, 1.3 и т.д., пункты – 1.1.1, 1.1.2, 1.1.3, подпункты – 1.1.1.1, 1.1.1.2, 1.1.1.3 и т.д. После номера раздела, подраздела, пункта и подпункта в тексте точку не ставят.

Разделы и подразделы должны иметь заголовки. Пункты, как правило, заголовков не имеют. Заголовки должны четко и кратко отражать содержание разделов, подразделов. Заголовки разделов и подразделов следует печатать с абзацного отступа, с прописной буквы без точки в конце, не подчеркивая.

Пример:

1 Математическое моделирование

1.1 Постановка задачи

Если заголовок состоит из двух предложений, их разделяют точкой. Переносы слов в заголовках не допускаются.

Расстояние между заголовком и текстом должно быть равно 3 или 4 интервалам (15 мм). Если пояснительная записка напечатана интервалом 1,5, то это значит, что расстояние между заголовком и текстом равно одной пустой строке. Расстояние между заголовками раздела и подраздела - 2 интервала (8 мм).

## **Содержание**

Заголовок СОДЕРЖАНИЕ пишется заглавными буквами посередине строки.

Содержание включает введение (если оно есть), наименование всех разделов, подразделов, пунктов, заключение, список использованных источников и наименование приложений с указанием номеров страниц, с которых начинаются эти элементы пояснительной записки.

## **Иллюстрации**

На все иллюстрации (чертежи, графики, схемы, компьютерные распечатки, диаграммы, фотоснимки) в тексте должны быть даны ссылки. Иллюстрации должны располагаться непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Иллюстрации, за исключением иллюстрации приложений, нумеруются арабскими цифрами, при этом нумерация сквозная, но допускается нумеровать и в пре-

делах раздела. В последнем случае, номер рисунка состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой (например: Рисунок 1.1).

Подпись к иллюстрации располагается под ней посередине строки. Слово «Рисунок» пишется полностью. Можно ограничиться только номером (т.е. оставить, например, подпись: Рисунок 2), но при необходимости, иллюстрации могут иметь наименование и пояснительные данные (подрисуночный текст). В этом случае подпись должна выглядеть так: Рисунок 2 – Алгоритм подпрограммы. Точка в конце подрисуночного текста не ставится.

Если в работе есть приложения, то иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением впереди обозначения приложения (например, Рисунок А.3).

При ссылках на иллюстрации следует писать "... в соответствии с рисунком 2" при сквозной нумерации и "... в соответствии с рисунком 1.2" при нумерации в пределах раздела.

### **Таблицы**

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. На все таблицы в тексте должны быть ссылки. При ссылке следует писать слово "таблица" с указанием ее номера. Таблица должна располагаться непосредственно после текста, в котором она упоминается впервые, или на следующей странице.

Все таблицы нумеруются (нумерация сквозная, либо в пределах раздела - в последнем случае номер таблицы состоит из номера раздела и порядкового номера внутри раздела, разделенных точкой (например: Таблица 1.2). Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением впереди обозначения приложения (например, Таблица В.2). Слово «Таблица» пишется полностью.

Если в пояснительной записке одна таблица, то она должна быть обозначена "Таблица 1" или "Таблица В.1", если она приведена в приложении В.

Наличие у таблицы собственного названия не обязательно, но при необходимости название таблицы следует помещать над таблицей слева, без абзачного отступа, в одну строку, с ее номером через тире (например, Таблица 3 - Доходы фирмы). Точка в конце названия не ставится. Название таблицы, при его наличии, должно отражать ее содержание, быть точным, кратким.

Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово "Таблица", ее номер и наименование указывают один раз слева над первой частью таблицы, а

над другими частями также слева пишут слова "Продолжение таблицы" и указывают номер таблицы.

Таблицу с большим количеством столбцов допускается делить на части и помещать одну часть под другой в пределах одной страницы.

Если строки и столбцы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае - боковик.

При делении таблицы на части допускается ее головку или боковик заменять соответственно номером столбцов и строк. При этом нумеруют арабскими цифрами столбцы и (или) строки первой части таблицы.

Заголовки столбцов и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки столбцов - со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков столбцов и строк таблиц точки не ставят.

Заголовки столбцов, как правило, записывают параллельно строкам таблицы, но при необходимости допускается их перпендикулярное расположение.

Таблицы слева, справа и снизу, как правило, ограничивают линиями. Допускается применять размер шрифта в таблице меньший, чем в тексте.

Разделять заголовки и подзаголовки боковика и столбцов диагональными линиями не допускается.

Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить, если их отсутствие не затрудняет пользование таблицей. Но головка таблицы должна быть отделена линией от остальной части таблицы.

Цифровой материал, как правило, оформляют в виде таблиц. Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

### **Формулы и уравнения**

Формулы и уравнения следует выделять из текста в отдельную строку. Над и под каждой формулой или уравнением нужно оставить по пустой строке. Если уравнение не умещается в одну строку, то оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:), или других математических знаков, причем этот знак в начале следующей строки повторяют.

Если нужны пояснения к символам и коэффициентам, то они приводятся сразу под формулой или уравнением в той же последовательности, в которой они идут в формуле.

Все формулы (уравнения) нумеруются. Обычно нумерация сквозная. Номер проставляется арабскими цифрами в круглых скобках в крайнем правом положении на строке.

Пример

$$A = a / (b + c) \quad (1)$$

$$B = c / (b - c) \quad (2)$$

Ссылки в тексте на порядковые номера формул дают в скобках, например: в формуле (1).

### **Перечисления**

Перед каждым перечислением следует ставить дефис или, при необходимости ссылки в тексте на одно из перечислений, строчную букву (за исключением ё, з, й, о, ч, ь, ы, ь).

Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа.

а) \_\_\_\_\_

б) \_\_\_\_\_

1) \_\_\_\_\_

2) \_\_\_\_\_

в) \_\_\_\_\_

### **Список использованных источников**

В пояснительной записке должны быть сделаны ссылки на использованную литературу. Ссылаются следует на документ в целом или его разделы и приложения. Ссылки на подразделы, пункты, таблицы и иллюстрации не допускаются, за исключением подразделов, таблиц и иллюстраций данного документа. Ссылки на использованные источники следует приводить в квадратных скобках, например, [15].

Список литературы должен называться «Список использованных источников». Заголовок данного структурного элемента печатается прописными буквами в центре строки.

Независимо от вида носителя, на котором размещён документ, в его библиографическую запись включается следующий перечень полей.

Область заглавия включает заголовок библиографической записи, который представляет собой фамилию первого автора с инициалами или их расшифровкой. Если авторов более трёх, то книгу следует описывать под её заглавием.

После заголовка следует заглавие, которое приводится без сокращения.

Элемент «Общее обозначение материала» используют в тех случаях, когда из имеющихся элементов записи не видна характеристика физического носителя доку-

мента (например, описание базы данных, расположенной на локальном компьютере и др.). Элемент указывают в квадратных скобках сразу после заглавия (*например: [Электронный ресурс], [Аудиозапись] и т.д.*). Если из других элементов библиографической записи понятно, о каком физическом носителе идет речь, допустимо опустить этот элемент.

Сведения, относящиеся к заглавию, приводятся после двоеточия и представляют собой сведения, дополняющие или разъясняющие заглавие и расположенные на титульном листе документа.

Сведения об ответственности приводятся после косой черты и включают в себя фамилии авторов, указанных на титульном листе книги, начиная с первого и до четвертого, если больше четырех – первые три и слова «и др.»; сведения о редакторах, художниках, переводчиках и т.д. с указанием их роли.

Сведения об издании приводятся в том случае, когда присутствует переиздание. *Например: 3-е изд., доп. и перераб.* Для электронного ресурса характерен выпуск версии издания, о чём и дается информация. *Например: – версия 5.61.*

Элемент «Выходные сведения» присутствует тогда, когда описывается документ целиком. В этом случае указываются место издания, издательство, год издания (например, М. : Проспект, 2018). При указании места издания указывается название города полностью - Кострома, сокращаются только Москва (М.) и Санкт-Петербург (СПб.).

Количественные характеристики содержат для описания документа на бумажном носителе информацию о количестве страниц в издании.

В процессе написания пояснительной записки курсовой работы могут быть использованы статьи, помещенные в сборниках или статьи из журналов, а также электронные ресурсы.

Среди электронных ресурсов выделяют два вида: имеющие физическую основу (CD-ROM; DVD и т.д.) и не имеющие физической основы, то есть доступные удалённо (сайты, базы данных по подписке и т.д.).

В ситуации с электронным ресурсом основным источником информации является экранная титульная страница сайта, а также весь ресурс в целом.

Для электронных ресурсов, имеющих физическую основу (CD-ROM; DVD и т.д.) в качестве титульного листа выступает этикетка компакт-диска.

Для удалённых электронных ресурсов приводятся сведения о режиме доступа. В этом случае требуется указать адрес (URL) и условия доступа (чаще рекомендуется указывать ограничения доступа). Например, <http://www.spbbt.ru/> или же UPL: <http://www.nlr.ru/>. Также указываются сведения об истории удаленного ресурса -

указывают дату последнего посещения ресурса. Эти сведения приводятся в скобках и не отделяются от адреса какими-либо знаками. Например: (22.03.2019).

Пример выполнения списка литературы:

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шилдт, Герберт. Java: руководство для начинающих, 7-е изд. : Пер. с англ. - М. ООО "Диалектика", 2019. - 816 с.
2. Шилдт, Герберт. Java 8. Полное руководство; 9-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильямс", 2015. - 1376 с.
3. Хорстманн, Кей С., Корнелл, Гари. Java. Библиотека профессионала, том 1. Основы. 9-е изд.: Пер. с англ. - М.: ООО "И.Д.Вильямс", 2014.- 864 с.
4. Хорстманн, Кей С., Корнелл, Гари. Java. Библиотека профессионала, том 2. Расширенные средства, 9-е изд. :Пер. с англ. - М.:ООО "И.Д.Вильямс", 2014. - 1008 с.
5. Монахов, В. Язык программирования Java и среда NetBeans. - М.: БХВ-Петербург, 2012. - 720 с.
6. Блох, Дж. Java: эффективное программирование / Дж. Блох. - М.: Диалектика, 2019. - 464 с.

Приложение оформляют как продолжение пояснительной записки на последующих ее листах. В тексте пояснительной записки на все приложения должны быть даны ссылки. Приложения располагают в порядке ссылок на них в тексте. Каждое приложение следует начинать с новой страницы с указанием наверху справа страницы слово "Приложение" и его обозначение. Приложения обозначают заглавными буквами русского алфавита или цифрами, начиная с А, за исключением букв Е, З, Й, О, Ч, Ъ, Ы, Ь. После слова "Приложение" следует буква, обозначающая его последовательность. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O.

Если в пояснительной записке одно приложение, оно обозначается "Приложение А".

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения.

Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

#### 4. ЗАЩИТА КУРСОВОГО ПРОЕКТА

При выполнении курсового проекта необходимо помнить следующее:

- работа, не имеющая самостоятельного характера и представляющая собой механически списанные куски из различных источников, возвращается студенту для доработки, либо полного переделывания, при этом преподаватель вправе поставить студенту оценку “неудовлетворительно”;
- небрежно и безграмотно написанные работы существенно снижают оценку;
- курсовой проект должен быть представлен в срок, установленный для его приёма.

##### КРИТЕРИИ ОЦЕНКИ

Оценка			
«2» (неудовлетворительно)	Пороговый уровень освоения	Углубленный уровень освоения	Продвинутый уровень освоения
	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Студент не выполнил курсовой проект в соответствии с заданием. Не владеет теоретическими знаниями по изучаемой дисциплине. Необходимые практические компетенции не сформированы	Студент выполнил курсовой проект с существенными ошибками. При защите курсового проекта демонстрирует слабую теоретическую подготовку. При решении задач, предусмотренных программой учебной дисциплины, допускает неточности, существенные ошибки	Студент выполнил курсовой проект с некоторыми незначительными ошибками и неточностями. При защите курсового проекта демонстрирует хорошую теоретическую подготовку. Хорошо справляется с решением задач, предусмотренных программой учебной дисциплины	Студент выполнил курсовой проект полностью в соответствии с заданием. При защите курсового проекта демонстрирует высокую теоретическую подготовку. Успешно справляется с решением задач, предусмотренных программой учебной дисциплины

Защита курсового проекта проводится устно в форме доклада в случае, если работа допущена к защите, с целью формирования окончательной оценки за работу студента. На защиту представляется разработанный в интегрированной среде проект работающей программы и пояснительная записка.



При получении неудовлетворительной оценки студенты выполняют курсовой проект по новой теме или перерабатывают прежний проект в установленные сроки.

## 5. ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ

1. Разработка фрагмента информационной системы «Медицинские услуги» (регистратура, карточки пациентов, запись на приём по кабинетам).
2. Разработка фрагмента информационной системы «Недвижимость» (клиенты, агенты, сделки).
3. Разработка электронного учебника для библиотеки JavaFX.
4. Разработка фрагмента информационной системы «Конструктор путешествий».
5. Разработка приложения для учета расходов физических лиц.
6. Разработка фрагмента системы анализа текста на предмет выявления неправомерных заимствований.
7. Разработка фрагмента информационной системы для электронного голосования.
8. Разработка приложения для распознавания и идентификации с QR-кодов.
9. Информационная обучающая система программированию на Java (фрагмент).
10. Разработка тестирующей программы для оценки знаний по языку программирования Java.
11. Моделирование компьютерной сети (стационарной и с изменяемой топологией) (Жуковский).
12. Разработка имитационной модели функционирования коммутатора (внешний вид, логика коммутации кадров) (Жуковский).
13. Разработка системы классов и графического интерфейса для реализации матричной арифметики.
14. Разработка системы автоматизированного раскроя материала. Исходные данные: размер материала для раскроя, геометрия элементов, которые из него необходимо вырезать, режим работы (автоматизированный или ручной), критерии оптимальности раскроя.
15. Разработка системы – редактора диаграмм классов. Система должна позволять пользователю проектировать диаграммы классов с определением свойств, методов классов и отношений между ними.

16. Разработка системы моделирования движения транспортных средств (на нерегулируемых перекрестках). Система должна моделировать и визуализировать движение транспортных средств на перекрестке с учетом параметров (тип перекрестка, количество полос, ограничение скорости) и соблюдением водителями правил дорожного движения. Исходные данные: тип транспортного средства и размеры, стиль вождения (максимальная и минимальная скорость движения, склонность к перестроениям, допустимый интервал до впереди идущей машины).
17. Разработка программного средства поиска информации в видеоконтенте. Программа должна предоставлять пользователю возможность осуществлять поиск видеофайлов по графическому изображению одного из кадров или его фрагмента.
18. Графический редактор для изображения схем алгоритмов.
19. Разработка информационной системы «Компьютерный магазин».
20. Моделирование случайных процессов.
21. Разработка интерфейса ввода данных для фото-галереи.
22. Разработка логической игры с использованием стандартных шаблонов проектирования.
23. Моделирование канала связи и расчет его характеристик (Жуковский).
24. Разработка электронного учебника для библиотеки Swing.
25. Свободная тема.

ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
 федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
 «НАЦИОНАЛЬНЫЙ МИНЕРАЛЬНО-СЫРЬЕВОЙ УНИВЕРСИТЕТ «ГОРНЫЙ»

**Кафедра информационных систем и вычислительной техники**

## КУРСОВОЙ ПРОЕКТ

по дисциплине \_\_\_\_\_  
 (наименование учебной дисциплины согласно учебному плану)

### ЗАДАНИЕ

1. Тема проекта \_\_\_\_\_  
 \_\_\_\_\_
2. Исходные данные к проекту \_\_\_\_\_  
 \_\_\_\_\_
3. Содержание пояснительной записки \_\_\_\_\_  
 \_\_\_\_\_
4. Перечень графического материала \_\_\_\_\_  
 \_\_\_\_\_
5. Срок сдачи законченного проекта \_\_\_\_\_ 20 \_\_ г.
6. Задание выдал (Руководитель проекта) \_\_\_\_\_ / \_\_\_\_\_ /  
 (подпись) (должность, Ф.И.О.)
7. Задание принял к исполнению студент \_\_\_\_\_ / \_\_\_\_\_ /  
 (подпись) (Ф.И.О.)
8. Дата получения задания для курсового проекта: \_\_\_\_\_ 20 \_\_ г.

ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ

---



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  
«НАЦИОНАЛЬНЫЙ МИНЕРАЛЬНО-СЫРЬЕВОЙ УНИВЕРСИТЕТ «ГОРНЫЙ»

**Кафедра информационных систем и вычислительной техники**

## КУРСОВОЙ ПРОЕКТ

по дисциплине \_\_\_\_\_  
(наименование учебной дисциплины согласно учебному плану)

### ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Тема проекта: \_\_\_\_\_  
\_\_\_\_\_

Автор: студент гр. \_\_\_\_\_ / \_\_\_\_\_ /  
(шифр группы) (подпись) (Ф.И.О.)

Оценка: \_\_\_\_\_

Дата: \_\_\_\_\_

Проверил:  
руководитель проекта \_\_\_\_\_ / \_\_\_\_\_ /  
(должность) (подпись) (Ф.И.О.)

Санкт-Петербург  
202\_\_ г.