

Лабораторная работа №21

Изучение процесса разработки дочерних классов на C++

1 Цель работы

- 1.1 Изучить процесс разработки дочерних классов на C++;
- 1.2 Изучить реализацию механизмов наследования, инкапсуляции и полиморфизма на языке C++;
- 1.3 Закрепить навык разработки программ методами структурного, процедурного и модульного программирования.

2 Литература

2.1 Ашарина, И. В. Объектно-ориентированное программирование в C++: лекции и упражнения. Учебное пособие для вузов. / И. В. Ашарина. – 2-е изд., перераб. и доп. – Москва : Горячая Линия–Телеком, 2017. – 336 с. – URL: <https://ibooks.ru/reading.php?productid=359752>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный. – гл.21.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создание родительского и дочернего классов

5.1.1 Создать родительский класс `Person` (человек) с закрытым строковым полем `name` (имя) и защищенным числовым полем `age` (возраст) для хранения информации о человеке.

5.1.2 Добавить в класс `Person` следующие открытые конструкторы:

- конструктор по умолчанию для задания стандартных значений полям класса;
- конструктор с двумя параметрами для задания значений всем полям класса.

5.1.3 Создать класс `Worker` (работник) с закрытыми полями `seniority` (стаж) и `salary` (зарплата). Сделать его наследником класса `Person`.

5.1.4 Добавить в класс `Worker` следующие открытые конструкторы:

- конструктор по умолчанию для задания стандартных значений полям класса;
- конструктор с четырьмя параметрами для задания значений всем полям класса. После списка параметров вызвать родительский конструктор с параметрами.

5.1.5 Протестировать полученный дочерний класс, создав объекты класса с использованием конструктора по умолчанию и конструктора с параметрами. Данные для конструктора задать программно.

5.2 Создание методов в дочернем классе

5.2.1 Добавить в родительский класс `Person` открытый виртуальный метод `Print`, выводящий на экран информацию об объекте.

- 5.2.2 Добавить в дочерний класс `Worker` следующие открытые методы:
- метод `Print`, вызывающий реализацию метода `Print` из родительского класса и расширяющий его функциональность выводом информации о полях, объявленных в классе `Worker`;
 - метод `GetStartWorkAge`, возвращающий возраст, в котором человек устроился на работу (находится вычитанием стажа из возраста человека).

5.2.3 Протестировать работу методов, вызвав их у созданных объектов класса.

5.3 Создать абстрактный класс Фигура

5.3.1 В созданном классе объявить следующие чисто виртуальные функции:

- функцию для нахождения площади фигуры;
- функцию для нахождения периметра фигуры;
- функцию для вывода названия типа фигуры.

5.3.2 Создать два класса-наследника класса `Фигура`, в которых реализовать указанные функции и добавить требуемые поля (по вариантам):

№ варианта	Задание
1	классы <code>Квадрат</code> (поле <code>сторона</code>) и <code>Прямоугольник</code> (поля <code>длина</code> и <code>ширина</code>)
2	классы <code>Круг</code> (константа <code>Pi</code> и поле <code>радиус</code>) и <code>Прямоугольник</code> (поля <code>длина</code> и <code>ширина</code>)
3	классы <code>Квадрат</code> (поле <code>сторона</code>) и <code>Круг</code> (константа <code>Pi</code> и поле <code>радиус</code>)

5.3.3 Протестировать работу методов, создав объекты классов-наследников и вызвав переписанные методы.

6 Порядок выполнения работы

6.1 Используя `Microsoft Visual Studio`, создать проект `C++` и выполнить задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «наследование»?

8.2 Что такое «множественное наследование»?

8.3 Какие классы называются родительскими?

8.4 Какие классы называются дочерними?

8.5 Что такое «виртуальный метод»?

8.6 Какие классы называются абстрактными?

8.7 Для чего используются абстрактные классы?