ГВУЗ "ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Кафедра программной инженерии

Утверждаю

Федяев О.И.

 \_\_\_\_\_\_\_\_\_\_\_

\_\_. 02.2019 г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

к курсовому проекту по дисциплине

"Объектно-ориентированное программирование"

на тему: "Объектно-ориентированная система

"Банковский счет. Операции с банковским счетом" "

Руководитель: Выполнил:

\_\_\_\_\_\_\_ каф. ПИ студент гр.ПИ 18-Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Моргунов Арсений

 \_\_. 02.2019 г. \_\_. 02.2019 г.

Донецк 2019

ВВЕДЕНИЕ

На современном этапе развития программного обеспечения все большая роль отводится объектно-ориентированному стилю проектирования и программирования (ООП). Это позволяет разрабатывать большие программные комплексы быстрее, используя повторно имеющие модули, обеспечивать инкапсуляцию данных, полиморфизм операций, наследование свойств и методов. Ведущие фирмы по разработке прикладного и системного ПО выполняют свои разработки средствами ООП. Современные операционные системы, например, платформа .NET имеет внутренний объектно-ориентированный язык и аппаратную поддержку отдельных его компонент. Поэтому изучение и практическое освоение специалистами по программной инженерии этого стиля программирования является актуальной задачей.

Профессиональное обучение программированию по направлению подготовки “Программная инженерия” предусматривает освоение языка Си в качестве базового. Дальнейшее развитие навыков программирования на языке Си и изучение основ ООП выполняется в дисциплине “Объектно-ориентированное программирование”. Для закрепления навыков проектирования и разработки больших программ в объектно-ориентированном стиле учебным планом направления подготовки “Программная инженерия” предусмотрено выполнение курсового проекта по этой дисциплине.

Целью курсового проекта является закрепление практических навыков самостоятельной постановки и решения задачи обработки данных с помощью ЭВМ средствами ООП. Во время выполнения курсового проекта студент должен овладеть методикой объектно-ориентированного проектирования программ по формализации и решению поставленной задачи, технологическими приемами разработки объектно-ориентированных программ на языке Си++.

Условием успешного выполнения курсового проекта являются практические навыки ООП, полученные при выполнении лабораторных работ по дисциплине “ООП”.

Разработанная программа Money.exe является объектно-ориентированной системой для предметной области “Банковский счет и операции с ним”. Она предназначена для использования в высших учебных заведениях с целью демонстрации знаний, приобретенных обучаемым в области технологии разработки объектно-ориентированных программ средствами языка Си++.

1 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Курсовой проект выполняется на основании “Задания на курсовой проект” по дисциплине “Объектно-ориентированное программирование” для студентов направления подготовки “Программная инженерия”, выданное кафедрой “Программная инженерия” ДонНТУ.

2 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Программное изделие Money.exe является информационной системой типа “Объектно-ориентированная система”.

Должно быть разработано программное, которое может работать в следующих режимах:

а) демонстрация описания классов Money и Account и работоспособности функций-членов класса:

1) конструкторов;

2) функций-членов GetCurrency();

3) функций-членов GetSum();

4) функций-членов AddMoney(long Sum);

5) конструктор с параметрами Account(char \* FIO, long Ident, Money m);

6) деструктора;

7) функций-членов SubMoney(longSum);

б) демонстрация работоспособности класса Money при перегрузке унарного оператора класса !;

в) демонстрация работоспособности классов Money и Account при новом действии – обнулить счет;

г) система помощи:

1) о программе (программный документ ”Описание программы”);

2) руководство оператора (программный документ “Руководство оператора”);

3) об авторе.

Программное изделие должно функционировать в среде MS Windows версии не ниже XP.

3 ТРЕБОВАНИЯ К ПРОГРАММНОМУ ИЗДЕЛИЮ

3.1 Требования к функциональным характеристикам

Программное изделие должно удовлетворять следующим общим требованиям:

а) использование технологии раздельной компиляции файлов;

б) режим работы монитора для программы – текстовый (консольное приложение);

в) использование файлов для хранения всех данных;

г) обеспечение двуязычия в общении с пользователем.

Критерии оценивания программного изделия по степени выполнения требований технологии раздельной компиляции:

а) для оценки «удовлетворительно»: отсутствие файла проекта, объединение исходных файлов с помощью директив препроцессора;

б) для оценок «хорошо» и «отлично»: использование файла проекта и технологии раздельной компиляции файлов;

Критерии оценивания программного изделия по степени выполнения требований для инкапсуляции:

а) для оценки «удовлетворительно»: реализованы только два класса, описание которых представлено в словесном описании предметной области;

б) для оценки «хорошо»:

1) реализованы классы, описание которых представлено в словесном описании предметной области;

2) реализованы дополнительные классы, расширяющие словесное описание предметной области.

в) для оценки «отлично»:

1) реализованы два класса, описание которых представлено в словесном описании предметной области;

2) реализованы дополнительные классы, расширяющие словесное описание предметной области и возможности интерфейса программы;

3) для каждого из классов предусмотрена возможность подсчета количества объектов, существующих в текущий момент в программе.

Критерии оценивания программного изделия по степени выполнения требований для наследования:

а) для оценки «удовлетворительно»: реализованы только два класса, описание которых представлено в словесном описании предметной области, и указанные отношения между ними;

б) для оценки «хорошо»:

1) реализованы классы, описание которых представлено в словесном описании предметной области, и указанные отношения между ними;

2) реализованы дополнительные классы - иерархия классов и/или контейнерные классы (только одиночное наследование);

в) для оценки «отлично»:

1) реализованы классы, описание которых представлено в словесном описании предметной области, и указанные отношения между ними;

2) реализованы дополнительные классы - иерархия классов и/или контейнерные классы (одиночное наследование);

3) реализовано множественное наследование.

Критерии оценивания программного изделия по степени выполнения требований для полиморфизма:

а) для оценки «удовлетворительно»: реализована перегрузка только тех операторов, описание которых представлено в словесном описании предметной области;

б) для оценки «хорошо»:

1) реализована перегрузка операторов, описание которых представлено в словесном описании предметной области;

2) реализована перегрузка операторов для некоторых дополнительных классов;

3) перегрузка операторов выполнена как внутри, так и вне класса.

в) для оценки «отлично»:

1) реализована перегрузка только тех операторов, описание которых представлено в словесном описании предметной области;

2) реализована перегрузка операторов для всех дополнительных классов;

3) перегрузка операторов выполнена как внутри, так и вне класса.

4) в реализации полиморфизма использованы абстрактные классы и виртуальные функции.

Критерии оценивания программного изделия по степени выполнения требований к демонстрации работы системы:

а) для оценки «удовлетворительно»: все результаты работы выводятся в системный поток вывода;

б) для оценок «хорошо» и «отлично»:

1) результаты работы выводятся в системный поток вывода;

2) состояние объектов системы выводится в файловый поток вывода.

Критерии оценивания программного изделия по степени выполнения требований в режиме помощи:

а) для оценки «удовлетворительно»: использование файла с условием задачи и файла помощи;

б) для оценок «хорошо» и «отлично» - использование файлов:

1) о программе (программный документ “Описание программы”);

2) руководство оператора (программный документ);

3) об авторе.

Критерии оценивания программного изделия по степени выполнения требований по организации работы с файлами:

а) для оценки «удовлетворительно»:

1) путь доступа файлам данных установлен программным путем (является константой в программе);

б) для оценки «хорошо»:

1) путь доступа к файлам определяется пользователем на этапе выполнения программы (наличие в главном меню проекта опции настройки, в которой пользователь указывает путь доступа к файлам данных и/или их имена);

в) для оценки «отлично»:

1) путь доступа к файлам определяется пользователем на этапе выполнения программы (наличие файла конфигурации с данными о пути доступа к файлам данных и/или их именах);

2) предусмотрена обработка исключительных ситуаций при работе с файлами.

Критерии оценивания программного изделия по степени выполнения требований по обеспечению двуязычия интерфейса пользователя:

а) для оценки «удовлетворительно»:

1) приложение имеет фиксированную пару языков общения (тексты надписей являются константами в программе);

2) переключение языков реализовано только в одном кадре интерфейса;

3) двуязычие реализовано не для всех режимов работы;

б) для оценок «хорошо» и «отлично»:

1) приложение имеет произвольную пару языков общения (тексты надписей читаются из файлов);

2) переключение языков реализовано на каждом кадре интерфейса;

3) двуязычие реализовано для всех режимов работы;

3.2 Требование к надежности

Программное изделие для обеспечения надежности функционирования должно:

а) проверять наличие всех файлов данных;

б) обеспечить минимизацию количества информации, вводимой пользователем;

в) контролировать корректность ввода данных пользователем;

г) обрабатывать исключительные ситуации, вызванные не корректностью ввода данных пользователем с целью предотвращения прерывания выполнения программы.

3.3 Условия эксплуатации

Пользователь должен иметь квалификацию не ниже “оператор ПЭВМ”.

Для эксплуатации программного изделия необходимо наличие программиста, в функции которого входит внесение изменений в информационную часть системы.

3.4 Требования к составу и параметрам технических средств

Для функционирования программного изделия необходим персональный компьютер со стандартным набором периферийных устройств (монитор, клавиатура, мышь).

3.5 Требования к информационной и программной совместимости

Курсовой проект должна быть выполнена на языке Си++ в операционной системе MS Windows версии не ниже XP.

4 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Пояснительная записка по курсовому проекту должна сопровождаться следующими программными документами:

а) описание программы;

б) руководство оператора;

в) текст программы.

5 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Перечень этапов и сроки приведен в табл.5.1

Таблица 5.1 - График выполнения курсового проекта

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование этапа | Срок выполнения | |
| Неделя | Дата |
| 1 | Выдача задания на курсовой проект. Уточнение задачи. | 1 | 05.02.19 |
| 2 | Постановка задачи. Определение требований к программе.  а) запись постановки задачи;  б) составление технического задания и его утверждение. | 1-2:  2  2 | 12.02.19-26.02.19  12.02.19-19.02.19 |
| 3 | Объектно-ориентированный анализ и проектирование:  а) объектно-ориентированный анализ;  б) проектирование структуры классов;  в) проектирование отношений между классами;  г) проектирование структуры объектов и их поведения. | 3-8:  3-4  4-5  5-6  6-8 | 19.02.19-26.03.19  19.02.19-  26.02.19;  26.02.19-  05.03.19;  05.03.19-  12.03.19;  12.03.19-  26.03.19 |
| 4 | Техническое проектирование:  а) модульный анализ: определение структуры программы, ее модулей и их взаимосвязей;  б) разработка основного алгоритма функционирования программы;  в) создание спецификации модулей. | 4-6  4  5-6  6 | 26.02.19-  12.03.19  26.02.19-  05.02.19  05.03.19-  12.03.19  12.03.19-19.03.19 |
|  | Рабочее проектирование:  а) определение структур данных и разработка алгоритмов работы модулей;  б) разработка классов:  1) реализация инкапсуляции;  2) реализация наследования;  3) реализация полиморфизма. | 5-10:  5-6    6-7  7-9  9-10 | 05.03.19-  09.04.19  05.03.19-  12.03.19  12.03.19-  26.03.19  26.03.19-  09.04.19  09.04.19-  16.04.19 |
| 6 | Написание программы | 9-12 | 09.04.19-  30.04.19 |
| 7 | Настройка программы | 11-12 | 23.04.19-  30.04.19 |
| 8 | Комплексная отладка и тестирование | 12 | 23.04.19-  30.04.19 |
| 9 | Написание пояснительной записки | 13 | 30.04.19-07.05.19 |
| 10 | Защита курсового проекта | 14 | 07.05.19-14.05.19 |

6 ПОРЯДОК КОНТРОЛЯ И ПРИЕМА

Курсовой проект выполняется 14 недель. Пояснительная записка по курсовому проекту предоставляется на проверку преподавателю не менее чем за 3 рабочих дня до даты защиты.

Защита происходит в присутствии комиссии в составе 2-3 человек и включает:

а) доклад (до 3 минут), отражающий все этапы выполнения курсового проекта;

б) презентацию программы (с демонстрацией всех ее функций);

в) ответы на вопросы комиссии.

7 СЛОВЕСНОЕ ОПИСАНИЕ ПPЕДМЕТНОЙ ОБЛАСТИ

На первом этапе проектирования информационной системы необходимо выполнить анализ предметной области, определить объекты предметной области.

Предположим, что подход к исследованию предметной области будет основываться на выявлении повседневных или глобальных задач (проблем) той или иной аудитории. В таком случае можно по ходу выделить минимальный набор объектов предметной области для их описания.

Воспользуемся комбинированием этого подхода с подходом, который подразумевает нахождение объектов с таким расчетом, чтобы их можно было использовать при решении наибольшего количества задач предметной области.

Свойства объекта «Money»:

- поле «валюта» char \* Currency;

- поле «сумма» long Sum;

Расширенный набор свойств будет варьироваться в зависимости от назначения субкласса, который унаследует данные от основного класса.

8 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Титульный лист

Реферат

Содержание

Введение

1 Постановка задачи

2 Объектно-ориентированный анализ и проектирование

2.1 Объектно-ориентированный анализ

2.2 Объектно-ориентированное проектирование

2.2.1 Диаграмма классов

2.2.2 Диаграмма объектов

2.2.3 Диаграмма последовательности

2.2.4 Диаграмма состояний

3 Рабочее проектирование

3.1 Модульная структура программы

3.2 Структура файлов и каталогов программы

3.3 Разработка классов

3.3.1 Реализация наследования

3.3.2 Реализация полиморфизма

3.3.2.1 Перегрузка операторов

3.3.2.2 Абстрактные классы

3.3.2.3 Виртуальные функции и чистые виртуальные функции

3.4 Тестирование работоспособности системы классов

3.5 Обоснование выбора структур данных

3.6 Обоснование выбора алгоритмов

3.6.1 Интерфейс

3.6.2 Работа с файлами

3.6.3 Подсчет количества экземпляров класса

Выводы

Перечень ссылок

Приложение А. Техническое задание

Приложение Б. Описание программы

Приложение В. Руководство программиста

Приложение Г. Руководство оператора

Приложение Д. Текст программы

Приложение Е. Файлы исходных данных

Приложение Ж. Файлы выходных данных

Приложение К. Экранные формы