**Методические указания по прохождению практики для студентов 1 курса специальности ПИ**

**1 Общие сведения о прохождении практики**

Практика – это особый вид учебной работы студента. Практика предполагает: получение студентом определенного задания, выполнение этого задания, оформление отчета о прохождении практики, сдачу и защиту выполненной работы. Выполнение практики похоже на выполнение курсового проекта. Но курсовой проект выполняется в течение учебного семестра, этапы курсового проекта обычно соответствуют определенным лабораторным работам, материал, необходимый для выполнения курсового проекта, обычно разбирается на занятиях с преподавателем. Практика же выполняется после окончания семестра, учебных занятий по расписанию во время прохождения практики не предусмотрено. Задание по практике обычно подразумевает работу с новой технологией программирования, с новой системой программирования, решение новых для студентов типов задач и т.п. Соответствующий материал детально на занятиях с помощью преподавателя не изучается. Итак, практика – это, прежде всего, самостоятельная работа студента, в процессе с которой он должен разобраться с новой для себя областью знаний, решить новую для себя проблему и т. д. Безусловно, в процессе прохождения практики предусмотрены консультации преподавателя.

**2 Задание на 1 учебную практику (ознакомительную практику)**

Студенты специальности ПИ после окончания второго семестра проходят 1-ую учебную практику (ознакомительную практику). База проведения практики – АлтГТУ, кафедра ПМ. Практика посвящена разработке графических интерфейсов в среде Microsoft Visual Studio. Время прохождения практики – 4 недели. По результатам прохождения практики студенту выставляется итоговая оценка «отлично», «хорошо», «удовлетворительно» и соответствующий рейтинг по 100-балльной шкале.

**Цель работы**: изучение принципов разработки графических приложений, получение навыков разработки приложений с графическим интерфейсом в среде Microsoft Visual Studio.

**Главная задача:** создание программного продукта с удобным графическим интерфейсом, позволяющего администратору в защищенном режиме работать с базой данных определенного назначения, а рядовому пользователю - эффективно взаимодействовать с ней.

Приведенные выше формулировки цели и главной задачи можно **включить в раздел «Введение»** в отчет по практике.

Практика включает в себя выполнение 2 заданий: основного и вспомогательного.

Основное задание: реализовать работу, выполненную на курсовом проектировании во 2 семестре, с визуальным графическим интерфейсом. Все основные виды работ: хранение данных в файле, диалоговое меню, ввод, редактирование, вывод, универсальный запрос, защиты и т.д., - должны присутствовать и в этой работе. Но жестких требований к организации взаимодействия с файлом, как это было в курсовом проекте, нет. То есть можно работать с файлом так, как требовалось в курсовом проекте, а можно копировать всю базу в массив, так как основной акцент в практике – не работа с файлом, а организация графического интерфейса. Желательно реализовать два режима работы: администратор (вход по паролю) и пользователь. Функции создания и редактирования базы доступны только администратору.

Дополнительное задание: реализовать игру с графическим интерфейсом. Можно реализовывать ту же игру, которая была в 7 лабораторной работе во 2 семестре, можно – другую. Требования обязательно что-то хранить в файле (как это было в 7 работе), нет, по причине, озвученной выше. Игра может быть совсем простой, но некоторые элементы графического интерфейса должны обязательно присутствовать.

Итоговое приложение должно содержать минимум две формы: одна основная форма – работа с базой, а вторая форма, с игрой, может активироваться в процессе основной работы по желанию пользователя (пользователь хочет передохнуть, немного поиграть, а потом опять вернуться к основной работе).

Выполнение основного задания – обязательно. Если в итоге студент выполнит основное задание, но не сделает дополнительное, максимальная оценка будет «хорошо». Если же студент выполнит дополнительное задание, но не выполнит основное, то практика вообще не зачитывается. Поэтому рекомендуется вначале полностью выполнить основное задание, а только после этого приступать к дополнительному заданию.

При выполнении практики потребуется использовать знания по всем разделам языка С, которые рассматривались в процессе изучения дисциплины «Программирование» /1/. Кроме того, студенту необходимо самостоятельно познакомиться с технологией разработки графических интерфейсов в среде Microsoft Visual Studio. Начальное знакомство рекомендуется начать с книги Пахомова Б. И. «C/C++ и MS Visual C++ 2010 для начинающих» /2/. Можно, естественно, взять книгу этого же автора для более поздних версий студии, можно использовать другие источники. В процессе работы настоятельно рекомендуется пользоваться информационным сервисом для разработчиков MSDN /3/ и другими электронными ресурсами. Кроме того, студенту необходимо самостоятельно решить некоторые вопросы, связанные с переходом от языка С, изучаемого на 1 курсе, к языку С++, который используется при работе с визуальными компонентами. Некоторые возникающие в связи с этим проблемы приведены в 5 разделе данных методических указаний.

По результатам выполненной работы необходимо оформить **отчет** (пояснительную записку). Отчет по практике похож на отчет по курсовому проекту. Формат титульного листа и задания будут уточнены позже. В процессе оформления рисунков, диаграмм и прочее рекомендуется познакомиться с UML (универсальный язык моделирования).

При выполнении практики рекомендуется также познакомиться с системой контроля версий Git.

РЕКОМЕНДАЦИЯ. Главное – выполнить основное задание. На первом этапе надо разобраться с визуальным интерфейсом, и на эту работу уйдет примерно неделя. Потом реализуйте основное задание, особенно не заботясь о защитах и работе с файлом. Только потом занимайтесь подробно защитами, файлами, игрой, читайте о Git и прочее.

**3 Комплект материалов, предоставляемых в электронном формате**

Как результат выполнения практики необходимо сдать следующие материалы в виде zip.файла:

1) папку с проектом, содержащим \*.exe файл;

2) отчет (пояснительную записку) о выполненной практике (оформление отчета похоже на отчет по курсовой работе, некоторые особенности см. подраздел 7 данных методических указаний) в виде \*.doc файла;

3) шаблон отзыва о прохождении практики в виде \*.doc файла;

4) заявление о приеме практики в дистанционном формате в виде \*.doc файла (pdf, jpg).

Будьте ВНИМАТЕЛЬНЫ!!! При использовании образцов титульного листа, задания, заявления, отзыва вставьте свои данные! Например, в титульном листе надо указать свою ФИО, группу и т.д.

На титульном листе, на листе с заданием и на заявлении должна быть ПОДПИСЬ студента. Проще всего сфотографировать свою подпись и вставить затем в соответствующие \*.doc файлы.

**4 Краткая информация о разработке графических интерфейсов в среде Microsoft Visual Studio**

Создание проекта, например в Visual 2010:

1) выполнить команду File \ New \ Project;

2) выбрать тип приложения:

Visual C++; CLR

Windows Forms Application.

После создания проекта Вы увидите на рабочем столе объект Form1, который называется формой. Это экземпляр класса Form. Далее в форму надо поместить компоненты (кнопки, метки и др.), которые программист планирует использовать в соответствии с решаемой задачей, выбирая их из палитры (набора) компонентов, расположенных в окне вкладки Toolbox.

Основные компоненты:

1) Label – метка;

2) Textbox – окно для ввода-вывода строчных данных;

3) Button – кнопка.

Существует много других компонентов, которые будут полезны при выполнении практики, например:

DataGridView – для работы с таблицами;

MenuStrip – для организации главного «ленточного» меню в виде полоски;

MaskedTextbox – позволяет ввести ограничения на вводимые символы;

ListBox – для «выпадающих» списков;

CheckBox и RadioBox – флажки-переключатели по типу «вкл/выкл».

Каждый компонент характеризуется свойствами Properties и событиями Events, которыми можно воздействовать на компонент. При написании кода основная задача программиста состоит в написании программ – обработчиков событий.

Пошаговую инструкцию на начальных этапах работы смотрите в книге Б. Пахомова /2/ или в др. источниках. Рекомендуется в плане ознакомления разработать приложение «Простейший калькулятор».

**5 Дополнительные проблемы**

5.1 Преобразование типов

Вместо простых типов int, double, char лучше использовать типы, наследуемые от базового типа Object: классы Int32, Int64, Double, Char, String и др. Преобразование этих типов можно произвести с помощью класса Convert или метода Parse (TryParse).

5.2 Запись / чтение из файла

Лучше всего использовать классы:

Console,

StreamWriter (далее метод WriteLine),

StreamReader (далее метод ReadLine).

5.3 Защиты

Некоторые защиты можно установить на этапе выбора визуальных компонентов и их свойств. Смотрите, например, компонент MaskedTextbox, который позволяет ввести ограничения на вводимые символы (только цифры, только буквы и т.д.), а также на длину вводимой строки. Используется для ввода дат, времени, номеров телефонов и т.д.

Для правильного реагирования на ошибки, возникающие в ходе выполнения программы, в языке С++ используется механизм обработки исключительных ситуаций. Этот механизм основан на 3 ключевых словах: try – пытаться, catch – перехват (ошибки), throw – генерировать (ошибку).

5.4 Указатели

Здесь нет никакой проблемы, просто надо иметь в виду, что при работе с визуальными компонентами есть некоторые особенности при объявлении указателей и выделении памяти.

При создании приложений с визуальным интерфейсом мы работаем в специальной среде CLR. В этой среде используются так называемые регулируемые указатели. Для них принято специальное обозначение: вместо символа «\*» при объявлении такого указателя применяется символ «^». Память выделяется не функцией malloc(), а ее аналогом – утилитой gcnew.

Пример объявления указателя p на класс Form2:

Form2^ p = gcnew Form2();

**6 Дополнительные системы и сервисы**

В процессе прохождения практики рекомендуется познакомиться с некоторыми системами и сервисами, которые программисты активно используют в своей работе, а именно MSDN, UML, Git. В принципе, использование MSDN – обязательно в процессе работы, а с понятиями UML и Git рекомендуется хоты бы ознакомиться. Если что-то из указанных систем будет использовано в процессе выполнения практики, следует отметить это в отчете в первой части раздела «Заключение».

6.1 MSDN

MSDN (Microsoft Developer Network) – подразделение компании Майкрософт, ответственное за взаимодействие фирмы с разработчиками. Сайт MSDN содержит подробнейшую справку по всем инструментам и библиотекам Microsoft. Очень удобно то, что в MSDN есть примеры на С++ (обычно внизу страницы).

Официальный сайт MSDN на русском языке:

<http://msdn.microsoft.com/ru-ru/ms348103>.

Если Вы хотите быстро узнать какую-то конкретную информацию, лучше использовать поисковик. Например, если Вы хотите узнать информацию о butten, то просто наберите в строке поискового запроса:

butten msdn

6.2 UML

UML (Unified Modeling Language) – универсальный язык моделирования. Это – язык графического описания моделей в области разработки программного обеспечения.

Язык UML – это не язык программирования. Это визуальный графический язык, который предназначен для визуализации (для наглядного представления) и документирования информационных систем и бизнес-процессов с ориентацией на их последующую реализацию в виде программного обеспечения.

Язык UML позволяет описывать модели сложных систем с помощью специальных диаграмм. Существует несколько видов диаграмм, основные из которых:

1) диаграмма вариантов использования;

2) диаграмма взаимодействия;

3) диаграмма классов;

4) диаграмма состояний;

5) диаграмма деятельности;

6) диаграмма компонентов;

7) диаграмма размещения.

Отметим диаграмму деятельности (activity diagram). Она позволяет детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Чаще всего эти диаграммы иллюстрируют сложные алгоритмы (что-то типа структурных схем).

Для построения UML-диаграмм существует много пакетов и интегрированных сред (IDE), как коммерческих, так и свободно распространяемых. Можно набрать в поисковой строке «UML IDE» и подобрать для себя подходящую систему. Широко используется, например, онлайн редактор диаграмм

<https://creately.com/>

6.3 Git

Git - это распределенная система контроля версий (version control system – VCS). Контроль версий означает то, что вы храните все версии редактируемых документов и можете вернуться к любой сохраненной версии в любой момент времени. Другими словами, это программное обеспечение для управления версиями файлов. С его помощью можно хранить хронологию всех изменений документа с комментариями (при их наличии) и возможностью в любой момент вернуть нужную версию файла.

**7 Оформление пояснительной записки (отчета) по практике**

Пояснительная записка по практике, или отчет, должна включать следующие структурные единицы:

- титульный лист;

- задание;

- содержание;

- перечень условных обозначений, символов, терминов (при необходимости);

- введение;

- основная часть, разделенная на нумерованные разделы, подразделы, пункты, подпункты;

- заключение;

- список использованных источников;

- приложения.

Каждая структурная единица отчета начинается с нового листа. В основной части, кроме того, каждый раздел начинается с нового листа.

Основная часть отчета должна состоять минимум из 3 обязательных разделов, посвященных постановке задачи, проектированию программного обеспечения (ПО), реализации программного обеспечения. Точные наименования разделов автор устанавливает самостоятельно.

В 1-ый раздел отчета, в части, посвященной обзору предметной области, следует отметить два момента: 1) работу со структурированными списками данных, с базами данных; 2) создание визуальных графических интерфейсов.

В 3-ий раздел отчета, посвященный реализации программного обеспечения, следует добавить подраздел с примерным названием «Описание реализации ПО». В этом подразделе должно содержаться описание того, какие визуальные компоненты и с какими целями были использованы, какие обработчики событий были написаны.

# **Список использованных источников**

1. Егорова Е.В. Программирование на языке СИ. Учебное пособие/Алт. Госуд. Технич. Ун-т им. И.И.Ползунова.-Барнаул:2013.-184с.
2. Пахомов Б.И. C/C++ и MS Visual C++ 2010 для начинающих. – СПб.: БХВ-Петербург, 2011. – 736с.
3. “MSDN” – информационный сервис для разработчиков [Электронный ресурс] – Режим доступа http://msdn.microsoft.com/ru-ru/ms348103/library, свободный.