МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное

образовательное учреждение высшего образования

**«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий

Кафедра математического и программного обеспечения ЭВМ

КУРСОВАЯ РАБОТА

по дисциплине Объектно-ориентированное программирование

на тему Объектно-ориентированное программирование на языке С++

Выполнил студент группы

  1ПИб-01-21оп

*группа*

направления подготовки

  09.03.04 Программная инженерия

*шифр, наименование*

  Скрыник Кирилл Андреевич

*фамилия, имя, отчество*

Руководитель

Ершов Евгений Валентинович

*фамилия, имя, отчество*

д. т. н., профессор

*должность*

Дата представления работы

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_ г.

Заключение о допуске к защите

Оценка \_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*количество баллов*

Подпись преподавателя \_\_\_\_\_\_\_\_\_ \_\_

Череповец, 2020

Аннотация

Данная расчётно-пояснительная записка содержит ….

Оглавление

[Введение 4](#_Toc40796729)

[1. Объектно-ориентированный анализ предметной области 5](#_Toc40796730)

[Приложение 1. Техническое задание 8](#_Toc40796731)

# Введение

**Объектно-ориентированное программирование** (сокращенно ООП) — это парадигма разработки программных систем, в которой приложения состоят из объектов.

Ключевые черты ООП:

1. **Инкапсуляция** - это определение классов - пользовательских типов данных, объединяющих своё содержимое в единый тип и реализующих некоторые операции или методы над ним. Классы обычно являются основой модульности, инкапсуляции и абстракции данных в языках ООП.
2. **Наследование** - способ определения нового типа, когда новый тип наследует элементы (свойства и методы) существующего, модифицируя или расширяя их. Это способствует выражению специализации и генерализации.
3. **Полиморфизм**, позволяет единообразно ссылаться на объекты различных классов (обычно внутри некоторой иерархии). Это делает классы ещё удобнее и облегчает расширение и поддержку программ, основанных на них.

ООП дает возможность создавать расширяемые системы. Это одно из самых значительных достоинств ООП и именно оно отличает данный подход от традиционных методов программирования. Расширяемость означает, что существующую систему можно заставить работать с новыми компонентами, причем без внесения в нее каких-либо изменений. Компоненты могут быть добавлены на этапе выполнения.

Использование принципов ООП помогает в разработке программ: объекты зачастую спокойно могут транслироваться в объекты реального мира, что упрощает моделирование всей системы в целом, а также упрощает восприятие всей задачи программистом.

# 1. Объектно-ориентированный анализ предметной области

В центре ООП находится понятие объекта. Объект — это сущность, которой можно посылать сообщения и которая может на них реагировать, используя свои данные. Объект — это экземпляр класса. Данные объекта скрыты от остальной программы. Инкапсуляция включает в себя сокрытие.

Наличие инкапсуляции достаточно для объектности языка программирования, но ещё не означает его объектной ориентированности — для этого требуется наличие наследования.

Но даже наличие инкапсуляции и наследования не делает язык программирования в полной мере объектным с точки зрения ООП. Основные преимущества ООП проявляются только в том случае, когда в языке программирования реализован полиморфизм подтипов — возможность единообразно обрабатывать объекты с различной реализацией при условии наличия общего интерфейса.

Абстракция данных – абстрагирование означает выделение значимой информации и исключение из рассмотрения незначимой. В ООП рассматривают лишь абстракцию данных (нередко называя её просто «абстракцией»), подразумевая набор наиболее значимых характеристик объекта, доступных остальной программе.

Инкапсуляция — свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе.

Наследование — концепция объектно-ориентированного программирования, согласно которой абстрактный тип данных может наследовать данные и функциональность некоторого существующего типа, способствуя повторному использованию компонентов программного обеспечения.

Полиморфизм подтипов — свойство системы, позволяющее использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Другой вид полиморфизма — параметрический — в ООП называют обобщённым программированием.

Класс — в объектно-ориентированном программировании, представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов. Другие абстрактные типы данных — метаклассы, интерфейсы, структуры, перечисления, — характеризуются какими-то своими, другими особенностями. Класс является ключевым понятием в ООП. Существуют и бесклассовые объектно-ориентированные языки, например, Self, Lua. Суть отличия классов от других абстрактных типов данных состоит в том, что при задании типа данных класс определяет одновременно как интерфейс, так и реализацию для всех своих экземпляров (т.е. объектов), вызов метода-конструктора обязателен.

На практике объектно-ориентированное программирование сводится к созданию некоторого количества классов, включая интерфейс и реализацию, и последующему их использованию. Графическое представление некоторого количества классов и связей между ними называется диаграммой классов. Объектно-ориентированный подход за время своего развития накопил множество рекомендаций (паттернов) по созданию классов и иерархий классов. В процессе объектно-ориентированного анализа основное внимание уделяется определению и описанию объектов (или понятий) в терминах предметной области.

Объектно-ориентированный анализ — это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области.

Объектно-ориентированное проектирование — это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

Основные принципы объектно-ориентированного анализа и проектирования.

1. Сначала производится анализ требований, во время кот выделяются основные процессы, происходящие в моделируемой системе и их формулировка в виде прецедентов. Прецедент - это текстовое описание процессов, происходящих в предмет области.
2. ООА предмет области. Задача: определение видов деятельности участников процесса (и относящиеся к делу понятия) и составление концептуальной модели, которая отражает различных категории элементов предметной области.
3. Объектно-ориентированным называется проектирование, при котором основное внимание сосредоточено на распределении обязанностей, которое означает выделение задач и обязанностей различных программных объектов в приложении.

Предметной областью являются «Сельскохозяйственные машины», поэтому были выбраны наиболее популярные представители этой области, а именно: трактор, грузовик, автомобиль, комбайн, зерноуборочный комбайн, кормоуборочный комбайн, форвардер.

Так как все машины имеют некоторые общие параметры, например: марка, мощность, расход топлива, то нужен некоторый класс-интерфейс, который согласует работу с основными полями всех сельскохозяйственных машин для дальнейшей полиморфной обработки. Также нужен базовый класс, который будет содержать в себе основные поля для каждой машины. Классы самих инструментов могут наследоваться, так зерноуборочный комбайн будет наследником класса комбайн, т.к. является разновидностью комбайна, но имеет другую специализацию с другим оборудованием. В результате проведенного объектно-ориентированного анализа была построена контекстная диаграмма классов (рис. 1)

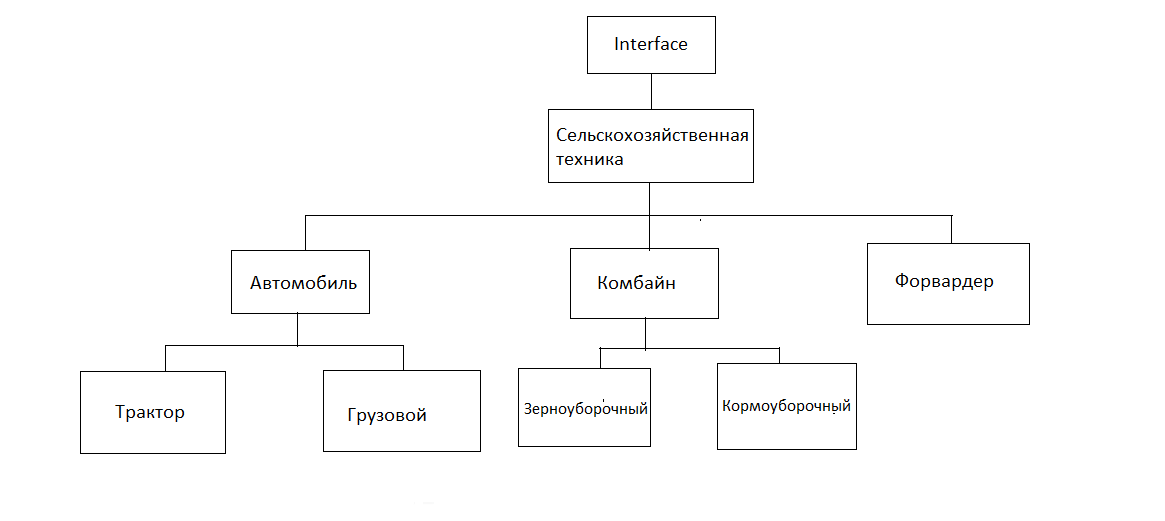


Рис. 1. Контекстная диаграмма классов

# Приложение 1. Техническое задание

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное

образовательное учреждение высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт Информационных Технологий

наименование института (факультета)

Кафедра математического и программного обеспечения ЭВМ

наименование кафедры

Программирование, модуль: ООП

наименование дисциплины в соответствии с учебным планом

УТВЕРЖДАЮ

Зав. кафедрой МПО ЭВМ,

д.т.н., профессор \_\_\_\_\_\_\_\_\_\_ Ершов Е.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ С++

Техническое задание к курсовой работе

Листов 5

Руководитель: Ершов Е. В.

Ф.И.О преподавателя

Исполнитель:

студент 1ПИб-01-21оп

группа

Скрыник К.А.

Фамилия, Имя, Отчество

2020 год

Введение

Тема курсовой работы: "Объектно ориентированное программирование на языке С++".

Цель курсовой работы: написание программы для представления предметной области “Сельскохозяйственные машины”.

Программа предназначена для защиты курсовой работы.

1. Основания для разработки

Основанием для разработки является задание на курсовую работу по дисциплине “ООП”, выданное на кафедре МПО ЭВМ ИИТ ЧГУ.

Общие требования к программному обеспечению для разработки следующие:

* Разработать иерархии родственных типов, корневой класс которых класс-интерфейс, для моделирования и обработки данных предметных областей набором отложенных методов - полиморфная обработка родственных объектов (согласно вариантов А.х.х и А.у.у). Организовать взаимодействие иерархий типов через указатели на интерфейсы и виртуальные функции (отложенные методы).
* Создать обобщенный (void\*) контейнерный класс (базовый) и от него, используя закрытое наследование, производный класс – шаблон для хранения указателей на абстрактный базовый класс - интерфейс (согласно варианта В.х).
* Для хранения объектов каждого производного класса использовать структуру данных (согласно варианта С.х).
* Реализовать функции обработки данных (сортировка и поиск по выбранным полям и задаваемым диапазонам значений, другие функции, в том числе перегруженные).
* Реализовать файловый ввод/вывод и ввод данных с клавиатуры, вывод данных на дисплей.
* Предусмотреть обработку различных исключительных ситуаций.
* Работа всех функций должна быть проверена и результаты проверки оформлены протоколом тестирования.

2. Назначение разработки

Функциональным назначением программы является представление предметной области «Сельскохозяйственные машины».

3. Требования к программе

3.1. Требования к функциональным характеристикам

* Разработать иерархию родственных типов, корневой класс которых – абстрактный базовый класс (интерфейс), для моделирования и обработки данных предметной области «Сельскохозяйственные машины»;
* реализовать файловый ввод/вывод данных на дисплей;
* обработка исключительных ситуаций при добавлении объектов с помощью пользовательской формы, и загрузке объектов из файлов;
* ввод данных с клавиатуры;
* создание иерархии родственных типов;
* создание обобщенного контейнерного класса и шаблонного контейнерного класса посредством закрытого наследования для хранения указателей на объекты;
* создание циклического списка для хранения объектов производных классов;
* реализация механизмов наследования;
* реализация функции полиморфной обработки данных: поиска по полю модель, получение информации об объекте.

3.2. Требования к надёжности

* Работоспособность элементов;
* Интуитивно понятный интерфейс;
* Наличие обработчиков исключительных ситуаций;
* Корректный вывод информации на экран.

3.3. Условия эксплуатации

Диск или флеш-накопитель хранить в сухом, недоступном для детей месте.

3.4. Требования к составу и параметрам технических средств

Требования к оборудованию

* Процессор с тактовой частотой 2400 MHz или более мощный;
* оперативная память 512 Мб или больше;
* свободное место на жёстком диске от 120 Мб;
* архитектура с разрядностью 32 бита или 64 бит;
* операционная система Windows 7, 8, 10;
* клавиатура;
* компьютерная мышь;
* монитор.

3.5. Требования к информационной и программной совместимости

Код программы реализуется на языке MS Visual C++ в среде .NET Framework 4.6.1 в системе разработки MS Visual Studio 2019. Данный продукт может быть использоваться в операционной системах Windows 7, 8, 10 при наличии установленного программного обеспечения.

3.6. Требования к маркировке и упаковке

Требования не предъявляются.

3.7. Требования к транспортировке и хранению

Для распространения и хранения приложения можно использовать любые носители: диски, флэш накопители, облачные хранилища. Хранение накопителей, которые содержат данное ПО должно производится в сухом месте, недоступном для детей месте.

4. Требования к программной документации

Программная документация должна содержать расчетно-пояснительную записку с содержанием:

1) Титульный лист.

2) Оглавление.

3) Введение.

4) Основная часть.

5) Заключение.

6) Список литературы.

7) Приложения (техническое задание, руководство пользователя, а также исходный текст программы).

5. Стадии и этапы разработки

Стадии и этапы разработки представлены в таблице П1.1.

Таблица П1.1

Стадии и этапы разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапа разработки | Сроки разработки | Результат выполнения | Отметка о выполнении |
| Разработка ТЗ | 15.03.20 – 23.03.20 | Оформленное ТЗ | ✓ |
| Анализ предметной области | 24.03.20 – 05.04.20 | Проведён анализ предметной области; изучены понятия | ✓ |
| Создание иерархии родственных классов | 06.04.20 – 14.04.20 | Иерархия классов разработана | ✓ |
| Проектирование приложения | 15.04.20-20.05.20 | Разработано приложение | ✓ |
| Тестирование приложения | 20.05.20-25.05.20 | Приложение протестировано, ошибки исправлены | ✓ |
| Оформление РПЗ | 26.05.20 – 31.05.20 | Оформлено РПЗ | ✓ |

6. Порядок контроля и приёмки

Проверка контроля проходит в соответствии с графиком стадий и этапов разработки:

* сдача технического задания и иерархии родственных типов;
* обработка исключительных ситуаций и тестирование программы;
* предоставление программы в электронном виде;
* сдача расчетно-пояснительной записки.