

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**



МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению курсовой работы

по дисциплине «Технологии программирования»

для студентов направления

09.03.02 «Информационные системы и технологии»

Ставрополь
2020

Методические указания предназначены для студентов специальности 09.03.02 «Информационные системы и технологии» по дисциплине «Технологии программирования». В них содержатся основные требования к объему, оформлению, структуре и содержанию разделов пояснительной записки, а также объясняется порядок выполнения и защиты курсовой работы. Приведен список рекомендуемой литературы, примерная тематика курсовых работ.

Составители: доцент кафедры ИСТ,
к.т.н. Е.И. Николаев.

Рецензент:

Современные технологии разработки приложений позволяют программисту сосредоточиться на решении конкретной задачи, не акцентируя внимания на вопросах взаимодействия приложения с операционной системой, периферийными устройствами, памятью, переносимости приложений. Это стало возможным в результате появления новых программных продуктов, ориентированных на технологию Microsoft .NET. Программные продукты для создания приложений для Windows предыдущего поколения (такие как MS Visual Studio 6, ориентированная на использование Win32 API или библиотек MFC и ATL) позволяли создавать приложения с использованием непосредственного вызова функций библиотек операционной системы Windows или используя надстройки над этими библиотеками.

В первом случае появлялась возможность создавать программные продукты, обладающие переносимостью между различными версиями операционных систем семейства Windows. Недостатком являлась сложность разработки приложений.

Во втором случае программист использовал библиотеки классов, позволяющие существенно упростить разработку программ, но объем кода оставался большим и данный подход все еще базировался на языке C++, что, в свою очередь, снижало устойчивость программ и усложняло отладку и модернизацию программ.

Появление технологии .NET позволило сохранить все преимущества предыдущих подходов и устранить недостатки. При этом стали доступны новые возможности в разработке приложений Windows:

- ориентированность на Microsoft .NET Framework, который является динамической средой выполнения и библиотекой классов;
- новый язык C#, использующий концепцию безопасного кода; новые типы данных, поддержка строгой типизации;
- новые коммерческие и бесплатные интегрированные среды разработки;
- высокая степень переносимости приложений за счет динамической компиляции и интеграции Framework и операционной системы;

- новые технологии в реализации основных механизмов взаимодействия программы и операционной системы;
- новые подходы к реализации взаимодействия с базами данных и сервисами операционной системы;
- новые технологии разработки сетевых приложений.

Принципы, реализованные в технологии .NET интегрированы в операционные системы Windows и существенно образом определяют не только порядок разработки пользовательских приложений, но и работу большинства современных коммерческих продуктов и составляющих сервисов операционных систем семейства Windows.

Базой для выполнения курсовой работы являются дисциплины; «Информатика», «Основы алгоритмизации и программирования», «Технология программирования». Навыки, полученные в ходе выполнения курсовой работы, будут использоваться в большинстве общепрофессиональных и специальных дисциплин.

1. ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Целью выполнения курсовой работы по дисциплине «Технологии программирования» является получение практических навыков по разработке программ для операционных систем семейства Windows с применением технологий .NET.

Задачами являются:

- изучение возможностей интегрированной среды разработки Microsoft Visual Studio;
- получение практических навыков по использованию технологий .NET: визуальных компонент;
- получение навыков программирования на языке C#;
- получение навыков программирования приложений с использованием технологий GDI+;
- получение навыков программирования приложений с использованием классов System.IO;
- получение навыков по самостоятельной формализации требований при проектировании и разработке приложений;
- возможна реализация приложения с использованием иных технологий, которые выбираются студентом и согласуются с преподавателем (Unity, стек технологий Java, Python).

2. ФОРМУЛИРОВКА ЗАДАНИЯ И ЕГО ОБЪЕМ

В ходе выполнения курсовой работы студент разрабатывает приложение (консольное или оконное приложение – в зависимости от поставленной преподавателем задачи) в соответствии с индивидуальным заданием. Задание к курсовой работе выбирается студентом самостоятельно или формулируется преподавателем. Примерные темы курсовых работ:

1. Разработка приложения «Интерактивная карта»
2. Разработка приложения «Менеджер фотографий»
3. Разработка приложения «Траектория движения»
4. Разработка приложения «Игра «Собиратель монет»
5. Разработка приложения «Свертка изображения»
6. Разработка приложения «Матричный калькулятор»
7. Разработка приложения «Аквалайзер»
8. Разработка приложения «Графический пароль»
9. Разработка приложения «Игра «Танки»
10. Разработка приложения «Игра «Змейка»
11. Разработка приложения «Расчет стоимости поездки»
12. Разработка приложения «Расчет баллистической траектории»
13. Разработка приложения «Интерактивный анализ набора данных»
14. Разработка приложения «Моделирование популяции»
15. Разработка приложения «Обработка изображения фильтрами»
16. Разработка приложения «Игра «Цветы против грибов»
17. Разработка приложения «Фракталы»
18. Разработка приложения «Искусственный интеллект»
19. Разработка приложения «Филиалы и сотрудники»
20. Разработка приложения «Игра «Ракета»
21. Разработка приложения "Игра «Блиц»
22. Разработка приложения «Поиск текста в документах»
23. Разработка приложения «Анализ матричных данных»

24.Разработка приложения «Игра «Поиск крота»

25.Разработка приложения «Вычисления со скобками»

Пояснительная записка должна состоять из следующих элементов:

- титульный лист по образцу (приложение 1);
- аннотация;
- бланк задания, подписанный руководителем, студентом и утвержденный заведующим кафедрой (приложение 2);
- содержание;
- введение;
- основная часть (построение функциональной диаграммы, диаграммы классов, обобщенный алгоритм работы программы, результаты, анализ входных данных и выходной информации);
- заключение и выводы;
- приложения.

Объем пояснительной записки не менее 25 листов и не более 40. Пример содержания пояснительной записки к курсовой работе приводится в приложении 3. Студент предоставляет преподавателю следующие материалы в электронном виде:

- проект приложения (исходный код, откомпилированные сборки, файлы с данными);
- пояснительная записка в формате MS Word;
- презентация, демонстрируемая в ходе защиты работы, в формате MS PowerPoint.

3. ОСНОВНОЕ СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ ПО РАЗДЕЛАМ, ПОСЛЕДОВАТЕЛЬНОСТЬ И ПОРЯДОК ИХ ВЫПОЛНЕНИЯ

Основная часть пояснительной записки к курсовой работе представляет собой подробное описание процесса разработки, тестирования и отладки программы, а также полное описание назначения, логической структуры и функциональных возможностей разрабатываемого программного продукта.

Основная часть должна содержать следующие подразделы:

1. Общие сведения.
2. Функциональная структура проекта.
3. Описание логической структуры программы.
4. Требования к техническому и программному обеспечению.
5. Запуск программы. Описание процесса установки.
6. Входные и выходные данные.

В зависимости от особенностей программы допускается вводить в структуру пояснительной записки дополнительные подразделы. Содержание разделов иллюстрируется примерами, графиками, схемами, таблицами.

3.1 Содержание подраздела «Общие сведения»

В подразделе «Общие сведения» необходимо указать краткое и полное наименование программы, языки программирования (среды разработки), а также другое прикладное программное обеспечение, используемые при разработке программы. Необходимо привести перечень программ, библиотек и другого программного обеспечения, необходимого для функционирования разрабатываемого приложения. Из данного подраздела непрофессиональный программист должен понять: для чего предназначена разрабатываемая программа; какие программные средства и библиотеки использовались при разработке.

3.2 Содержание подраздела «Функциональная структура проекта»

В разделе «Функциональное назначение» необходимо указать классы решаемых задач, ограничения на функциональное применение. В данном подразделе описываются возможности, предоставляемые программой. В данном разделе необходимо привести **диаграмму вариантов использования** разрабатываемой программы. Желательно указать недостатки приложения.

3.3 Содержание подраздела «Описание логической структуры программы»

В данном подразделе должны быть описаны:

1. Обобщенный алгоритм работы программы.

Необходимо представить алгоритм работы программы и описать каждую стадию работы разрабатываемого приложения: запуск программы, процессы аутентификации пользователя, загрузка данных, создание визуальных элементов управления и т.д. При описании логической структуры программы необходимо ссылаться на исходный текст программы, выносимый в приложение.

2. Объектно-ориентированная декомпозиция.

Необходимо описать каждый класс (назначение полей и методов класса) и роль объектов данного класса в приложении, а также взаимосвязи между объектами различных классов в программе (при описании взаимосвязей необходимо ссылаться на диаграмму классов, выносимую в приложение).

Необходимо рассмотреть приложение и как взаимосвязанную совокупность модулей, и как совокупность классов (таблицы 1 и 2). При описании логической структуры программы необходимо представить оба способа декомпозиции проекта разрабатываемого приложения.

Таблица 1 – Спецификация функций модуля «имя модуля» (сборка)

Прототип функции	Назначение	Назначение формальных параметров

Декомпозиция проекта с использованием спецификаций функций модулей должна отражать общую структуру полученного приложения и указывать на сборки .NET, которые образуют или используются данным приложением.

Таблица 2 – Спецификация класса «имя класса» (в заголовке таблицы указываются сборка, пространство имен, в которых содержится данный класс)

Поле класса (метод)	Способ доступа	Назначение	Назначение параметров методов класса

В таблице 2 необходимо привести только те поля классов, которые проектировались непосредственно программистом.

3. Алгоритмы, имеющие важное значение в программе и реализуемые отдельными модулями, методами, функциями. Необходимо привести блок-схемы отдельных алгоритмов, реализующих выполнение требований дипломного проектирования и наполняющих программу функциональностью: расчет агрегирующих показателей, выборка данных, формирование выходных данных.

В данном подразделе целесообразно использование таких диаграмм: **диаграмма компонентов, диаграмма последовательности, диаграмма размещения, диаграмма классов.**

3.4 Содержание подраздела «Требования к техническому и программному обеспечению»

В данном подразделе необходимо произвести расчеты минимальных требований к центральному процессору, оперативной памяти, жесткому диску, а также периферийным устройствам (монитору, приводам дисков, манипуляторам) для обеспечения функционирования программы.

В данном разделе нужно описать системное программное обеспечение, необходимое для выполнения разработанной программы.

3.5 Содержание подраздела «Запуск программы. Описание процесса установки»

В данном подразделе описывается процесс установки, возможные проблемы и способы их устранения, приводится инструкция пользователя по работе с программой. Необходимо привести копии наиболее значимых экранных форм. По тексту инструкции пользователя необходимо ссылаться на копии экранных форм, выносимых в приложения.

3.6 Содержание подраздела «Входные и выходные данные»

В разделе «Входные и выходные данные» описываются характер, предварительная организация и подготовка входных данных, а также требования к входным данным, описывается характер хранения обрабатываемых данных (в оперативной памяти, на жестком диске). При описании выходных данных необходимо описать формы представления, назначение, форматы итоговых данных программы. Необходимо ссылаться на копии отчетов, выносимые в приложение.

Материалы, которые нецелесообразно включать в разделы описания (листинг программы с комментариями, диаграммы классов, объектов, алгоритмические схемы отдельных модулей, формы отчетов и т.п.) рекомендуется выносить в приложения.

3.7 Содержание подраздела «Входные и выходные данные»

Каждая программа обрабатывает данные, которые могут вводиться пользователем и/или поставляются сервисами хранения данных (файловые хранилища, базы данных). В данном подразделе необходимо описать источники данных для приложения, форматы хранения данных, описать алгоритмы сохранения и восстановления информации из подсистем хранения.

Дополнительно, в данном подразделе, необходимо описать форматы выходных данных. Для программ учета – это отчеты, созданные с использованием внутренних средств приложения или внешних средств; для игровых приложений – это подсчет игровых очков, экономика игры, отображение информации о ходе выполнения игровой задачи и т.п. В случае, если программа не только отображает некоторые выходные данные, но и позволяет сохранять данные – необходимо описать форматы хранения со ссылками на алгоритмы сохранения данных.

4. ОБЩИЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

4.1 Основные требования к форматированию

Пояснительная записка к курсовой работе относится к текстовым документам и должна соответствовать требованиям ГОСТ.

Поля документа: верхнее – 2,5 см, нижнее – 2 см, левое – 2,5 см, правое – 1,5 см. выравнивание заголовков, подзаголовков, основного текста – по ширине. Отступ первой строки абзаца – 1,25 см. Отступы перед и после абзаца – 0 см.

По всему тексту межстрочный интервал – полуторный. Исключение: в таблицах разрешается применять одинарный интервал.

После заголовков разделов и заголовков подразделов – 1 пустая строка.

По всей пояснительной записке: размер шрифта – 14. Исключение: в объемных таблицах разрешается уменьшить размер шрифта до 12.

Текст следует разделить на разделы и подразделы. Разделы должны иметь сквозные номера в пределах всего документа, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Подразделы имеют нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и подраздела, разделенных точкой. В конце номера пункта точка не ставится. Каждый раздел начинается с новой страницы.

Содержание пояснительной записки следует помещать в начале, список использованной литературы – в конце (перед приложениями). В содержание включают наименование всех разделов, подразделов, пунктов без перефразировки с указанием номеров страниц, на которых размещается начало раздела. Введение и заключение не нумеруются.

Разделы и подразделы должны иметь заголовки. Их печатают с прописной буквы без тире и точки в конце, не подчеркивая. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой.

Слово «Содержание» записывается в виде заголовка с прописной буквы.

Каждая таблица должна иметь номер и название. Нумерация – сквозная по документу (1, 2, 3 и т.д.) или в пределах раздела (1.1, 1.2, 2.1, 2.2 и т.д.; первая цифра – номер раздела, вторая – номер таблицы в данном разделе). На каждую таблицу необходимо делать ссылку в основном тексте пояснительной записки.

Пример. В таблице 2.1 представлена информация о структуре класса Form1. Таблица получилась объемной, поэтому межстрочный интервал уменьшен до одинарного. Выравнивание текста в заголовке – по центру, в строках таблицы – по ширине. Расположение таблицы на странице выполнено таким образом, чтобы предотвратить перенос таблицы на новую страницу. Не допускается разрывать название таблицы и саму таблицу.

Таблица 2.1 – Методы и поля данных класса Form1

Наименование члена класса	Тип	Формальные параметры	Описание
Form1_Load	void	object sender, EventArgs args	Метод для обработки события загрузки формы: выполняется загрузка данных в таблицу m_datagridview
btnSave	Button	–	Кнопка для сохранения изменений в базу данных

Рисунки располагаются в основном тексте по центру. Подписуемая подпись – по центру. На рисунок должна быть сноска в основном тексте. Текст на рисунках должен быть читаемым.

Пример. На рисунке 2.1 представлена диаграмма классов, отражающая связи между классами, предназначенными для отображения сущностей реляционной базы данных (БД) «Курсы». Данные классы предназначены для представления данных из БД.

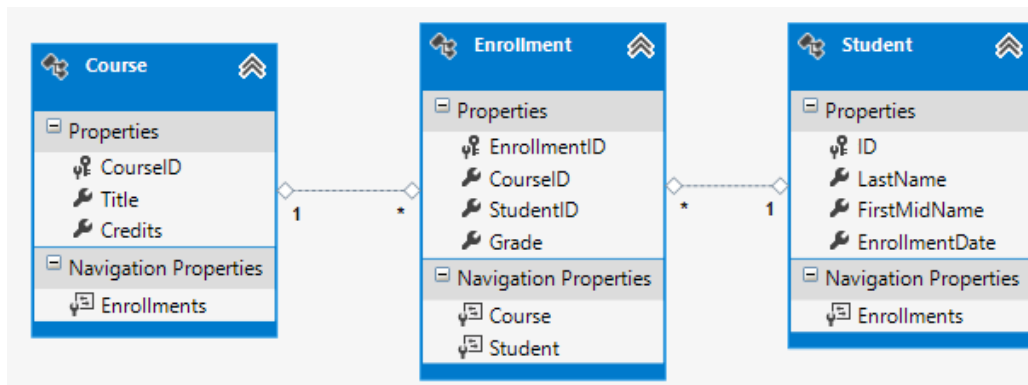


Рисунок 2.1 – Классы для сущностей из базы данных «Курсы»

Обратите внимание на технику вставки рисунка:

1. Создается таблица с одним столбцом и двумя строками. Выравнивание таблицы – «Автоподбор по ширине окна». Выравнивание содержимого таблицы – по центру. В первую строку таблицы помещается рисунок, во вторую – подрисуночная подпись.

2. Очевидно, что границы таблицы необходимо сделать невидимыми.

В дальнейшем для вставки нового рисунка необходимо просто скопировать данную структуру и заполнить строки таблицы новым содержимым.

Литература оформляется по образцу:

1. Алексеев, В. Е., Графы и алгоритмы. Структуры данных. Модели вычислений. Учебник / В. Е. Алексеев, В. А. Таланов. – М. : Бином, 2013. – 320 с.

Приложения располагаются в конце пояснительной записки в порядке их появления в тексте. Каждое приложение начинается с нового листа, в правом верхнем углу которого ставится обозначение «Приложение 1», «Приложение 2» и т.д.

Рисунки, таблицы нумеруются в пределах каждого приложения. В пояснительной записке все страницы, в том числе титульный лист, содержание нумеруются арабскими цифрами. На титульном листе и содержании номер не ставят, на последующих листах проставляют в правом верхнем углу.

При изложении обязательных требований в тексте пояснительной записки должны применяться слова: «должен», «необходимо», «требуется, чтобы», «разрешается только», «не допускается», «запрещается». При изложении других положений следует применять слова: «могут быть», «как правило», «при необходимости» и т.д.

4.2 Важные замечания по форматированию

В данном подразделе описаны различные аспекты форматирования, на которые следует обратить внимание

1. Списки необходимо использовать или нумерованные, как в данном примере, или маркированные, но только с маркером типа «—». Необходимо обратить внимание на форматирование списков (оно должно осуществляться по тем же правилам, что и основной текст).

2. Каждый новый раздел начинается с новой страницы.

3. Точки в конце названий таблиц и рисунков не ставятся.

4. Текст должен быть оформлен единообразно. Никаких «украшений» использовать не нужно! Никаких непонятных разрывов текста! Никаких непонятных отступов после (перед) абзацами использовать не нужно!

5. Нумерация страниц – сквозная по всему документу. Нумерация ведется с титульного листа, но первый номер страницы появляется только на листе раздела «ВВЕДЕНИЕ». То есть на титульном листе, бланке задания, листе оценки, содержания – страницы не ставятся.

5. РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ РАБОТ НАД КУРСОВОЙ РАБОТОЙ, ПРИМЕРНЫЙ КАЛЕНДАРНЫЙ ПЛАН ВЫПОЛНЕНИЯ

Работу над КР целесообразно реализовать поэтапно и придерживаться следующего графика:

Порядок выполнения	Неделя
Выдача задания	1
Реализация подраздела «Общие сведения»	2
Работа над подразделом «Функциональное назначение»	3-4
Разработка приложения	4-11
Работа над пояснительной запиской	12-13
Сдача на проверку, доработка программы и пояснительной записки	14
Защита	15

Работу над курсовой работой по дисциплине «Технологии программирования» рекомендуется выполнять в следующей последовательности:

1 этап. Получение задания. Тема и задание курсовой работы выбираются студентом из списка, утвержденного на заседании кафедры, согласуется с руководителем и утверждается заведующим кафедрой.

2 этап. Изучение теоретического материала. Руководитель рекомендует список литературы, с которой студенту необходимо ознакомиться перед выполнением основной части работы.

3 этап. Разработка приложения. В соответствии с заданием к курсовой работе студент разрабатывает приложение на языке C# в среде MS Visual Studio.

4 этап. Составление пояснительной записки к курсовой работе. Общий объем пояснительной записки не должен превышать 40 страниц, в том числе: введение – не более 1 страницы, основная часть – 25-30 страниц, заключения и выводы – 2-5 страницы. Записка иллюстрируется схемами, копиями экранных форм, выходными документами, текстами программ (выносятся в приложение).

Список литературы должен содержать не менее 30 источников, на которые обязательно присутствуют ссылки в основном тексте пояснительной записки.

5 этап. Контроль результатов руководителем. Пояснительная записка сдается на проверку руководителю работы в срок не менее чем за 10 дней до дня защиты. По итогам проверки руководитель либо допускает студента к защите, либо возвращает пояснительную записку на доработку. Программная реализация обязательно прилагается на электронном носителе и демонстрируется руководителю.

6. ПОРЯДОК ЗАЩИТЫ И ОТВЕТСТВЕННОСТЬ СТУДЕНТА ЗА ВЫПОЛНЕНИЕ ЗАДАНИЯ ПО КУРСОЙ РАБОТЕ

Защита работы осуществляется перед комиссией. Состав комиссии утверждается на заседании кафедры. Студент делает доклад (5-7 мин), в котором кратко излагает основные результаты курсового проектирования, отвечает на вопросы членов комиссии. В ходе защиты используется презентация. Затем демонстрируется разработанное приложение.

В случае полного несоответствия разработанного приложения заданию на выполнение курсовой работы, студент к защите работы не допускается руководителем. В случае небольших отклонений функций разработанного приложения или его интерфейса от задания на курсовое проектирование, студент к защите допускается. Комиссия вправе снизить оценку на защите курсовой работы, если студент не сможет доказать целесообразность внесенных изменений в функциональность приложения или его интерфейс.

7. СПИСОК ЛИТЕРАТУРЫ

1. Нортроп, Т. Основы разработки приложений на платформе Microsoft .NET Framework. Учебный курс Microsoft. Перевод с англ./ Т. Нортроп, Ш. Уилдермьюс, Б. Райан. – М. : «Русская редакция», 2010. – 864 с.
2. Шилдт, Г. С#, учебный курс. / Г. Шилдт. – СПб. : Питер, 2010. – 512 с.
3. Робинсон, С. С# для профессионалов. Том 1. / С. Робинсон, О. Корнес, Д. Глин, Б. Харвей. – М. :Лори, 2010. – 1002 с.
4. Робинсон, С. С# для профессионалов. Том 2. / С. Робинсон, О. Корнес, Д. Глин, Б. Харвей. – М. :Лори, 2010. – 998 с.
5. Троелсен, Э. Язык программирования С# 2012 и платформа .NET 4.5: Пер с англ. / Э. Троелсен. – М. : ООО «И.Д. Вильямс», 2013. – 1168 с.
6. Методические указания к выполнению лабораторных работ по дисциплине «Технологии программирования» для студентов специальности 09.03.02 "Информационные системы и технологии" / сост. Николаев Е.И.; рец. Маликов А.В. – Ставрополь : СКФУ, 2016. – 50 с.
7. Учебное пособие «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И.; рец. Маликов А.В. – Ставрополь : СКФУ, 2014. – 150 с.

ПРИЛОЖЕНИЕ 1. ТИТУЛЬНЫЙ ЛИСТ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

Министерство науки и высшего образования российской федерации
ФГАОУ ВО «Северо-Кавказский федеральный университет»
Институт математики и информационных технологий им. профессора Н.И. Червякова
Кафедра информационных систем и технологий

КУРСОВАЯ РАБОТА

по дисциплине

«Технологии программирования»

на тему:

«Разработка приложения для автоматизации предприятия в сфере розничной торговли»

Выполнил:

Иванов Иван Иванович

студент 2 курса

группы **ИНС-б-о-191**

направления 09.03.02

информационные системы и
технологии

очной формы обучения

(подпись)

Руководитель работы:

Николаев Е.И., доцент ИСТ

(ФИО, должность, кафедра)

Работа допущена к защите _____
(подпись руководителя)

(дата)

Работа выполнена и
защищена с оценкой _____

Дата защиты _____

Члены комиссии:

(должность)

(подпись)

(И.О. Фамилия)

Ставрополь, 20__ г.

УТВЕРЖДАЮ

В.И. Дроздова

(подпись)

« » 2021 г.

ЗАДАНИЕ на курсовую работу

студента *Иванов Иван Иванович*

(фамилия, имя, отчество)

по дисциплине Технологии программирования

1. Тема работы *Разработка приложения «Траектория движения»*

2. Цель Разработка приложения на основе технологий .NET Framework для визуализации траекторий точки на плоскости

3. Задачи:

1. Разработать приложение для построения графика $y(x)$, если заданы функции $x = x(t)$, $y = y(t)$.
2. Реализовать механизм сохранения/загрузки параметров функций $x(t)$ и $y(t)$.
3. Реализовать анимацию, демонстрирующую построение графика в замедленном режиме.
4. Разработать механизм редактирования параметров функций $x(t)$ и $y(t)$ и обновление графика.

4. Перечень подлежащих разработке вопросов:

а) по теоретической части: архитектура приложений, построенных на основе технологии .NET Framework; основные библиотечные классы, используемые в процессе разработки приложения; основные процессы предметной области, подлежащие автоматизации.

б) по практической части: дизайн приложения; сценарий режима слайдов; разработка многомодульного приложения; использование GDI+; использование классов System.IO; реализация механизма пользовательских настроек.

5. Исходные данные:

Внешний вид приложения студент выбирает самостоятельно. Количество точек при построении графика должно меняться в настройках; количество параметров в каждой функции ($x(t)$, $y(t)$): более 5; количество пользовательских настроек: более 5.

6. Критерии оценивания

Наименование критерия	Балл
Реализованы все задачи индивидуального задания	20
Выполнены все требования к исходным данным приложения	20
Выполнены требования к содержанию пояснительной записки	20
Выполнены требования к форматированию пояснительной записки	10
Презентация отражает все стадии работы над проектом	10
В ходе защиты студент отвечает на все вопросы, касающиеся программы и содержимого пояснительной записки	20
Максимальное количество баллов:	100

7. Шкала оценивания

Баллы	Оценка	Баллы	Оценка
[90; 100]	Отлично	[60; 80)	Удовлетворительно
[80; 90)	Хорошо	менее 60	Неудовлетворительно

8. Список рекомендуемой литературы

1. Язык программирования C# 7 и платформы .NET и .NET Core (8-е издание) / Э. Троелсен. – М.: Диалектика, 2018. – 1330 с.
2. Учебное пособие по дисциплине «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 148 с.
3. Учебное пособие (лабораторный практикум) по дисциплине «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 150 с.
4. Методические указания к выполнению курсовой работы по дисциплине «Технологии программирования» для студентов специальности 09.03.02 «Информационные системы и технологии» / сост. Николаев Е.И. – Ставрополь: СКФУ, 2020. – 25 с.

9. Контрольные сроки:

- предоставление работы на проверку «7» мая 2021 г.
- доработка программы, пояснительной записки «14» мая 2021 г.
- защита курсовой работы с презентацией «21» мая 2021 г.

Руководитель курсовой работы
канд. техн. наук, доцент

(ученая степень, звание)

(личная подпись)

Е.И. Николаев

(инициалы, фамилия)

Задание принял(а) к исполнению студент(ка)

(дата)

(личная подпись)

(инициалы, фамилия)

ПРИЛОЖЕНИЕ 3. ПРИМЕР СОДЕРЖАНИЯ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

Введение.....	4
1 Разработка приложения «Игра «Морской бой»	6
1.1 Общие сведения о проекте.....	6
1.2 Функциональная структура проекта	7
1.3 Описание логической структуры приложения	12
2 Информационное и программное обеспечение	22
2.4 Требования к техническому и программному обеспечению для приложения «Игра «Морской бой».....	22
2.5 Руководство пользователя. Запуск программы. Описание процесса установки.....	24
2.6 Формат входных данных. Выходные данные приложения	26
Заключения и выводы.....	28
Список литературы	30
Приложение 1. Диаграмма классов приложения.....	32
Приложение 2. Диаграмма вариантов использования	33
Приложение 3. Диаграмма размещения «Структура приложения»	34
Приложение 4. Исходный код метода DoAll()	35
Приложение 5. Блок-схема алгоритма поиска цели.....	37
Приложение 6. Исходный код модуля OpenData.cs	38

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению курсовой работы по дисциплине
«Технологии программирования»
для студентов направления
09.03.02 «Информационные системы и технологии»

Составитель: Николаев Е. И., к.т.н.

Редактор:

Подписано в печать __.__.10

Формат 60x84 1/16.

Усл. п. л. – ____

Уч.-изд. л. – ____.

Бумага газетная.

Печать офсетная.

Заказ

Тираж 50 экз.

ФГАОУ ВО «Северо-Кавказский федеральный университет»

355011, г. Ставрополь, ул. Пушкина, 1

Издательство Северо-Кавказского федерального университета
Отпечатано в типографии СКФУ