

**Факультет Информационных систем и технологий**  
**Кафедра Безопасности информационных систем**

# **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**

## 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 C++.

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

- выбор темы задания и детализация поставки задачи;
- разработка проекта приложения;
- создание текста программы на языке программирования;
- тестирование и отладка программы;
- разработка пояснительной записки к курсовому проекту.

### 1. ЭТАПЫ КУРСОВОГО ПРОЕКТИРОВАНИЯ

**Начальным этапом** курсового проекта является выбор (формулирование) и уяснение задания (форму см. в Приложении А). 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 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.

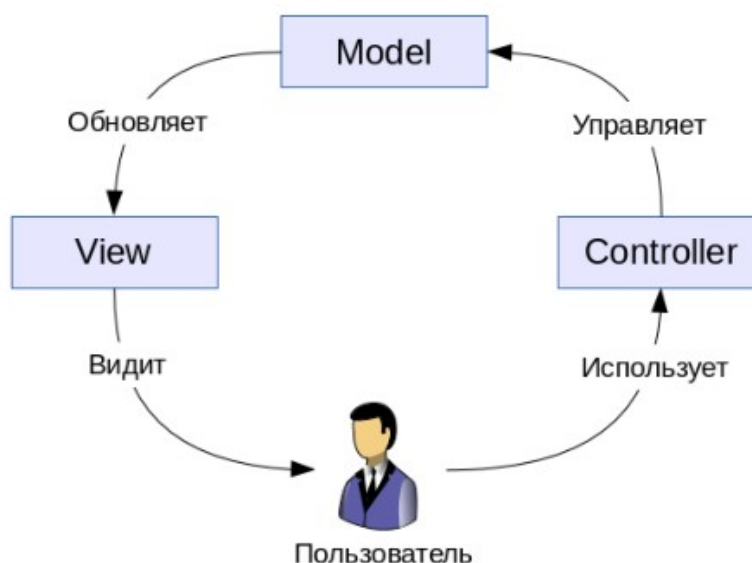


Рис. Схема MVC

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,...).

**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.

*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 получают 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;
- содержание (оглавление);
- основную часть;
- заключение;
- список использованной литературы;
- приложения.

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

**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, классов или основных функций (см. Приложение В).
- 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:

- какая решена задача и какой использовался подход при программировании (объектно-ориентированный или процедурный);
- основные составляющие проекта (классы) и его возможности;
- особенности программы (какие основные средства языка программирования использовались при реализации проекта и почему, как реализован дружественный интерфейс пользователя, как осуществлена проверка вводимых данных и др.)
- какие задачи оказались сложными и почему;
- техническая характеристика разработанной программы (какая среда разработки применялась, какие библиотеки использовались (стандартных функций, STL), объём кода и т.д.);
- в какой области деятельности может использоваться созданное приложение и направление его развития.

**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.

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

Пояснительная записка представляется *в электронном виде*.

### **Общие требования**

Изложение текста и оформление пояснительной записки курсовой работы выполняют в соответствии с общеустановленными требованиями. Страницы текста и включенные в пояснительную записку иллюстрации и таблицы должны соответствовать формату А4. Цвет шрифта - черный. Размер шрифта (кегель) - не менее 12 пт. Обычная практика - кегль 14 пт в основном тексте. Тип шрифта произвольный, но обычно - Times New Roman. Размеры полей: правое - не менее 10 мм, верхнее и нижнее - не менее 20 мм, левое - 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 включается в общую нумерацию, но номер на нём не ставится. 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.

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: "Содержание", составляющие основной части, "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.

### **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 (графики, схемы, компьютерные распечатки, диаграммы) 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 пределах 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 абзацного 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.

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.

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.

### **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. Ссылаться 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.

Сведения об использованных в пояснительной записке источниках следует располагать в порядке появления ссылок на источники в тексте и нумеровать арабскими цифрами без точки и печатать с абзацного отступа.

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 документа (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 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.

Если материал помещен в журнале, то сначала указываются фамилия и инициалы автора материала, название материала, затем ставится знак «//», после которого помещаются сведения об издании, в котором помещен данный материал: указывается название журнала и все его «выходные данные», помещенные на оборотной стороне титульного листа (серия, год, номер и т.д.). Важно помнить о том, что в конце сведений о материале указывается нумерация страниц, на которых начинается и заканчивается публикация.

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: (22.03.2022).

Пример выполнения списка литературы:

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шилдт, Герберт. С++: базовый курс; 3-е изд.: Пер. с англ. - М. : ООО "И.Д. Вильямс", 2019. - 624 с.
2. Шилдт, Герберт. С++: методики программирования Шилдта.: Пер. с англ. - М. : ООО "И.Д. Вильямс", 2017. - 480 с.
3. Страуструп, Бьёрн. Язык программирования С++ (стандарт С++11). Краткий курс. Пер. с англ. – М.: ИД Бином, 2019. – 176 с.
4. Павловская Т. А. С/С++. Программирование на языке высокого уровня. – СПб.: Питер, 2018. – 461 с.

#### 4. ЗАЩИТА КУРСОВОГО ПРОЕКТА

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 в электронном виде.

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

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;
- курсовой проект должен быть выполнен в рамках объектно-ориентированного подхода, однако допускается его разработка только средствами процедурного программирования (без объектов); в последнем случае оценка снижается на один балл.

#### 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	Студент выполнил курсовой проект с существенными	Студент выполнил курсовой проект	Студент выполнил курсовой проект

соответствии с заданием. Не владеет теоретическими знаниями по изучаемой дисциплине. Необходимые практические компетенции не сформированы	ошибками. При защите курсового проекта демонстрирует слабую теоретическую подготовку. При решении задач, предусмотренных программой учебной дисциплины, допускает неточности, существенные ошибки	с некоторыми незначительными ошибками и неточностями. При защите курсового проекта демонстрирует хорошую теоретическую подготовку. Хорошо справляется с решением задач, предусмотренных программой учебной дисциплины	полностью в соответствии с заданием. При защите курсового проекта демонстрирует высокую теоретическую подготовку. Успешно справляется с решением задач, предусмотренных программой учебной дисциплины
---	---	---	---

## 5. ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ

Тематика курсовых проектов предусматривает моделирование определённого мероприятия (вида деятельности). В каждой теме указаны (не полностью) составляющие её элементы, раскрывающие содержание темы. Это содержание можно (и нужно) корректировать в соответствии со своим замыслом, тем самым выбирая направление раскрытия темы и масштаб (сложность) проекта. Кроме этого следует выбрать вычисляемые показатели, которые будут характеризовать созданную модель мероприятия (деятельности). Например: стоимость путешествия, продолжительность прогулки, количество калорий в пище и т.д.

Перечень тем:

1. «Куда пойти учиться»: обсуждение и выбор направления обучения, тип учебного заведения и его статус, правила поступления, особенности обучения и быта ...
2. «Покупка компьютера»: назначение компьютера, финансовые возможности, характеристики компьютера, консультанты, продавцы ...
3. «Покупка квартиры (дома)»: выбор местоположения, финансовые возможности, тип постройки, агентство недвижимости, сроки ...
4. «Посещение фитнес-клуба»: выбор направления и вида тренировочных программ, дата и время, снаряжение ...
5. «День рождения»: выбор места проведения, приглашение гостей, выбор (подготовка блюд), планирование мероприятий...

6. «Выезд на отдых»: выбор места отдыха и его продолжительности, покупка билетов на транспорт, формирование багажа, финансовые возможности, условия размещения...
7. «Посещение Эрмитажа»: выбор попутчика, обсуждение плана выставок и маршрута, дата и время, как добираться...
8. «Shopping»: планирование ассортимента покупок, финансовые возможности, выбор магазинов, разработка маршрута, выбор попутчиков...
9. «Курсовая работа»: выбор темы (предметной области), разработка содержания и графика выполнения работы, источники, консультанты ...
10. «На рыбалку»: дата и время, выбор водоема, снасти, наживка, прикормка, как добираться...
11. «Прогулка по городу»: дата и время, выбор маршрута и продолжительности прогулки, попутчики, использование транспорта ...
12. «Строительство дачной постройки»: тип и назначение постройки, финансовые возможности, смета и материалы, сроки, участники ...
13. «Посещение салона красоты»: выбор даты и дня недели, выбор салона и перечня услуг, на чем добираться, финансовые возможности...
14. «Пропала собака»: у человека пропала собака (кошка). Он искал её повсюду: обошёл приюты для животных, обзвонил все ветклиники по близости, обращался к прохожим, развешивал объявления ... и, наконец, нашёл.
15. «Поход в кино»: выбор фильма и кинотеатра, с кем идти, финансовые возможности, попутные мероприятия, как добраться...
16. «Скачки»: ипподром, как добраться, попутчики, бега, команды, ставки ...
17. «В лес за грибами»: состав компании, выбор места, экипировка, транспорт и маршрут ...
18. «Прогулка по каналам Санкт-Петербурга»: выбор маршрута и продолжительности прогулки, цена, попутчики, как добраться до пристани ...
19. «Посещение зоопарка»: планирование даты и времени, попутчики, план посещения и временные ресурсы, кормление животных ...
20. «Один день в Университете»: расписание, подготовка к занятиям, дорога, кто был на занятиях, что выполнено, ... , итоги дня.
21. «Биатлон»: вид гонки, состав участников и их возможности, жеребьевка, процесс гонки, победитель.

22. «На футбол (хоккей)»: выбор матча, состав команды, заказ и покупка билетов, транспорт и маршрут, игра (опасные и эмоциональные моменты), ... , итоги.
23. «Один дома»: наконец-то один, чем заняться, план или его отсутствие, мероприятия, ... , итоги дня.
24. «Диета»: выбор типа диеты, продолжительность, формирование меню, разработка графика, консультанты, результаты ...
25. «Обслуживание клиентов банка»: информация о клиентах, виды вкладов, счета клиентов, операции ...
26. «Ремонт бытовой техники»: типы обслуживаемой техники, вид работ (ремонт, гарантийное обслуживание ), заказы, клиенты, оплата услуг ...
27. «Доставка пиццы»: ассортимент, клиенты, заказы, скидки ...
28. «Обслуживание пациентов в поликлинике»: пациенты, врачи, приём пациентов, диагноз, стоимость обслуживания ...
29. «Регистрация на авиарейсы»: расписание и данные авиарейсов, количество свободных мест, текущая регистрация пассажиров и багажа ...
30. «Работник библиотеки»: информация о книгах ( автор, наименование, жанр, другие реквизиты, дата выдачи, дата возврата) и о читателях (паспорт, адрес, дата рождения, номер читательского билета, полученные книги), выдача и возврат книг.
31. «Продажа автомобилей»: информация об автомобилях, данные менеджеров, осуществляющих сделки, продажи, скидки ...
32. «Прокат электросамокатов»: виды сервисов, компании, типы самокатов, условия аренды, сделки ...
33. «Конструктор путешествий»: туроператоры, данные тура, выбор тура.
34. «Проверка компетенций»: разработка тестирующей программы для оценки уровня подготовки по языку программирования C++.
35. **Свободная тема** (требуется ОБЯЗАТЕЛЬНОГО обсуждения с преподавателем).



**Факультет Информационных систем и технологий**  
**Кафедра Безопасности информационных систем**

**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 \_\_\_\_\_ 202\_\_ 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: \_\_\_\_\_ 202\_\_ г.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**  
**федеральное государственное бюджетное образовательное учреждение высшего профессионального образования**  
**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ**  
**ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

---

**Факультет Информационных систем и технологий**  
**Кафедра Безопасности информационных систем**

**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.)

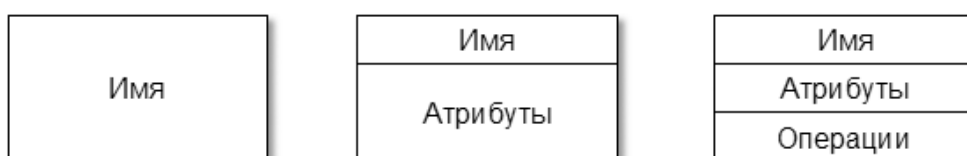
Санкт-Петербург  
202 \_\_ г.

## Графические диаграммы классов

**Диаграмма классов** (class diagram) — структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

Графически класс изображается в виде прямоугольника, разделенного на 3 части горизонтальными линиями:

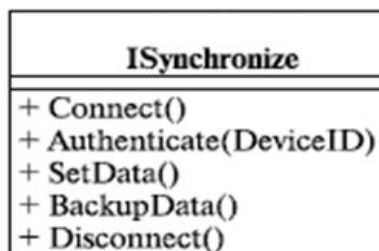
имя класса;  
атрибуты (поля) класса;  
операции (методы) класса.



Обозначения для спецификаторов доступа:

— — private (закрытый);  
# — protected (защищенный);  
+ — public (открытый).

Пример интерфейса:



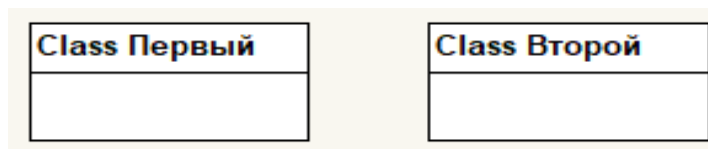
Дальнейшая детализация содержимого класса даёт указание типов данных членов класса.

Например:

Class Name
+ Attribute 1 : Type + Attribute 2 : Type - Attribute 3 : Type - Attribute 4 : Type
+ Operation 1 ( arg list ) : return + Operation 2 ( arg list ) : return + Operation 3 ( arg list ) : return + Operation 4 ( arg list ) : return

BankAccount
owner : String balance : Dollars = 0
deposit ( amount : Dollars ) withdrawal ( amount : Dollars )

Если классы являются автономными, то между ними связи отсутствуют.  
Например:

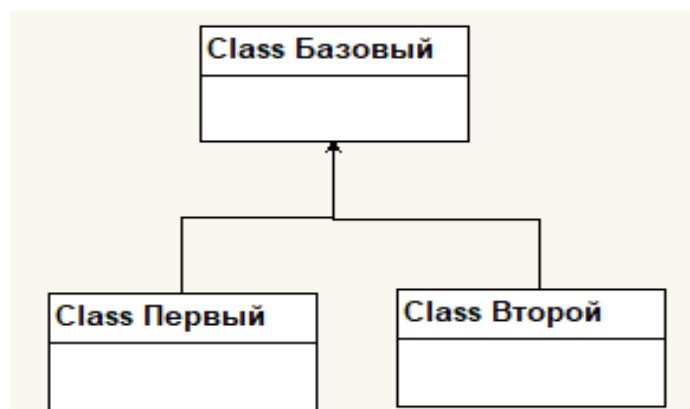


Наследование классов обозначается сплошной линией со стрелкой.  
Если

**class Первый: public Базовый ;**

**class Второй: public Базовый ;**

то схема классов формируется так:



Для представления размещения классов в файлах программы следует привести файловую структуру с указанием в файлах имён классов.

