

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Моисеев Роман Евгеньевич

Должность: Проректор по образовательной деятельности

Дата подписания: 26.01.2024 14:19:43

Уникальный программный ключ:

8332314f4b9fba696810b838ac7765c5742d0fe

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Казанский национальный исследовательский  
технический университет им. А.Н. Туполева-КАИ»**

**Чистопольский филиал «Восток»**

**Кафедра компьютерных и телекоммуникационных систем**

---

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

**по выполнению**

**КУРСОВОЙ РАБОТЫ**

**по дисциплине (модулю)**

**Объектно-ориентированное программирование**

Методические указания (рекомендации) по выполнению курсовой работы/курсового проекта предназначены для обучающихся всех форм обучения по направлениям подготовки:

Код и наименование направления подготовки / специальности	Направленность (профиль, специализация, магистерская программа)	ФГОС ВО утвержден приказом Минобрнауки России
09.03.03 Прикладная информатика	Прикладная информатика в цифровой экономике	№ 922 от 19.09.2017

В методических указаниях приведены требования к выполнению курсовой работы, даны рекомендации по структуре, содержанию, оформлению, порядку выполнения и защите курсовой работы по дисциплине (модулю) Программирование и основы алгоритмизации.

Разработчик(и):

Ефимова Юлия Викторовна, кандидат педагогических наук

Методические указания рассмотрены на заседании кафедры Компьютерных и телекоммуникационных систем, протокол № 6 от 27.02.2023г.

Заведующий кафедрой Компьютерных и телекоммуникационных систем

Классен Виктор Иванович, д.т.н.

## Содержание

1 ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ.....	4
2 ТРЕБОВАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ.....	5
3 СТРУКТУРА КУРСОВОЙ РАБОТЫ .....	7
4 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ.....	8
5 ТИПОВЫЕ ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ.....	11
6 ЛИТЕРАТУРА.....	13

## **1 ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ**

**Целью курсовой работы** является приобретение практических навыков поэтапной разработки нераспределённых объектно-ориентированных программ с использованием .NET технологии, разработки собственных классов и применения классов библиотеки .NET Framework, создания из них объектов и обеспечении требуемого совместного функционирования этих объектов как объектно-ориентированной модели, реализующей поставленную задачу, а также описания разработанного программного продукта как с точки зрения его пользователя, так и с точки зрения разработчика.

В результате выполнения курсовой работы студенты должны:

1. Изучить и применить принципы поэтапной разработки и отладки программ средней сложности.
2. Уметь разрабатывать собственные классы, создавать из них разнообразные объекты и применять их совместно для реализации требуемых алгоритмов функционирования программы.
3. Уметь применять необходимые классы библиотеки NET Framework, создавать из них требуемые объекты и употреблять их в программе.
4. Уметь разрабатывать собственный интерфейс программы.

## **2 ТРЕБОВАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ**

При выполнении курсовой работы обучающийся прежде всего должен проанализировать предметную область, связанную с разрабатываемой программной системой, выделить основные объекты и разработать для них соответствующие классы или воспользоваться готовыми классами библиотеки NET Framework.

Тщательно необходимо продумать вопросы, связанные с применением потоков и использования ими событий.

После анализа совокупности предполагаемых объектов и их взаимодействия и разработки соответствующих им классов надлежит перейти к поэтапной разработке создаваемой программной системы. Перед переходом к созданию программы на компьютере желательно на бумаге прикинуть поэтапную разработку программы, при которой программа из основного ядра будет обрастать объектами добавляемых классов, приближая своё функционирование к требуемому по заданию.

Желательно на каждом этапе добавлять по одному классу и каждый этап тщательно отлаживать. Ввиду отлаженных предыдущих этапов ошибки на данном этапе будут вызваны некорректностью добавляемых классов или неправильным применением их объектов.

Очень полезно для просмотра деталей функционирования программы любого этапа и облегчения поиска ошибок в Windows приложении кроме прикладного окна приложения иметь и консольное окно, выдавая на него изменяющиеся состояния объектов. По мере отладки целесообразно закомментировать в программе некоторые операторы выдачи на консоль, дабы сократить объём выводимой информации.

Результаты разработки каждого этапа и возникшие при этом проблемы должны быть отражены в пояснительной записке по курсовой работе.

Тематика курсовых работ должна быть связана с разработкой программ, реализующих модель, состоящую из разнотипных объектов, взаимодействующих между собой. Программный интерфейс должен включать

интерфейсные элементы типа меню или кнопок, объекты которых создаются из соответствующих классов библиотеки NET Framework.

Задание на курсовую работу описывает поведение некоторого множества объектов, подлежащих реализации в виде программной системы.

Это задание должно быть уточнено разработчиком путём уточнения поведения объектов, предложением новых событий, потоков и новых взаимосвязей между объектами. Разработчик должен предложить программный интерфейс, позволяющий не только управлять функционированием системы объектов, но и отображать информацию об их функционировании дополнительных окнах или текстовых редакторах.

Разработанная программа должна быть реализована на указанном языке .NET платформы на последней версии Visual Studio .NET.

### **3 СТРУКТУРА КУРСОВОЙ РАБОТЫ**

Текстовая часть курсовой работы оформляется в виде пояснительной записки (ПЗ) в соответствии с требованиями стандарта [13,14].

Функционирование программы иллюстрируется приведёнными прикладными окнами, отображающими особые взаимодействия объектов, представленными в окне в виде некоторых фигур или изображениями.

Тексты программ помещаются в приложении, в конце пояснительной записки.

Пояснительная записка должна содержать:

Титульный лист;

Лист оглавления;

1 Задание;

2. Основной раздел с описанием разрабатываемой программной части.

3. Список используемой литературы.

4. Приложение. Текст программы.

Описание функции должно включать ее назначение, интерфейс (заголовок функции с пояснением ее параметров и возвращаемого значения), описание локальных переменных и описание алгоритма функции.

Приложения к отчету должны содержать листинг программы, примеры (один или несколько) входного файла и результаты тестирования программы.

## **4 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ**

В рамках рассмотренной выше структуры курсовой работы рекомендуется использовать следующие правила оформления.

Объем курсовой работы: 20-30 страниц стандартного текста формата А4 (210 x 297), набранных через полтора интервала на одной стороне листа белой бумаги в текстовом процессоре Word. Шрифт текста должен быть четким. Размер шрифта – 14 пунктов.

Поля: левое – 25-30 мм, правое - 10 мм, верхнее -20 мм, нижнее - 25 мм. Абзацный отступ должен быть одинаковым и равным 1,25-1,27 см (равен одному нажатию клавиши Tab).

Титульный лист оформляется по образцу, приведенному в приложении.

Каждый раздел (глава) начинаются с нового листа. Каждый параграф (подзаголовок) отделяются от текста двумя интервалами.

Все страницы курсовой работы, включая иллюстрации и приложения, нумеруются по порядку от титульного листа до последней страницы без пропусков и повторений. Первой страницей является титульный лист, оформленный в соответствующем порядке (см. приложение 1), номер страницы на нем не ставится. На последующих страницах порядковый номер печатается в середине нижнего края страницы или в правом нижнем углу.

За титульным листом следует страница с указанием содержания (оглавления) работы в соответствии с ее планом и рубрикацией в тексте.

Иллюстрации (кроме таблиц) обозначаются словом «Рисунок» и нумеруются последовательно арабскими цифрами в пределах раздела (главы).

Номер рисунка и его наименование размещают ниже самого рисунка, подрисуночная подпись выравнивается по центру строки.

Если в работе приведена одна иллюстрация, то ее не нумеруют и слово «Рисунок» не пишут.

Таблицы нумеруют последовательно арабскими цифрами в пределах раздела (главы). Каждой таблице предшествует заголовок таблицы, который

помещается перед таблицей с выравниванием по правому краю текста. Заголовок начинается со слова «Таблица» с указанием номера этой таблицы, состоящего из номера раздела и порядкового номера таблицы, далее следует текстовая часть заголовка, например:

### Таблица 2.3

Точка в конце заголовка таблицы и подрисуночной подписи не ставится. Таблицу размещают после первого упоминания о ней в тексте таким образом, чтобы читать ее можно было без поворота работы или с поворотом по часовой стрелке. Ссылка на таблицу по ходу текста выполняется так: в табл. 2.3 приводятся данные о..., при повторной ссылке – см. табл. 2.3.

Примечания к таблицам, иллюстрациям или пунктам и подпунктам текста размещают непосредственно после пункта, подпункта, таблицы, иллюстрации, к которым они относятся, и печатают с прописной буквы с абзацного отступа. Слово «Примечание» следует печатать с абзацного отступа жирным шрифтом.

Ссылки на разделы, подразделы, пункты, подпункты, иллюстрации, таблицы, формулы, уравнения, перечисления, приложения, на литературные источники следуют указывать порядковым номером, например: «... в разделе 4», «... по пункту 3.3.4», «... в подпункте 2.3.41, перечисление 3», «... по формуле (3)», «... в уравнении (2)», «... на рисунке 8», «... в приложении б», «... в работе [2]».

Если в работе одна иллюстрация, таблица, формула, уравнение, или приложение следует при ссылках писать «на рисунке», «в таблице», «по формуле», «в уравнении», «в приложении».

Пояснение значений символов и числовых коэффициентов следует приводить непосредственно под формулой в той же последовательности, в которой даны в формуле. Значение каждого символа и числового коэффициента следует давать с новой строки. Первую строку пояснения начинают со слова «где» без двоеточия.

Формулы в работе следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке. Если в работе только одна формула или уравнение, их не нумеруют.

Разделы пояснительной записки могут начинаться с нового листа, либо следовать за предыдущим разделом, продолжая его лист. Разделы, подразделы, и пункты нумеруются арабскими цифрами с точкой. Наименование раздела записывается в виде заголовка прописными буквами. Наименование подразделов записываются с абзаца строчными буквами (кроме первой прописной), переносы запрещены.

Приложение помещают в конце пояснительной записки и размещают в нем вспомогательный материал. Приложение может иметь заголовок, который записывается симметрично тексту прописными буквами.

## 5 ТИПОВЫЕ ЗАДАНИЯ НА КУРСОВУЮ РАБОТУ

Разработать игровое графическое приложение. В должны быть созданы два типа движущихся объектов: управляемые и неуправляемые. Для изменения характеристик объектов используется меню. Приложение должно иметь хотя бы один класс.

Студент может предложить свой вариант задания по согласованию с преподавателем, тогда в задании указываются обязательные параметры программного приложения.

1. Программа состоит из двух форм: меню и игра. Цель игры в том, чтобы дойти до конца поля управляемым объектом, не сталкиваясь с препятствиями. С помощью меню, находящегося вверху формы, регулировать размеры и скорость фигур. Приложение должно иметь хотя бы один класс.

2. Приложение состоит из трех форм: меню игры, игровое поле и база данных. Цель игры состоит в том, чтобы игрок, управляя объектом, как можно больше раз смог достичь финиша, не задевая двигающиеся препятствия. При каждом достижении игроком финиша счет игры увеличивается на единицу. Для изменения свойств объектов используется меню.

3. Создать приложение с одним управляемым объектом, тремя движущимися и одним неподвижным объектами - препятствиями. Приложение должно содержать в себе как минимум один класс.

4. Создать приложение с одним управляемым объектом, двумя движущимися и тремя неподвижным объектами - препятствиями. Движущиеся объекты получают при загрузке формы случайные координаты. Приложение должно содержать в себе как минимум один класс.

5. Создать приложение с одним управляемым объектом, двумя движущимися и тремя неподвижным объектами - препятствиями. Движущийся управляемый объект получает при загрузке формы случайные координаты. Приложение должно содержать в себе как минимум один класс.

6. Создать приложение с одним управляемым объектом, двумя движущимися объектами - препятствиями. Движущиеся объекты получают при загрузке формы скорости движения случайно. Приложение должно содержать в себе как минимум один класс.

7. Создать приложение с двумя движущимися объектами – препятствиями и одним управляемым объектом, располагающимся в начале координат при загрузке формы. Движущиеся объекты получают при загрузке формы случайный цвет, из трех возможных. Приложение должно содержать в себе как минимум один класс.

8. Создать приложение с двумя движущимися объектами – препятствиями и одним управляемым объектом, располагающимся в заданной точке при загрузке формы. Приложение должно содержать в себе как минимум один класс. Движущиеся объекты при загрузке формы начинают двигаться параллельно оси ОУ.

9. Создать приложение с двумя движущимися объектами – препятствиями и одним управляемым объектом, располагающимся в заданной точке при загрузке формы. Движущиеся объекты при загрузке формы движутся параллельно оси ОХ. Приложение должно содержать в себе как минимум один класс.

10. Создать приложение с двумя движущимися объектами – препятствиями и одним управляемым объектом, располагающимся в заданной точке при загрузке формы. Приложение должно содержать в себе как минимум один класс. Движущиеся объекты при загрузке формы начинают двигаться параллельно оси ОУ, при выходе за границы формы появляются в точке со случайными координатами.

11. Создать приложение с двумя движущимися объектами – препятствиями и одним управляемым объектом, располагающимся в заданной точке при загрузке формы. Движущиеся объекты при загрузке формы движутся параллельно оси ОХ, при выходе за границы формы

появляются в точке со случайными координатами. Приложение должно содержать в себе как минимум один класс.

## **6 ЛИТЕРАТУРА**

1. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник для вузов. - СПб.: Питер, 2009. - 432 с.

2. Гуриков С.Р. Введение в программирование на языке Visual C#: учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 447 с. — Режим доступа: <http://znanium.com/bookread2.php?book=752394>

3. Объектно-ориентированное программирование с примерами на C#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с. — Режим доступа: <http://znanium.com/bookread2.php?book=529350>

4. Коу Дж., Джеани М. Объектно-ориентированное программирование. Учебный курс. - СПб: Питер, 2005. - 238 с.

5. Гагарина Л.Г. Введение в архитектуру программного обеспечения : учеб. пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. — М. : ФОРУМ : ИНФРА-М, 2017. — 320 с. — Режим доступа: <http://znanium.com/bookread2.php?book=615207>

6. Разработка приложений на C# с использованием СУБД PostgreSQL / Васюткина И.А., Трошина Г.В., Бычков М.И. - Новосиб.:НГТУ, 2015. - 143 с. — Режим доступа: <http://znanium.com/bookread2.php?book=556925>