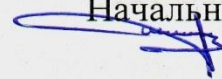


**Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Алтайский государственный технический университет
им. И. И. Ползунова»**

УТВЕРЖДАЮ

Начальник УМУ АлтГТУ



Н. П. Щербаков

" 20 " февраля 2017 г.

**Программа учебной практики
Практика по получению первичных профессиональных умений и
навыков, в том числе первичных умений и навыков научно-
исследовательской деятельности.**

**Направление подготовки
38.03.05 Бизнес-информатика**

**Направленность (профиль) подготовки
Информационные системы в экономике и управлении**

**Квалификация (степень) выпускника бакалавр
Для студентов очной, заочной и очно-заочной форм обучения**

Барнаул 2017

1 Цели учебной практики

Учебная практика ориентирована на профессиональнопрактическую подготовку студентов. Она направлена на закрепление и углубление теоретической подготовки студентов, приобретение ими практических навыков и компетенций, а также опыта самостоятельной профессиональной деятельности.

Целями практики являются:

- закрепление у студентов способностей и навыков по разработке и сопровождению программного обеспечения для систем различного назначения;
- приобретение навыков самостоятельного решения задач и выполнения работ по выбранному профилю подготовки и документальному оформлению достигнутых результатов;
- углубление теоретической подготовки в области связанной с профессиональной деятельностью;
- закрепление у студентов общекультурных и профессиональных компетенций, создание предпосылок самосовершенствования и профессионального роста личности.

2 Задачи практики

Задачами учебной практики являются:

в области научно-исследовательской деятельности:

- участие в проведении научных исследований (экспериментов, наблюдений и количественных измерений), связанных с объектами профессиональной деятельности (программными продуктами, проектами, процессами, методами и инструментами программной инженерии) в соответствии с утвержденными заданиями и методиками;
- построение моделей объектов профессиональной деятельности с использованием инструментальных средств компьютерного моделирования; составление описания проводимых исследований, подготовка данных для составления отчетов;

в области проектной деятельности:

- участие в проектировании компонентов программного продукта в объеме, достаточном для их конструирования в рамках поставленного задания;

- создание компонент программного обеспечения (кодирование, отладка, модульное и интеграционное тестирование);
- участие в интеграции компонент программного продукта;
- разработка тестового окружения, создание тестовых сценариев;
- разработка и оформление эскизной, технической и рабочей проектной документации.

Конкретный перечень задач, решаемых студентом в ходе практики, определяется поставленной перед ним алгоритмической задачей.

3 Место учебной практики в структуре образовательной программы

Учебная практика входит в раздел «Б2.У.1 Учебная практика» по направлению подготовки «Бизнес-информатика» (квалификация (степень) «Бакалавр»).

Учебная практика базируется на дисциплинах:

Программирование (1 сем.), Вычислительные системы, сети и телекоммуникации (2 сем), Объектно-ориентированное программирование (2 сем).

Задачи, поставленные студентам в период учебной практики, могут служить заданием для курсового проектирования при изучении таких дисциплин как Базы данных (5 сем.), Проектирование информационных систем (6 сем.) и Проектный практикум (7 сем.).

4 Формы, способы и тип проведения учебной практики

Форма проведения практики – дискретная по периодам проведения практик.

Способы проведения практики: стационарная практика, выездная практика.

Тип практики - практика по получению профессиональных умений и опыта профессиональной деятельности.

5 Место и время проведения практики

Практика может проводиться на кафедрах и в лабораториях ФГБОУ ВО «Алтайский государственный технический университет им.И.И. Ползунова», обладающих необходимым кадровым и научно-техническим потенциалом.

Продолжительность практики у студентов очной формы обучения– 2 недели во 2-ом семестре; у студентов очно-заочной формы- 1 1/3 недели 2-ом семестре; у студентов заочной формы- 1 1/3 недели 2-ом семестре.

6 Планируемые результаты обучения при прохождении практики

В результате прохождения учебной практики обучающийся должен приобрести следующие знания, умения, навыки:

знать:

1. задачи предметной области и методы их решения;
2. рынки информационных ресурсов и особенности их использования;
3. законы рыночной экономики.
4. сетевую экономику;
5. основы формирования и функционирования электронного бизнеса

уметь:

6. моделировать и оптимизировать бизнес-процессы предприятия
7. организовывать корпоративные информационные системы, их проектирование, внедрение и поддержку
8. планировать и организовывать исполнение IT-проектов разного уровня сложности.
9. заниматься финансовым анализом деятельности компании,
10. готовить периодическую отчетность для руководства компании,
11. разрабатывать бизнес-модели, требования к внедрению системы планирования ресурсов

владеть:

- 12.аналитической поддержкой принятия решений в управлении бизнесом
- 13.методологией построения бизнес-процесса,
- 14.технологиями анализа,
- 15.методами поиска и обработки информации

Код компетенции из УП и этап ее формирования	Содержание компетенции	В результате изучения дисциплины, обучающиеся должны:		
		знать	уметь	владеть
ПК-3 Начальный	выбор рациональных информационных систем и информационно-коммуникативных технологий решения для управления бизнесом	1-5	6-11	12-15
ПК-5 Начальный	проведение обследования деятельности и ИТ-инфраструктуры предприятий,	1-5	6-11	12-15

Код компетенции из УП и этап ее формирования	Содержание компетенции	В результате изучения дисциплины, обучающиеся должны:		
		знать	уметь	владеть
ПК-13 Начальный	умение проектировать и внедрять компоненты ИТ-инфраструктуры предприятия, обеспечивающие достижение стратегических целей и поддержку бизнес-процессов	1-5	6-11	12-15

7 Структура и содержание практики

Общая трудоемкость практики в 2 семестре:

у студентов дневной формы обучения составляет 3 зачетные единицы, 108 часов, 2 недели;

у студентов очно-заочной формы – 2 зачетных единицы, 72 часа, 1 1/3 недели;

у студентов заочной формы обучения – 2 зачетных единицы, 72 часа, 1 1/3 недели.

Таблица 1 – Содержание практики в 2 семестре.

№ п/п	Разделы (этапы) практики	Виды работы на практике, включая СРС и их трудоемкость в часах		Формы текущего контроля
		у студентов дневной формы обучения	у студентов очно-заочной и заочной форм обучения	
1	2	3	3	4
1	Инструктаж по технике безопасности	2	2	Запись в журнале инструктажа
2	Проектирование программного продукта в соответствии с заданием. Проектирование пользовательского интерфейса программного продукта. Конструирование, отладка и тестирование программного продукта в соответствии с заданием.	98	62	Представление руководителю практики промежуточных отчетов, содержание которых определяется вариантом задания.
3	Подготовка отчета по практике	8	8	Защита
	ИТОГО:	108	72	

8 Перечень информационных технологий, используемых при проведении практики

При прохождении учебной практики используются следующие технологии:

- технология поиска и отбора информации;
- технология развития критического мышления;
- Интернет - технологии;
- сетевые технологии;
- технологии использования программно-технического обеспечения;
- технология проектной деятельности;
- технология проблемного обучения путем инициирования самостоятельного поиска студентом знаний, необходимых для решения поставленной проблемы;
- технология контекстного обучения путем интеграции различных видов деятельности студентов: учебной, научной, практической и создания условий, максимально приближенных к реальным.

9 Учебно-методическое обеспечение самостоятельной работы студентов на практике

На практику направляются студенты, не имеющие академической задолженности.

Руководство практикой студентов со стороны кафедры "Информационные системы в экономике" осуществляется академическими руководителями, назначенными из числа преподавателей кафедры.

Обязанности руководителя практики от университета:

- подготовить проект приказа о прохождении практики;
- разработать, согласовать и выдать студентам индивидуальные задания;
- обеспечить прохождение практики студентами и строгое соответствие ее учебным планам и программам;
- осуществлять контроль за обеспечением нормальных условий труда и быта студентов, контролировать проведение со студентами обязательных инструктажей по охране труда;
- контролировать выполнение практикантами правил внутреннего распорядка;
- в установленные сроки организовать и лично участвовать в работе комиссии по приему зачетов по практике с выставлением оценок за практику и оформлением зачетных ведомостей.

В зависимости от сложности поставленной задачи возможно получение полного ее решения в виде программы для ЭВМ или математических формул, чтобы в дальнейшем перейти к аналогичным задачам и обобщениям данной задачи. В сложных случаях результатом практики является выполнение промежуточных этапов решения, включающих в себя разработку алгоритмов, состава и структур баз данных или математической модели. В любом случае, при решении задач математического моделирования к концу периода учебной практики студенты должны, как минимум, завершить построение математической модели изучаемого явления. Таким образом, они должны иметь математическую формулировку задачи с учетом возможности получить ее решение с помощью доступных средств вычислительной техники, имеющихся исходных экспериментальных данных и возможности проверки адекватности модели путем сравнения с экспериментом.

При исследовании проблем разработки информационных систем, задач автоматизации программирования, разработке Web-серверов студенты должны в период учебной практики разработать состав и структуру представления обрабатываемых данных или разобраться в предложенном составе и структуре; завершить, в основном, алгоритмы обработки данных в виде укрупненных блок-схем или диаграмм; разработать пользовательский интерфейс. Детализация алгоритмов происходит в период дальнейших исследований студентов.

Кроме продолжения научной работы студентов основными задачами практики являются: закрепление теоретических и практических знаний, полученных студентами в процессе обучения и приобретение навыков в математической постановке задачи и построении математических моделей или информационно-логических и физических моделей данных.

Задания на практику носят для каждого студента индивидуальный характер. В том случае, когда над одной и той же проблемой работает группа студентов (2-3 человека), допускается формулировка общего задания с обязательной конкретизацией работы для каждого студента.

Кроме этого, во время прохождения практики студенты

- знакомятся с организацией подразделения, в котором проходят практику и работ, ведущихся в данном подразделении;
- выясняют важность поставленной задачи и ее связь с другими исследованиями, проводимыми в подразделении, место своей задачи в структуре, создаваемой ими или модифицируемой информационной системе;

- изучают рабочие материалы исследований по своей проблеме;
- изучают литературу по теме своей работы;
- составляют отчет по результатам практики.

Во время прохождения практики студенты обязаны:

- проявить инициативу и самостоятельность, приобрести максимум практических навыков;
- подчиняться всем правилам внутреннего распорядка, действующего в организации, показывать пример трудовой дисциплины и исполнительности;
- изучить правила эксплуатации оборудования, техники безопасности и охраны труда, другие условия работ;
- нести ответственность за выполняемую работу и ее результаты наравне со штатными работниками.

10 Формы промежуточной аттестации (по итогам практики)

В ходе прохождения практики еженедельно по установленному графику происходит отчет студента перед руководителем от кафедры о проделанной работе.

На основании проделанной работы студент в конце практики составляет отчет. Оформление и содержание отчета должны соответствовать приложению В. Титульный лист отчета представлен в приложении А.

Отчет о практике студент защищает в комиссии, назначаемой заведующим кафедрой, в состав которой может входить представитель базы практики. Помимо отчета, в комиссию должен быть представлен отзыв руководителя от предприятия о работе студента. Контрольные вопросы при защите практики задаются по теме практики и являются индивидуальными для каждой темы и каждого студента.

Результатом защиты отчета является зачет с отметкой.

Оценка по практике проставляется в соответствии с Положением о модульно-рейтинговой системе квалитетрии учебной деятельности студентов, приравнивается к оценкам по теоретическому обучению, учитывается при подведении итогов общей успеваемости студентов.

В случае несвоевременной защиты практики оценка за практику снижается, причем, через три месяца после окончания практики прием отчетов по практике прекращается и практика считается не выполненной.

При защите отчета руководитель от кафедры оценивает перспективность данной работы студента. Студент, не сдавший отчет по практике, повторно направляется на практику или ставится вопрос перед ректором института об отчислении из института.

В отдельных случаях, связанных с болезнью студента, отпуском научного руководителя допускается перенос сроков практики. Перенос оформляется распоряжением по факультету на основании личного заявления студента с согласия научного руководителя и заведующего кафедрой. В случае переноса сроков практики, не связанного с болезнью студента, практика должна быть завершена не позднее 15 сентября.

11.Фонд оценочных средств для проведения промежуточной аттестации обучающихся по практике

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

Код контролируемой компетенции	Этап формирования компетенции	Способ оценивания	Оценочное средство
ПК-3 выбор рациональных информационных систем и информационно-коммуникативных технологий решения для управления бизнесом	Начальный	письменный отчет; защита отчета; зачет с оценкой	Комплект контролирующих материалов и иных заданий для защиты отчета о практике
ПК-5 проведение обследования деятельности и ИТ-инфраструктуры предприятий	Начальный	письменный отчет; защита отчета; зачет с оценкой	Комплект контролирующих материалов и иных заданий для защиты отчета о практике
ПК-13 умение проектировать и внедрять компоненты ИТ-инфраструктуры предприятия, обеспечивающие достижение стратегических целей и поддержку бизнес-процессов	Начальный	письменный отчет; защита отчета; зачет с оценкой	Комплект контролирующих материалов и иных заданий для защиты отчета о практике

11.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Показатели оценивания компетенций представлены в разделе «Компетенции обучающегося, формируемые в результате прохождения практики» программы практики с декомпозицией: знать, уметь, владеть.»

При оценивании сформированности компетенций по практике используется 100-балльная шкала.

Критерий	Оценка по 100-балльной шкале	Оценка по традиционной шкале
При защите отчета студент показал глубокие знания вопросов темы, свободно оперировал данными исследования и внес обоснованные предложения. Студент правильно и грамотно ответил на все поставленные вопросы. Практикант получил положительный отзыв от руководителя практики. Отчет в полном объеме соответствует заданию на практику.	75-100	<i>Отлично</i>
При ее защите отчета студент показал знания вопросов темы, оперировал данными исследования, внес обоснованные предложения. В отчете были допущены ошибки, которые носят несущественный характер. Практикант получил положительный отзыв от руководителя практики.	50-74	<i>Хорошо</i>
Отчет по практике имеет поверхностный анализ собранного материала, нечеткую последовательность изложения материала. Студент при защите отчета по практике не дал полных и аргументированных ответов на заданные вопросы. В отзыве руководителя практики имеются существенные замечания.	25-49	<i>Удовлетворительно</i>
Отчет по практике не имеет	<25	<i>Неудовлетвор</i>

детализированного анализа собранного материала и не отвечает требованиям, изложенным в программе практики. Студент затрудняется ответить на поставленные вопросы или допускает в ответах принципиальные ошибки. В полученной характеристике от руководителя практики имеются существенные критические замечания.		<i>ительно</i>
--	--	----------------

11.3 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

1. Структуры данных. Стек. Очередь. Список. Примеры алгоритмов на C#. (ПК-13)
2. Методы сортировки. Примеры алгоритмов на C#. (ПК-13)
3. Поиск данных в массивах и в файлах. Примеры алгоритмов (символьная строка, подстрока). (ПК-13)
4. Общеязыковые конструкции языков C#. Циклы, условные операторы. (ПК-13)
5. Классы и абстрагирование данных. Отношения между классами. Примеры на C#. Сопоставление с UML-представлением на диаграмме классов. (ПК-13)
6. Доступ к базам данных средствами C#. Примеры. (ПК-5)
8. Технология Microsoft .NET. Особенности, основные положения и характеристики.
9. Объектно-ориентированное программирование средствами C#. Пример наследования/инкапсуляции/полиморфизма. (ПК-3)
10. Создание Windows-приложений средствами C#. Пример. (ПК-3)
11. Виртуальные функции и полиморфизм. Реализация на C#. Примеры (консольное приложение). (ПК-3)
12. Элементы контроля. Примеры на C#. (ПК-3)
13. Понятие интерфейса. Интерфейс и его реализация. Примеры на диаграмме классов. (ПК-3)

11.4 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и(или) опыта деятельности, характеризующих этапы формирования компетенций, определены локальными нормативными актами СТО АлтГТУ 12100-2015 Фонд оценочных средств образовательной программы. Общие сведения, СТО АлтГТУ 12330-2014 Практика. Общие требования к организации, проведению и программе практики, СТО АлтГТУ 12560-2011 Текущий контроль успеваемости и промежуточной аттестации студентов и СМК ОПД-01-19-2008 Положение о модульно-рейтинговой системе квалитметрии учебной деятельности студентов, а также соответствующими разделами настоящей программы практики.

12 Перечень учебной литературы и ресурсов сети "Интернет"

Литература

Основная литература

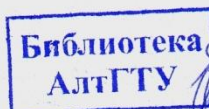
1. Золотов С.Ю. Проектирование информационных систем: учебное пособие / С.Ю. Золотов. - Томск: Эль Контент, 2013. - 88 с. Режим доступа: http://biblioclub.ru/index.php?page=book_view&book_id=208706
2. Мейер Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер Б. - Национальный Открытый Университет «ИНТУИТ», 2016. - 286 с. Режим доступа: https://biblioclub.ru/index.php?page=book_view_red&book_id=429034

Библиотека
АлтГТУ

Дополнительная литература

3. Пятковский О.И. Практикум по дисциплине «Проектирование информационных систем» Часть 1. Предпроектная стадия процесса проектирования экономических информационных систем : учебное пособие / О.И. Пятковский, М.В. Гунер; Алт. гос. техн. ун-т им. И. И. Ползунова. – Барнаул: кафедра ИСЭ, АлтГТУ, 2015. – 103 с. - Источник: Электронная библиотека образовательных ресурсов АлтГТУ. Режим доступа: <http://new.elib.altstu.ru/eum/download/ise/uploads/guner-m-v-ise-561e710ef3539.pdf>
4. Пятковский О.И. Практикум по дисциплине «Проектирование информационных систем» Часть 2. Техно-рабочее проектирование: учебное пособие / О.И. Пятковский, М.В. Гунер; Алт. гос. техн. ун-т им. И. И. Ползунова. – Барнаул: кафедра ИСЭ, АлтГТУ, 2015. – 113 с. - Источник: Электронная библиотека образовательных ресурсов АлтГТУ. Режим доступа: <http://new.elib.altstu.ru/eum/download/ise/uploads/guner-m-v-ise-561e7147d48e7.pdf>

5. Володин В.В. Управление проектом: учебное пособие / В.В. Володин, Ф.Б. Лобанов, Т.В. Алексеева и др. - М.: Московский финансово-промышленный университет «Синергия», 2013. - 96 с. – Источник: Университетская библиотека ONLINE. Режим доступа: <http://biblioclub.ru/index.php?page=book&id=252967>
6. Балдин, К.В. Информационные системы в экономике. [Электронный ресурс] / К.В. Балдин, В.Б. Уткин. — Электрон. дан. — М. : Дашков и К, 2013. — 395 с. — Режим доступа: <http://e.lanbook.com/book/56213>.



Интернет-ресурсы

16. Информационно-правовой сервер «Гарант» –<http://www.garant.ru/>
17. Информационно-правовой сервер «КонсультантПлюс» – <http://www.consultant.ru/>
18. Издательство «Открытые системы» - <http://www.osp.ru/>;
19. Корпорация «Университетские сети знаний» UNICOR - <http://www.rc.ac.ru>.
20. Библиотека учебников, руководств и текстов по программированию - <http://www.codenet.ru/>
21. Законодательство связанное с Интернет-деятельностью и информационной безопасностью <http://www.internet-law.ru/>

13. Перечень информационных технологий

Программное-обеспечение

- Прикладной пакет MS Office.
- Операционные системы семейства Windows.
- программные продукты в соответствие с темой практики;

Информационно-справочные системы

- Информационно-справочные (СПС «Гарант», 1С:ИТС);

14. Материально-техническая база.

- аудитории, оснащенные меловой или маркерной аудиторной доской, партами, кафедрами - для проведения обзорных занятий;
- лекционная аудитория (оборудованная видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном и имеющая выход в сеть Интернет), для организации консультаций;

- рабочее место практиканта, оборудованное для выполнения заданий практики персональным компьютером, рабочие станции, объединенные в локальные сети с выходом в Интернет, оснащенные программно-методическими комплексами для получения знаний и приобретения навыков решения задач

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика.

Автор

Д.Д. Барышев Д.Д. Барышев

Программа рассмотрена и одобрена на заседании кафедры
«Информационные системы в экономике» 22 декабря 2016 г, протокол № 4.

Заведующий кафедрой ИСЭ

О.И. Пятковский О.И. Пятковский

И.О. начальника отдела практик и трудоустройства

И. Г. Таран И. Г. Таран

Программа рассмотрена и одобрена на заседании факультета
информационных технологий 22 декабря 2016 г, протокол № 4.

Декан ФИТ

Е.А. Зрюмов Е.А. Зрюмов

ПРИЛОЖЕНИЕ А

Форма титульного листа отчета о практике

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Алтайский государственный технический
университет им. И. И. Ползунова»

Факультет

информационных технологий

Кафедра

информационных систем в экономике

Отчет защищен с оценкой _____

“ _____ ” _____ 20__ г.

Руководитель от вуза

_____ / _____ /

подпись

Ф. И. О.

ОТЧЕТ

Об учебной практике

общая формулировка задания в

наименование организации

Студент гр. БИ-_____ Иванов И.А._____

Руководитель от университета _____ *подпись*

Ф. И. О.

Барнаул 2017

ПРИЛОЖЕНИЕ Б

ФГБОУ ВО «Алтайский государственный технический университет
им. И. И. Ползунова»

Кафедра _____

Индивидуальное задание

на _____
(вид, тип и содержательная характеристика практики по УП)

студенту _____ курса _____ группы _____
(Ф.И.О.)

Профильная организация _____
(наименование)

Сроки практики _____
(по приказу АлтГТУ)

Тема _____

Рабочий график (план) проведения практики:

№ п / п	Содержание раздела (этапа) практики	Сроки выполнен ия	Планируемые результаты практики

Руководитель практики от университета _____

(подпись) (Ф.И.О., должность)

Руководитель практики от
профильной организации _____
(подпись) (Ф.И.О., должность)

Задание принял к исполнению _____
(подпись) (Ф.И.О.)

ПРИЛОЖЕНИЕ В

Содержание отчета

Отчет о практике должен содержать:

- титульный лист, оформленный согласно приложению;
- задание и календарный план выполнения практики, подписанные руководителем практики, оформленный согласно приложению Б;
- введение;
- развернутую постановку задачи в виде технического задания;
- проект программного продукта;
- описание программного продукта;
- заключение;
- источники информации;
- приложения.

Введение должно содержать общие сведения о практике и краткую характеристику базы практики (если практика проводится не на базе вуза).

Раздел «Техническое задание» содержит подробное описание функциональности разрабатываемого программного обеспечения в виде технического задания.

Раздел «Проект программного продукта» может состоять из нескольких глав и содержит в себе подробное описание результатов проектирования, выполненного студентом. Проект обязательно содержит описание используемых структур данных и файлов, описание модульной структуры программного обеспечения и проект графического интерфейса. Дополнительно этот раздел может содержать описание нетривиальных алгоритмов. Приводятся необходимые иллюстрации.

Раздел «Описание программного продукта» содержит краткие сведения об используемых методах и средствах программирования, а также описание спецификаций всех разработанных студентом элементов кода (типы данных, классы, функции и методы).

В разделе «Заключение» студент должен:

- кратко изложить результаты и перспективы развития разработанного в процессе практики программного продукта;
- отметить недостатки действующей системы (объектов, процессов) и конкретные пути ее улучшения и замены;
- проявить универсальные и профессиональные компетенции.

Обязательные приложения к отчету:

- код программного продукта;
- руководство пользователя;

Дополнительные приложения к продукту:

- план и результаты модульного тестирования, если оно проводилось;
- описание системных требований и дополнительных библиотек, если они отличаются от стандартных.

Текст отчета оформляется в виде принтерных распечаток на сброшюрованных листах формата А4 (210х297мм). При оформлении отчета необходимо соблюдать требования ГОСТ 2.105, ГОСТ 2.106, ГОСТ 3.1127, ГОСТ 3.1123, ГОСТ 3.1407, ГОСТ 8.417, ГОСТ 7.1, СТП 12 570-2006 СТАНДАРТ ПРЕДПРИЯТИЯ. Система менеджмента качества. Образовательный стандарт высшего профессионального образования АлтГТУ. ОБЩИЕ ТРЕБОВАНИЯ К ТЕКСТОВЫМ, ГРАФИЧЕСКИМ И ПРОГРАММНЫМ ДОКУМЕНТАМ.

Общий объем отчета по учебной практике, как правило, должен соответствовать 20-40 страницам печатного текста (с приложениями).

ПРИЛОЖЕНИЕ Г

Варианты заданий на практику

1. Клиентское приложение для предварительной записи на сервис в салонах красоты или фитнес-клубах.
2. Сервис оперативных скидок / купонный сервис “15-минутное спецпредложение”.
3. Приложение для поиска и резервирования в аптечной сети.
4. Агрегаторскидок.
5. Приложения для мастера / начальника производственной смены - регистрация заказов, работ, выдачи/сдачи оснастки, изделий, завершение обработки.
8. Приложение для консультанта в зале. Известно, что как только консультант отходит от клиента для проверки наличия товара или уточнения цены - клиент “срывается”. Легкое приложение на телефоне позволит консультанту непрерывно вести покупателя до совершения покупки.
9. Прием заказов в ресторане/кафе официантом.
10. Ввод первичной информации сразу на месте возникновения. Например, расходные кассовые ордера с фотографией чека и выгрузкой в центральную базу.
11. Контроль работы удаленных сотрудников. Например, представитель компании должен иметь возможность видеть свой маршрут на текущий день, а после выполнения задачи - сделать фото, прикрепить к документу и отправить в центральный офис. Плюс фиксация реального начала и конца рабочего дня, расходов на дорогу и т.п.

12. Приложение для отдела логистики - прием/отгрузка товара по штрих-коду, быстрое извещение клиента по телефону/смс, динамический расчет маршрута доставки. А для клиентов - получение сведений о доставке по номеру накладой, мониторинг местоположения груза, отзывы и документооборот.

13. Учет произведенных услуг ИТ подразделением при выезде. Сотрудник видит свой план, со списком задач, клиентами, маршрутом. На выезде в приложение вносятся услуги и время. Информация выгружается в 1С, из 1С строятся счета-фактуры на оплату и тут же, до отъезда, высылаются клиенту.

14. Монитор поставщика. Товар поступает на реализацию, приложение выводит основную информацию по товару поставщика в магазине - продажи, остаток, рекомендуемое количество к заказу, существующий заказ магазина.

15. Приложения для помощи удаленным сотрудникам. Часто сотрудники не могут достоверно диагностировать истинную причину проблемы. Они делают фото/видео, прикрепляют к заявке, и инженер поддержки может удаленно подсказать, как решить проблему. Все накапливается в базе отдела поддержки, со всеми вложениями, геопозициями.

ПРИЛОЖЕНИЕ Д

Методические указания по выполнению работы.

Задание. Разработать программу, имитирующую движение городского транспорта. Время процесса дискретно. В целях упрощения предполагается, что улицы города пересекаются под прямыми углами. Транспортные средства различных моделей двигаются с различными скоростями, меняя направление движения на перекрестках случайным образом. Любое транспортное средство имеет следующие характеристики – максимальную скорость, уровень загрязнения (г/м, считается, что за один шаг по времени загрязнение полностью разлагается), уровень шума (в децибелах), обратно пропорциональный расстоянию до транспортного средства (считается, что шум распространяется только вдоль проезжей части). Транспортные средства могут иметь внутренний или внешний (например, троллейбусы) источник энергии. Все транспортные средства делятся на пассажирские и грузовые. Пассажирские средства имеют максимальное количество пассажирских мест, грузовые – максимальный вес перевозимого груза. Транспортное средство может производить обгон, если оно имеет внутренний источник энергии, иначе оно должно двигаться со скоростью впереди идущей машины. Машины могут случайным образом покидать город, а также въезжать в город. Количество пассажиров, вес груза и начальная скорость задается случайно, так чтобы они не превышали свои максимальные значения. На каждом шаге по времени пользователь может вывести следующую информацию:

- Состояние любого транспортного средства;
- уровень шума в каждой точке проезжей части;
- уровень загрязнения в каждой точке проезжей части;
- количество пассажиров на любой улице;
- вес перевозимого груза на любой улице;
- количество пассажирских транспортных средств;
- количество грузовых транспортных средств;
- количество транспортных средств, не загрязняющих окружающую среду.

Запустите Microsoft Visual Studio 2008.

После запуска откроется стартовая страница – рисунок 1.

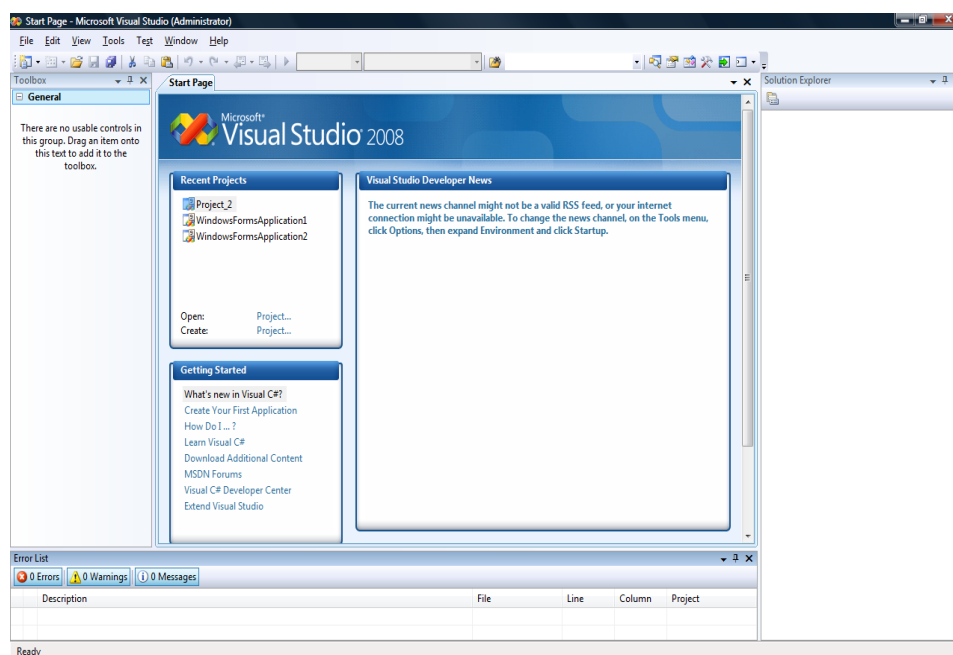


Рисунок 1 – Стартовая страница Microsoft Visual Studio 2008

Для создания нового проекта, выберите в меню File - пункт New, затем Project.

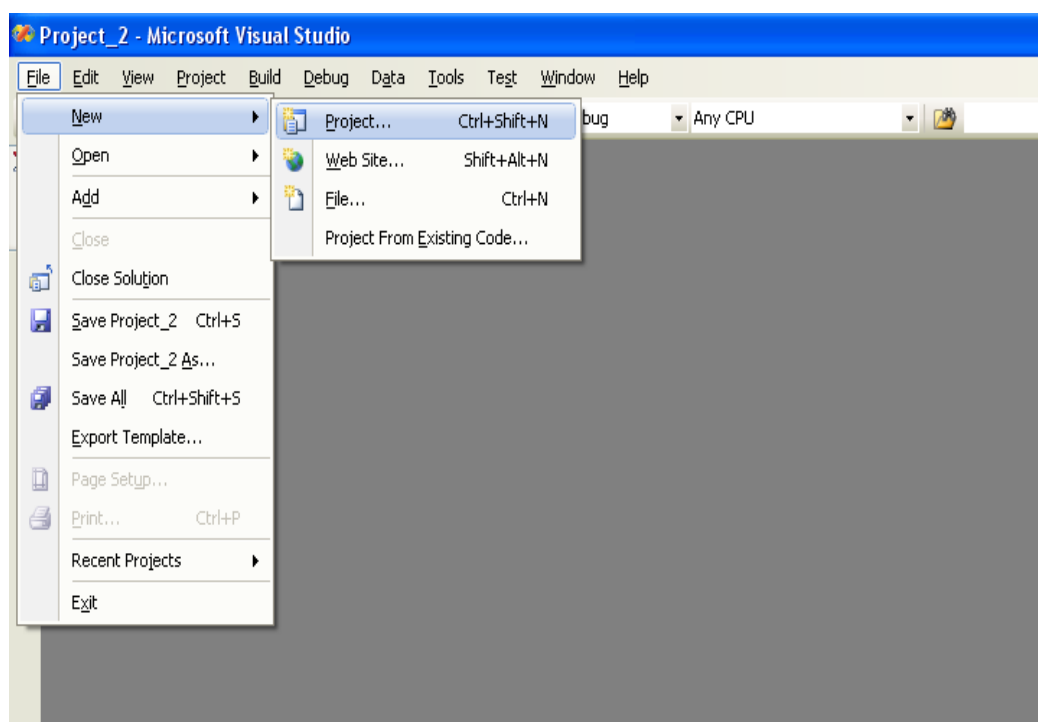


Рисунок 2 - Создание проекта с помощью меню

Так как вам необходимо создать проект, выберите WindowsFormsApplication. Также желательно изменить имя проекта.

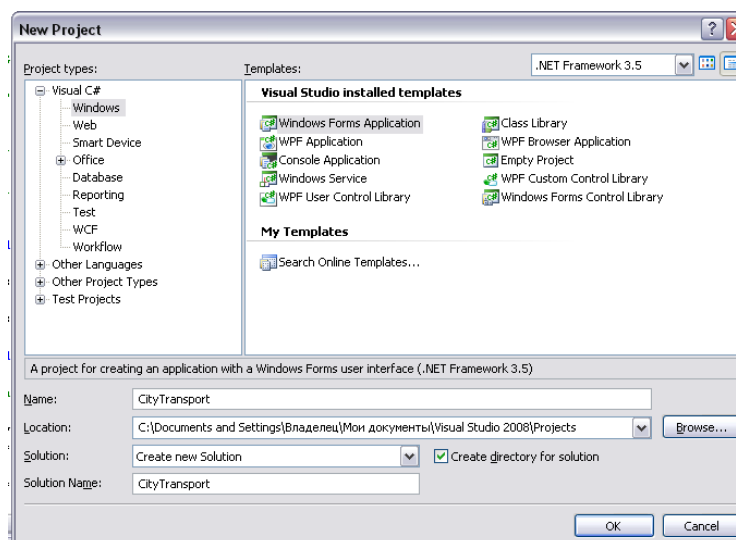



Рисунок 3 – Создание проекта WindowsFormsApplication

Для удобства работы с Microsoft Visual Studio 2008 необходимо добавить некоторые окна, работа с которыми вам предстоит. Для этого в меню View выберите пункты TaskList, ToolBox (для закрепления этого окна нажмите второй значок из ) и PropertiesWindow, а также ErrorList.

Новый проект создан. После создания проекта открывается вкладка с редактированием формы.

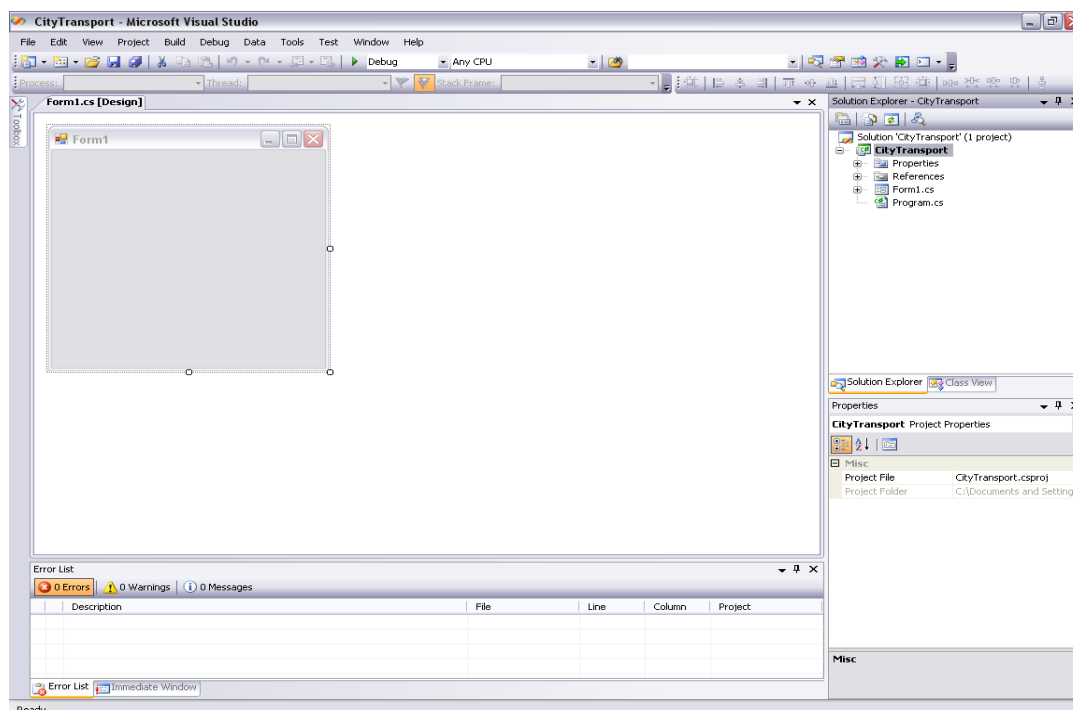


Рисунок 4 – Окно с редактированием формы

Если вам необходимо создать кнопки на форме, необходимо выбрать пункт Button и создать кнопку необходимого размера.

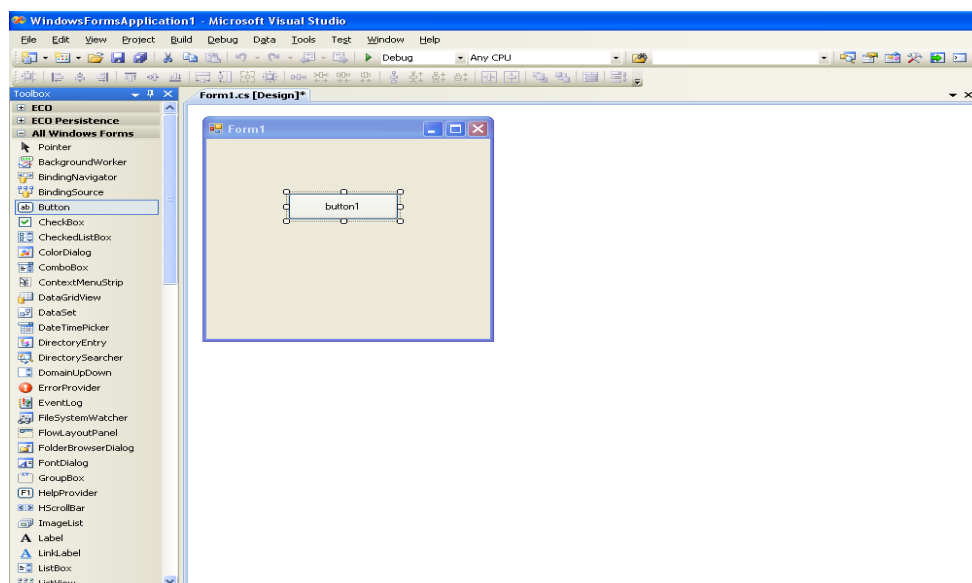


Рисунок 5 – Создание кнопки на форме

Для того, чтобы изменить текст, отображаемый на кнопке, изменить его в окне Properties, строка Text.

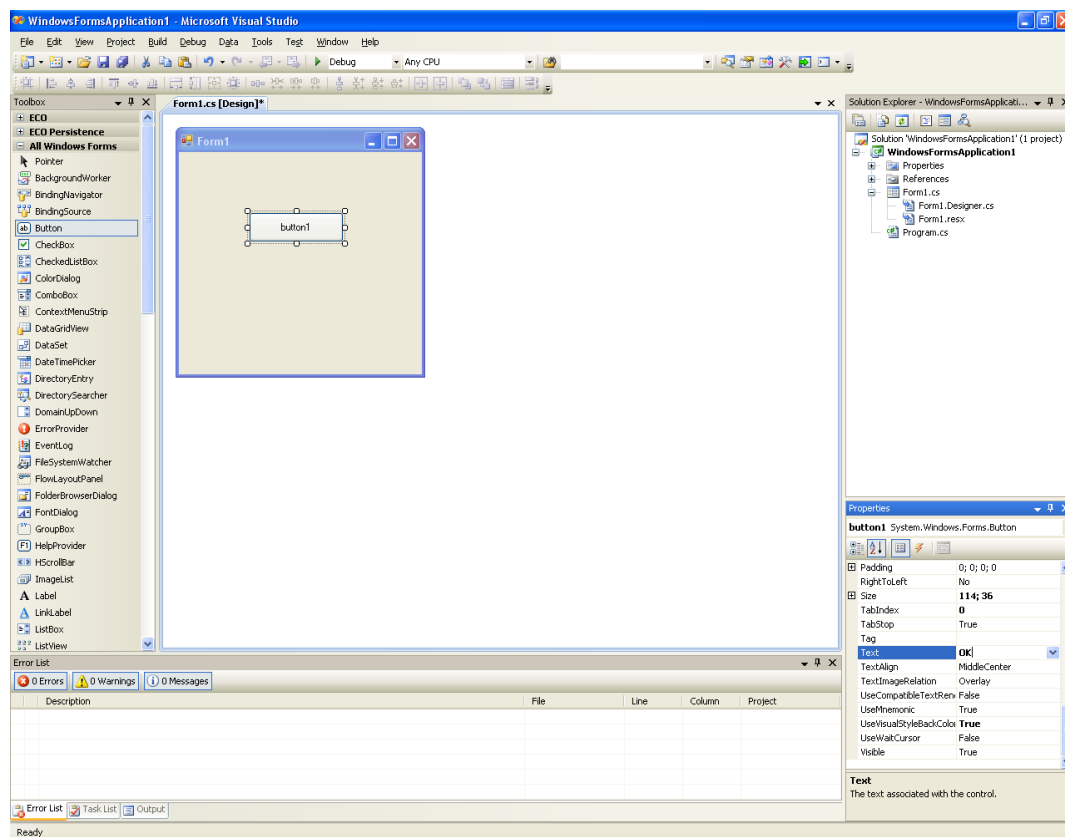


Рисунок 6 – Изменение текста, отображаемого на кнопке

После редактирования формы необходимо создать класс, который будет базовым для вашей программы. При определении класса объявляются данные, которые он содержит, и код, работающий с этими данными. Самые простые классы могут содержать только код или только данные, но в реальных программах классы включают обе эти составляющие. Данные содержатся в переменных экземпляра, которые определены классом, а код содержится в методах. При создании (определении) класса вначале указывается ключевое слово `class`.

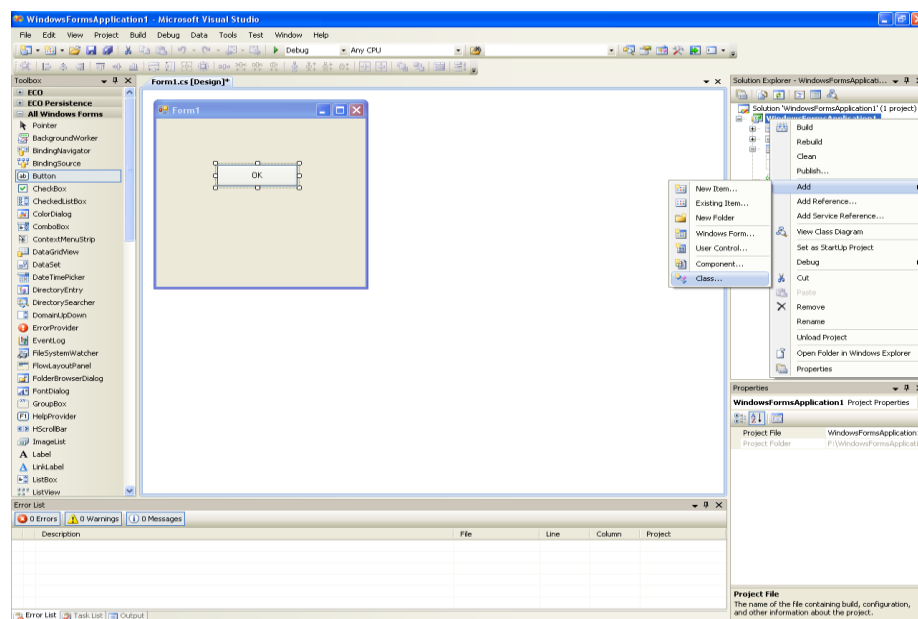


Рисунок 7 – Добавление класса из окна SolutionExplorer

Также это можно сделать, открыв меню Project:

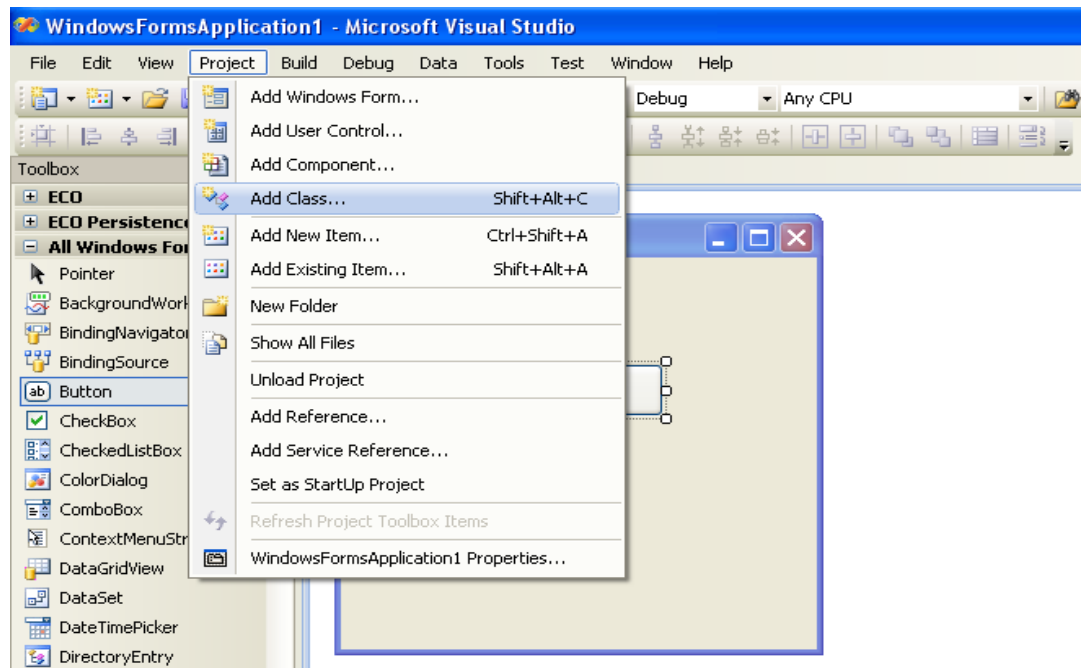


Рисунок 8 – Добавление класса из меню Project

Затем нужно изменить имя класса.

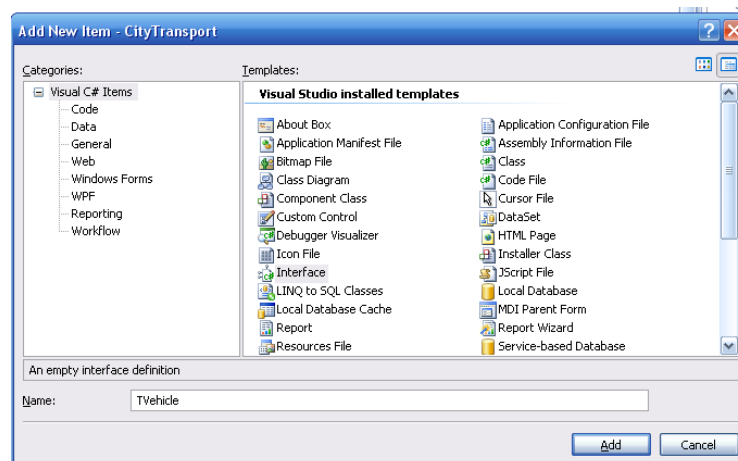


Рисунок 9 – Изменение имени добавляемого класса

Откроется окно с кодом класса.

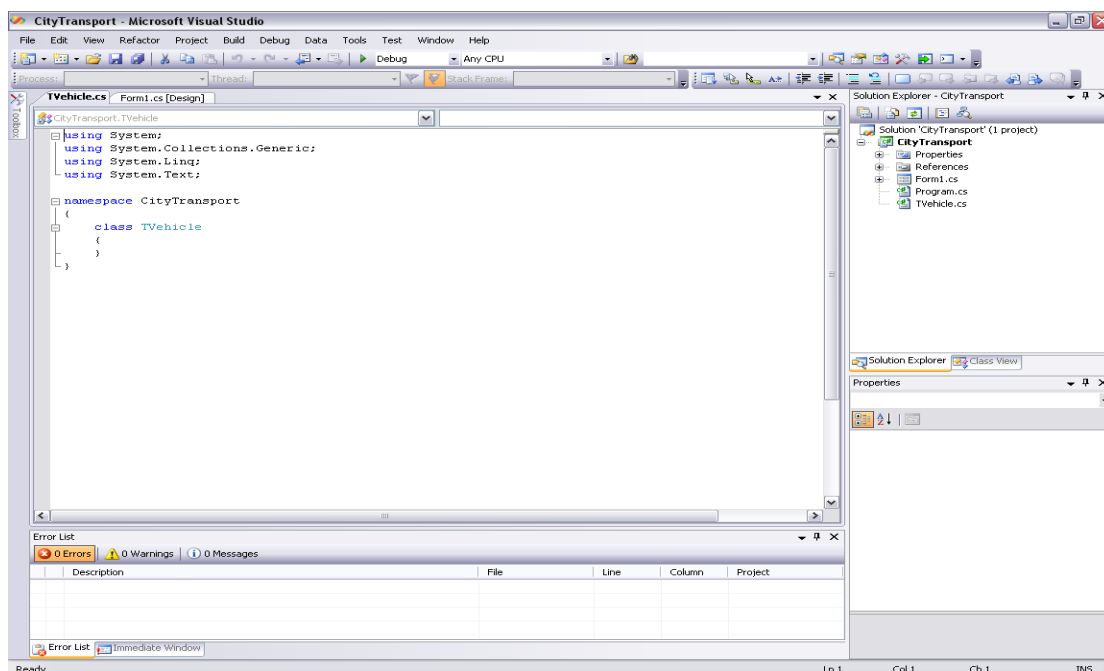


Рисунок 10 – Вкладка с кодом класса

Для редактирования класса удобно использовать диаграмму класса. Для ее создания необходимо открыть пункт ViewClassDiagram, щелкнув правой кнопкой мыши на названии класса.

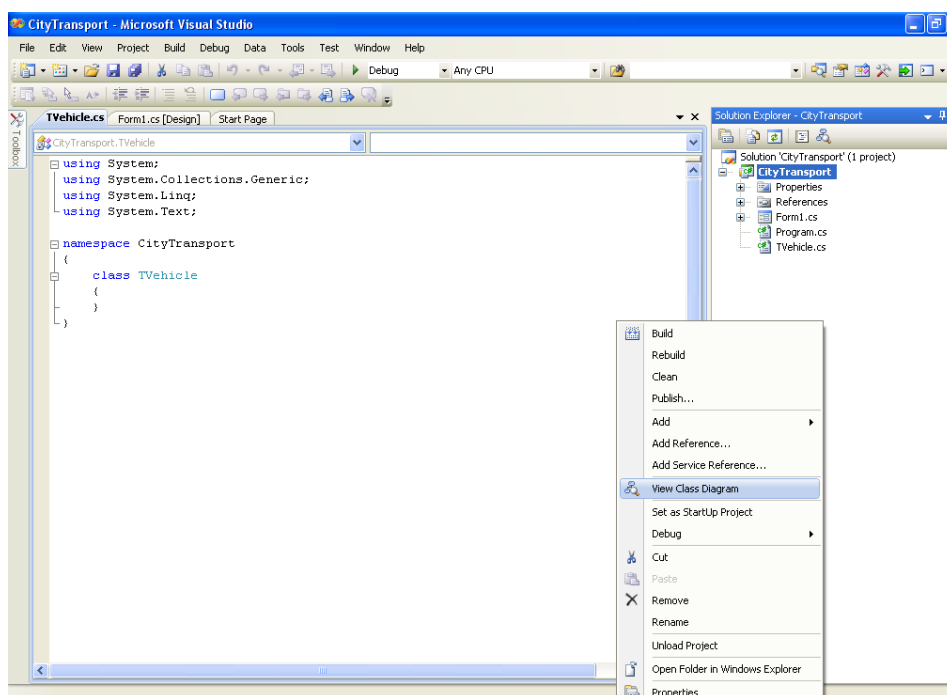


Рисунок 11 – Создание диаграммы класса

Для добавления методов, полей или конструктор можно использовать два способа.

Первый – щелкнуть правой кнопкой мыши и выбрать пункт Add.

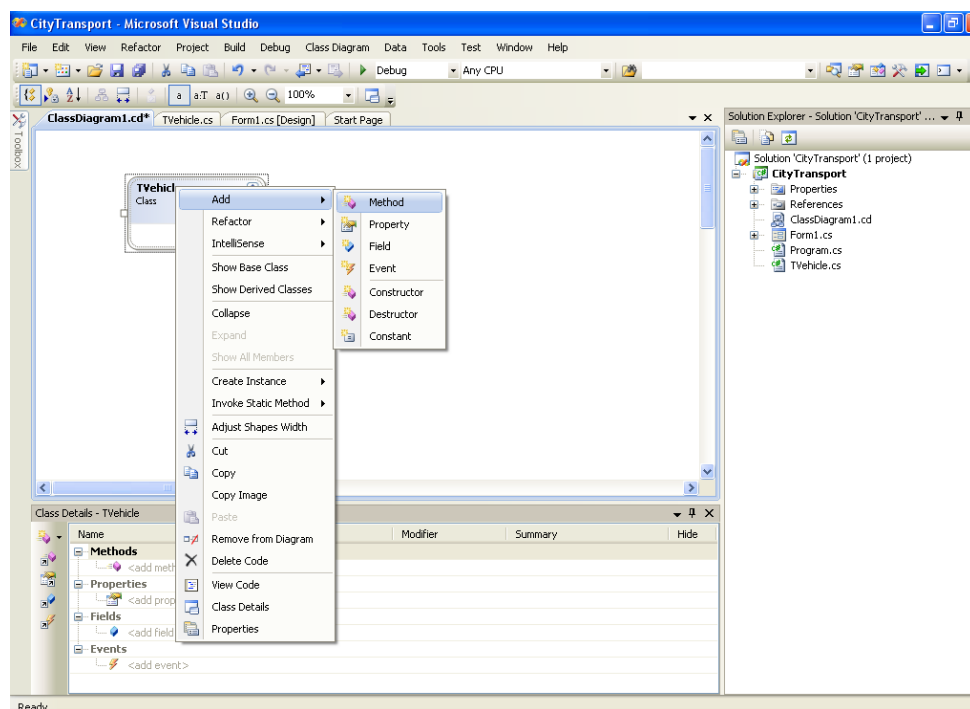


Рисунок 12 – Добавление методов, полей и конструкторов

Второй – в окне ClassDetails. Для открытия этого окна необходимо щелкнуть правой кнопкой мыши по блоку класса и выбрать пункт ClassDetails.

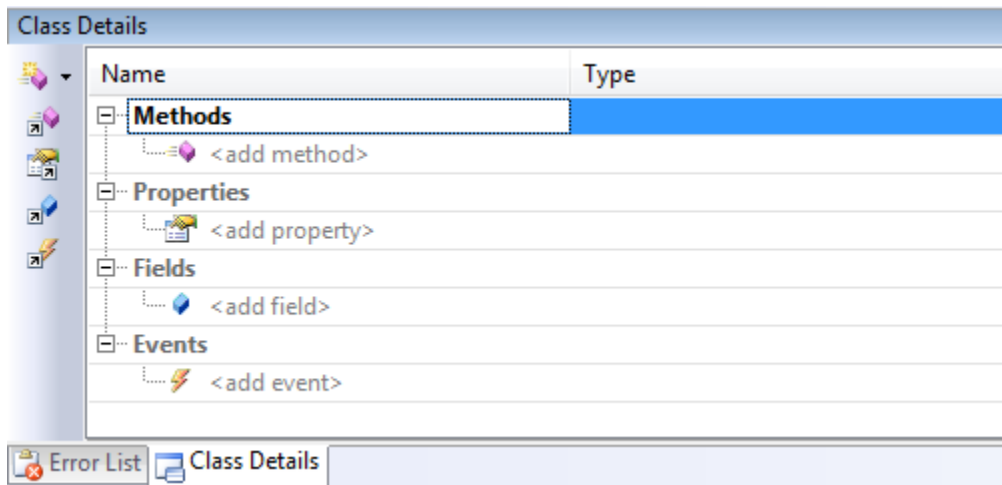


Рисунок 13 – Добавление методов, полей и конструкторов с помощью ClassDetails

В соответствии с заданием создадим класс **TVehicle** с полями:

```
MaxSpeed; //максимальная скорость

PolutionLevel; //уровень загрязнения

NoiseLevel; //уровень шума

EnergySource; // источник энергии 1- внутренний, 0- внешний

Speed, NormalSpeed; // текущая скорость и скорость движения без помех

Direction; // направление движения (в радианах, 0-восток, pi -
запад, pi/2 - север)

X, Y; //координаты

Street; // улица

и методами:

TVehicle // конструктор

Type //тип т/с

CanObgon//возможность обгона
```

После редактирования окно ClassDetails будет выглядеть следующим образом:

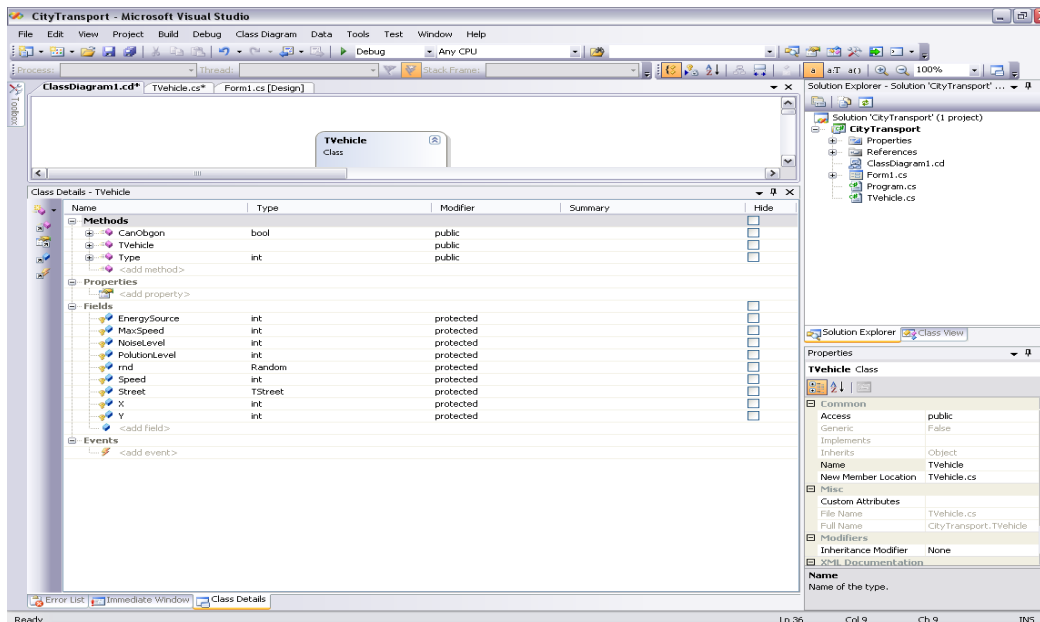


Рисунок 14 – Окно ClassDetails после редактирования класса

Методы и поля, записанные в ClassDetails, будут отображены и в диаграмме класса. Также они будут оформлены и в коде класса.

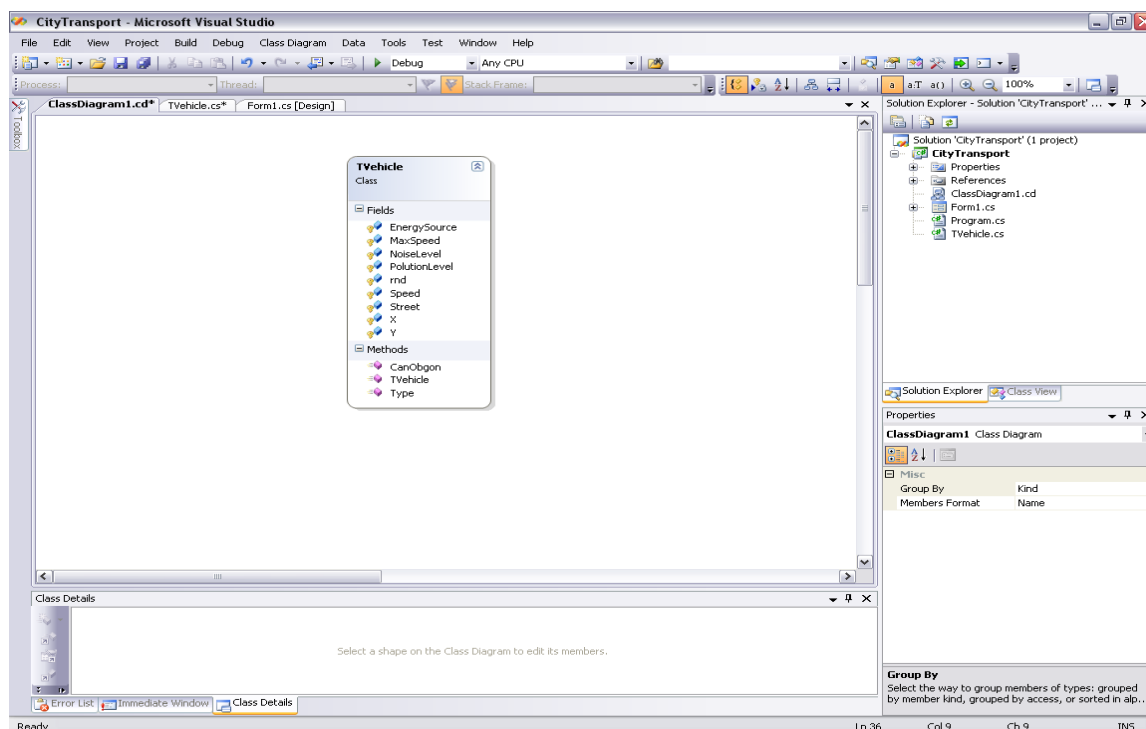


Рисунок 15 – Диаграмма класса после его редактирования

Класс TVehicle (базовый класс для всего транспорта)

```
public class TVehicle
{
    protected int MaxSpeed; //максимальная скорость
    protected int PollutionLevel; //уровень загрязнения
    protected int NoiseLevel; //уровень шума
    protected int EnergySource; // источник энергии 1- внутренний, 0- внешний
    protected int Speed; //скорость
    protected int X, Y; //координаты
    protected TStreet Street; //улица
    protected Random rnd = new Random();
    public TVehicle() //конструктор класса
    {
    }

    //методы класса
    public int Type() {} //тип т/с
    public bool CanObgon() {} //возможность обгона
}
```

НАСЛЕДОВАНИЕ – свойство, с помощью которого один объект может приобретать свойства другого. Без использования наследования каждый объект должен явно определять все свои характеристики. Используя наследование, объект должен определить только те качества, которые делают его уникальным в пределах своего класса.

Для создания классов наследников необходимо создать новые классы и в коде указать родительский класс, например,

```
public class TCivilTransport : TVehicle
```

На диаграмме класс-наследник отобразится после выбора пункта ShowDerivedClasses.

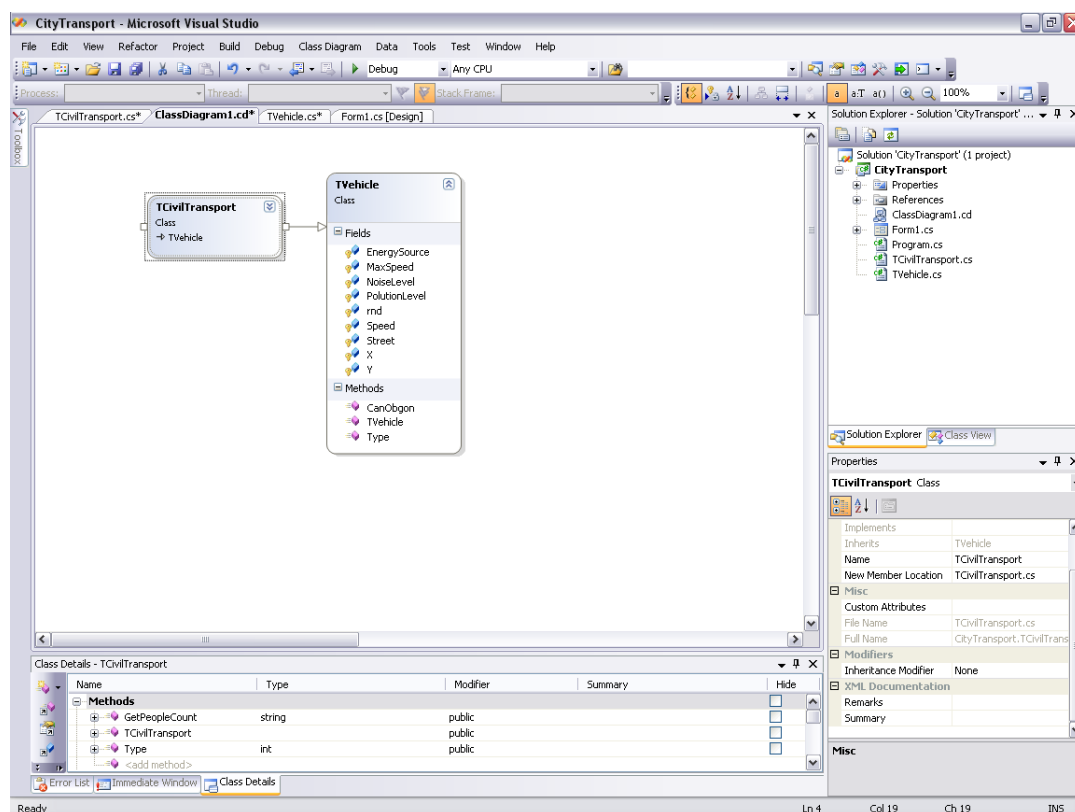


Рисунок 16 – Отображение классов-наследников

Редактируются классы-наследники аналогично. Однако могут возникнуть проблемы с объявлением конструктора, если в базовом классе он имеет свои параметры. Для решения этой задачи можно в базовом классе объявить еще один конструктор, но уже без параметров. Тогда у классов-наследников могут быть свои конструкторы со своими параметрами.

Класс TCivilTransport (Пассажирский транспорт)

Поля:

```
MaxPeopleCount; // кол-во пассажирских мест  
PeopleCount;    // кол-во пассажиров
```

Методы:

```
TCivilTransport //конструктор класса  
GetPeopleCount () //возвращает кол-во пассажиров  
Type () //возвращает тип т/с
```

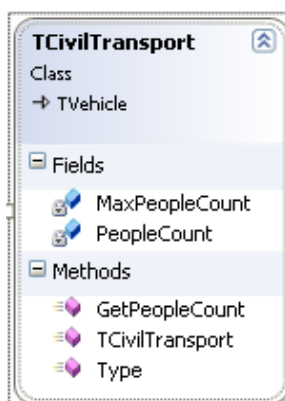


Рисунок 17 – Диаграмма класса TCivilTransport

```
public class TCivilTransport : TVehicle // наследуемый класс от TVehicle  
{  
    private int MaxPeopleCount; // переменная кол-во пассажирских мест  
    private int PeopleCount;    // переменная кол-во пассажиров  
    public TCivilTransport(int aMaxPeopleCount) //конструктор класса TCivilTransport  
    {  
        MaxPeopleCount = aMaxPeopleCount;  
        PeopleCount = rnd.Next(1, MaxPeopleCount); //кол-во пассажиров равно случайному числу от 1 до кол-ва пассажирских мест  
    }  
}
```



```

//методы класса

public string GetPeopleCount () //возвращает кол-во пассажиров

public int Type() //возвращает тип т/с
    }

```

Аналогично создадим наследуемый класс TLorryTransport и определим его методы и поля.

Класс TLorryTransport (Грузовой транспорт)

Поля:

```

MaxGruzMassa;    // переменная грузоподъемность

GruzMassa;       // переменная масса груза

```

Методы:

```

TLorryTransport(int aMaxGruzMassa) //конструктор класса TLorryTransport

GetGruzMassa () //возвращает массу груза

Type() //возвращает тип т/с

```

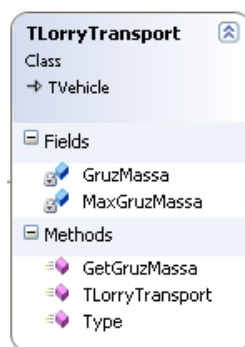


Рисунок 18 – Диаграмма класса TLorryTransport

```

public class TLorryTransport : TVehicle //
    грузовой транспортные средства, наследуемый класс от TVehicle

{

    private int MaxGruzMassa;    // переменная грузоподъемность

```

```

private int GruzMassa;          // переменная масса груза

public TlorryTransport(int aMaxGruzMassa) // конструктор класса TlorryTransport
{
    MaxGruzMassa = aMaxGruzMassa;

    GruzMassa = rnd.Next(1,
MaxGruzMassa); // масса перевозимого груза равна случайному числу от 1
до грузоподъемности

}

// методы класса

public string GetGruzMassa () // возвращает массу груза

public override int Type () // возвращает тип т/с
}

```

Создадим еще один класс **TStreet**(Улицы)

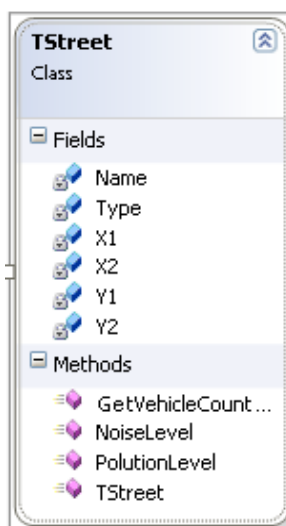


Рисунок 19 – Диаграмма класса TSreet

Класс TStreet (Улицы)

```

public class TStreet
{
    private int Type; // 1- вертикальная (улица), 2- горизонтальная (проспект)

    private int X1, X2, Y1, Y2; // координаты улицы

    private string Name; // название улицы

    public TStreet (string aName, int aX, int aY) // конструктор класса

```

```

        {
            Name = aName;

            X1 = aX; Y1 = aY;

            if (aX == 0) // проспект
            {
                Type = 2;

                X2 = 500; Y2 = Y1;
            }

            else// улица
            {
                Type = 1;

                X2 = X1; Y2 = 400;
            }
        }

// методы класса

public double NoiseLevel(int L) // уровень шума
{
    return 0;
}

public double PollutionLevel(int L) // уровень загрязнения
{
    return 0;
}

public int GetVehicleCountOnStreet(int aVehicleType) // кол-во т/с на улице
{
    return 0;
}
}

```

После редактирования классов, создания новых классов, не связанных с базовым, диаграмма может выглядеть следующим образом:

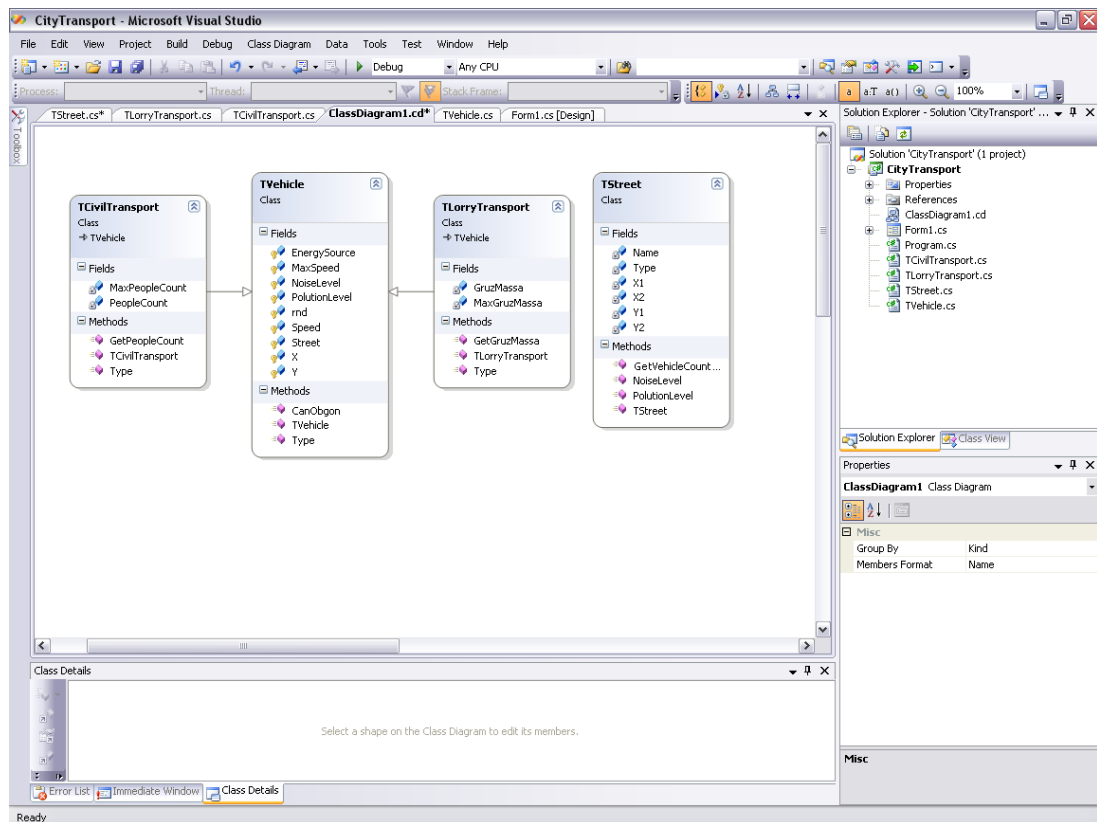


Рисунок 20 – Итоговый вариант диаграммы

Двойной щелчок по блоку класса переведет вас во вкладку с кодом программы этого класса.

После разработки базового класса TVehicle, производных классов TCityTransport, TLorryTransport реализуем их, добавив в базовый класс виртуальные методы.

КлассTVehicle

```
publicclassTVehicle
{
    protectedint MaxSpeed;//максимальнаяскорость
    protectedint PollutionLevel;//уровеньзагрязнения
    protectedint NoiseLevel;//уровеньшума
    protectedint EnergySource;    // источникэнергии 1- внутренний, 0- внешний
    protectedint Speed;//скорость
    protectedint X, Y;//координаты
```

```

protectedTStreet Street;//улица

protectedRandom rnd = newRandom();

public TVehicle()//конструкторкласса TVehicle
{
    EnergySource = rnd.Next(0, 2);//источник энергии - случайным образом выбирается
    0 или 1

    MaxSpeed = rnd.Next(1,60); //максимальная скорость - случайным
    образом выбирается от 1 до 60

    NoiseLevel = rnd.Next(1, 20); //уровень шума - случайным образом
    выбирается число от 1 до 20

    if (EnergySource == 0) //если источник энергии внешний,то уровень загрязнения
    равен 0

        PolutionLevel = 0;

    else PolutionLevel = rnd.Next(1, 10);//иначе уровень загрязнения выбирается
    случайным образом от 1 до 10

    Speed = rnd.Next(1,MaxSpeed); // скорость генерируется случайным
    образом от 1 до максимальной скорости

}

publicvirtualint Type () //виртуальный метод Type в базовом классе
{
    return 0;

}

publicbool CanObgon()// возможностьобгона
{
    returntrue;

}

}

```

КлассTCivilTransport

```

publicclassTCivilTransport: TVehicle// пассажирскиетранспортныесредства
{
    privateint MaxPeopleCount; // кол-вопассажирскихмест
    privateint PeopleCount;    // кол-вопассажиров

```

```

public TCivilTransport(int aMaxPeopleCount)
{
    MaxPeopleCount = aMaxPeopleCount;
    PeopleCount = rnd.Next(1, MaxPeopleCount);
}

public string GetPeopleCount ()
{
    string res;

    res = PeopleCount.ToString("F0");

    return res;
}

public override int Type() /*виртуальный метод, переопределенный в наследуемом классе*/
{
    return 1; // пассажирский ет/с
}
}

```

Класс TLorryTransport

```

public class TLorryTransport: TVehicle // грузовые транспортные средства
{
    private int MaxGruzMassa; // грузоподъемность
    private int GruzMassa; // масса груза

    public TLorryTransport(int aMaxGruzMassa)
    {

        MaxGruzMassa = aMaxGruzMassa;

        GruzMassa = rnd.Next(1, MaxGruzMassa);
    }
}

```

```

        }

publicstring GetGruzMassa ()
    {
        string res;

        res = GruzMassa.ToString("F0");

        return res;
    }

publicoverrideintType() /*виртуальныйметод, переопределенныйвнаследуемомклассе*/
{
    return 2; // грузовое транспортное средство
}
}

```

Также в программу были добавлены новые классы для дальнейшей работы с программой.

Класс Init

//класс для инициализации массивов транспортных средств и улиц

```

publicclassInit
{
    staticpublicTCivilTransport[] CivilTransportArray = newTCivilTransport[20];
    //массивпассажирскихсредств

    staticpublicTLorryTransport[] LorryTransportArray = newTLorryTransport[20];
    //массивгрузовыхсредств

    staticpublicTStreet[] StreetArray = newTStreet[4]; //массивулиц
    staticpublicTStreet[] ProspektArray = newTStreet[3]; //массивпроспектов

    publicList<TVehicle> VehicleMassiv = newList<TVehicle>(); //Представляет строго
    типизированный список объектов, доступных по индексу.

    int i;

    publicRandom rnd = newRandom();
}

```

```

public void FillTransportArray() //Заполнение массивов пассажирского и грузового
транспорта
{
    for (i = 0; i <= 19; i++)
    {
        CivilTransportArray[i] = new TCivilTransport(rnd.Next(1, 40));
        LorryTransportArray[i] = new TLorryTransport(rnd.Next(1, 20));
    }
}

public void FillStreetArray() //Заполнение массивов улиц и проспектов
{
    for (i = 0; i <= 3; i++)
        StreetArray[i] = new TStreet("Улица-" + (i + 1), (i + 1) * 100, 0);
    for (i = 0; i <= 2; i++)
        ProspektArray[i] = new TStreet("Проспект-" + (i + 1), 0, (i + 1) * 100);
}

```

Класс CityMovement

//класс для реализации движения транспортных средств

```

public class CityMovement: Init
{
    int i;
    private int VehicleCount;
    public int GetVehicleCount ()
    {
        VehicleCount = 0;
        VehicleCount = VehicleCount + CivilTransportArray.GetLength(0) +
        LorryTransportArray.GetLength(0);
        return VehicleCount;
    }
}

```



```

public int GetVehicleCount(int aType) // кол-во транспорта определенного типа в городе
{
    int res = 0;

    //TVehicle vvv = new TVehicle();

    foreach (TVehicle vvv in VehicleMassiv)
    {
        if (vvv.Type() == aType)
            res = res + 1;
    }

    return res;
}

public void test()
{
    VehicleMassiv.Add(new TCivilTransport(rnd.Next(1, 40)));
    VehicleMassiv.Add(new TCivilTransport(rnd.Next(1, 40)));
    VehicleMassiv.Add(new TLorryTransport(rnd.Next(1, 20)));
}

public CityMovement()
{
}

```

Диаграмма классов стала выглядеть следующим образом:

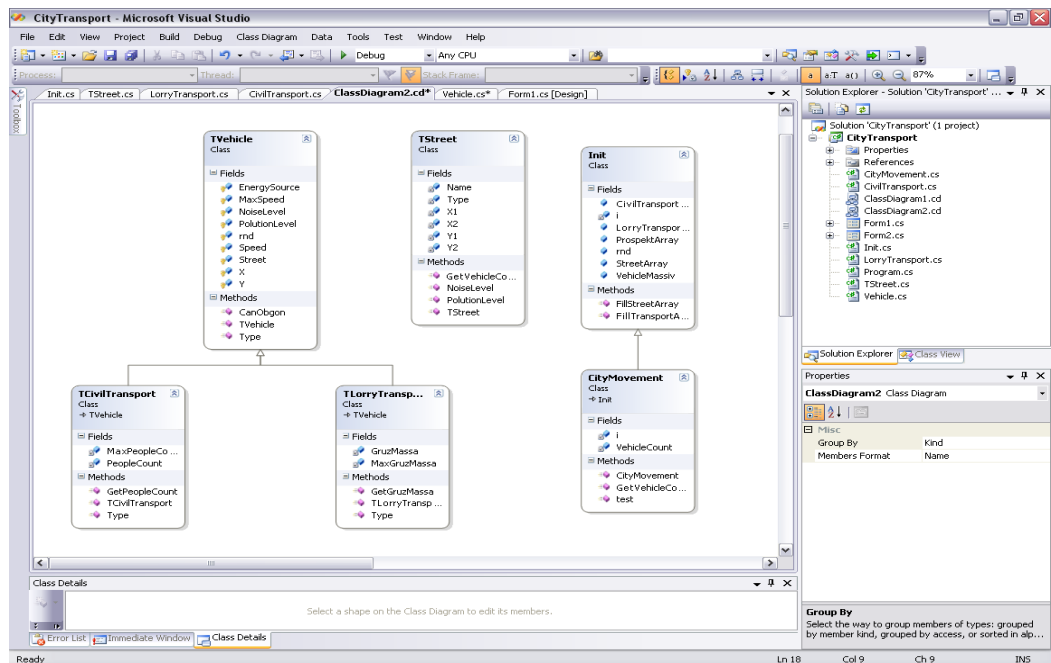


Рисунок 21 – Итоговый вариант диаграммы

Создадим интерфейс IObjectInfo для получения информации по состоянию движущегося объекта (т/с).

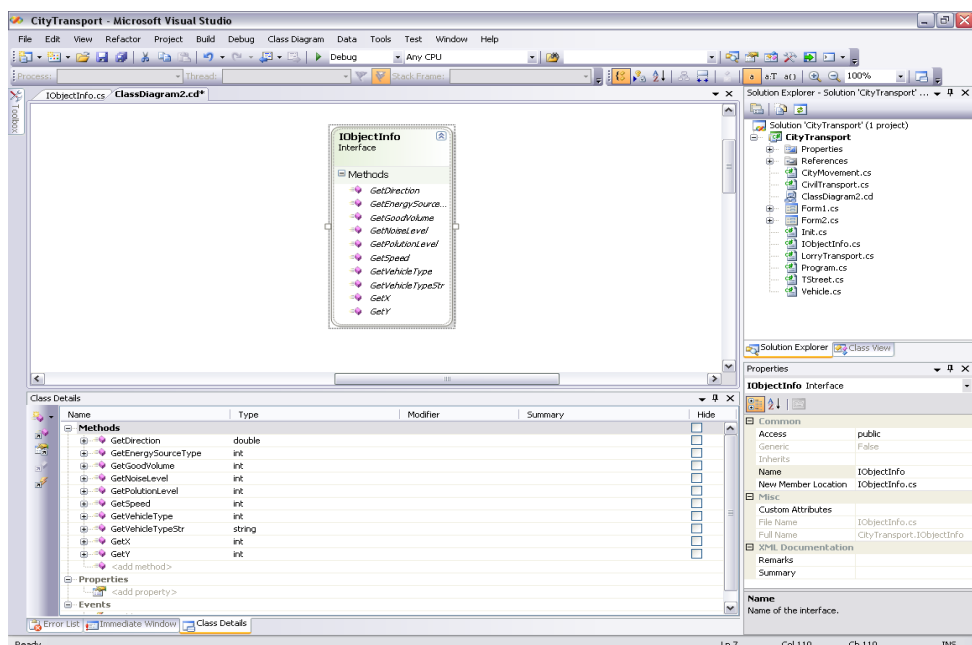


Рисунок 22 – Диаграмма и методы интерфейса IobjectInfo

Диаграмма классов после преобразований выглядит следующим образом:

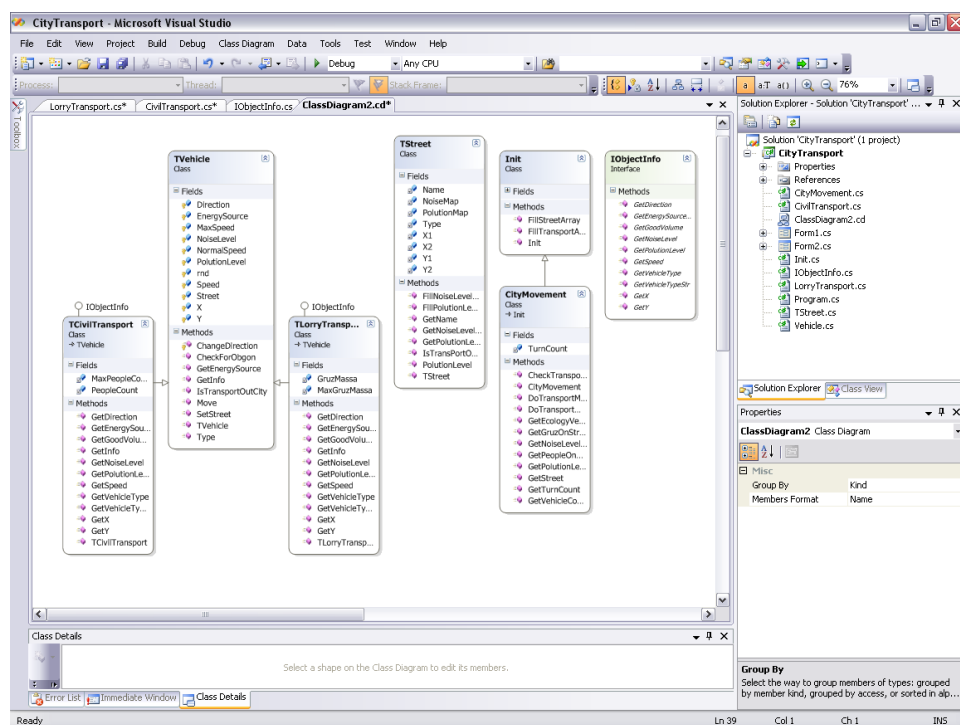


Рисунок 23 – Диаграмма классов