**Министерство науки и высшего образования Российской Федерации**

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

**(КНИТУ-КАИ)**

**Институт Компьютерных технологий и защиты информации**

**Кафедра Динамики процессов и управления**

**ОТЧЕТ**

**Лабораторная работа по ООП**

Направление подготовки/специальность:

**09.03.03 «Прикладная информатика»**

 Выполнил:

обучающийся гр. 4217 Замалетдинов Б.М..

*(группа)          (подпись практиканта)*

Лабораторная работа защищена с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_

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

**Казань**

**2020 год**

**Содержание**

[**Постановка задачи:** 3](#_Toc59377600)

[**Этапы выполнения работы:** 3](#_Toc59377601)

[**Описание файлов** 3](#_Toc59377602)

[**1.** **Класс Computer** 3](#_Toc59377603)

[**2.** **Дочерние классы Game и Notebook** 3](#_Toc59377604)

[**3.** **Класс Library** 4](#_Toc59377605)

[**4.** **Класс Reader** 4](#_Toc59377606)

[**5.** **Класс Operation** 4](#_Toc59377607)

[**6.** **Класс ActionsReader** 4](#_Toc59377608)

[**7.** **Класс LibraryWithEvents** 5](#_Toc59377609)

[**8.** **База данных** 6](#_Toc59377610)

[**9.** **Program** 6](#_Toc59377611)

[**Литература** 7](#_Toc59377612)

# **Постановка задачи:**

Создать программу работы компьютерного клуба. Программа должна иметь класс компьютер, ее дочерние классы в виде игрового компьютера и ноутбука. Необходимо создать класс библиотеки, которая выступает хранилищем компьютеров. Проект должен создавать и обрабатывать операции с компьютерами для каждого клиента компьютерного клуба в многопоточном режиме. Программа должна выполнять следующие операции: включить компьютер, начать пользование, завершить пользование, выключить компьютер. Дальше, с помощью класс библиотеки с событиями записывать операции в журнал событий, а также в ХML файл. Выводить в консоль журнал событий, операции по отдельному клиенту и XML файл. Также создать Базу Данных, где создать таблицы для компьютеров, клиентов, типов компьютеров, типов операций и для самих операций. Через библиотеку с событиями заполнять таблицы компьютеров, клиентов и операций. Выводить отчет по операциям из БД в консоль программы.

# **Этапы выполнения работы:**

1. Создание абстрактного класса Computer
2. Создание двух дочерних классов Game и Notebook
3. Создание интерфейса IActions
4. Создание и настройка библиотеки WareHouse
5. Создание класса читателя Clients
6. Создание класса Operation
7. Создание класса ActionsClients с описанием событий
8. Создание класса WareHouseWithEvent на основе WareHosue
9. Создание БД с таблицами
10. Создание Program для работы всей программы

**Описание файлов**

1. **Класс Computer**

Класс может принимать следующие значения:

* String Maker - Производитель
* Enum TypeProcessor – Перечисление процессоров
* Int Ram – Количество Оперативной памяти
* Enum TypeDrive – Перечисление накопителей

Содержит конструктор с вышеперечисленными данными. Конструкторы формируют ID компьютера. Имеет метод GetInfo() для вывода информации о компьютере

1. **Дочерние классы Game и Notebook**

Game – класс игровых компьютеров. Может принимать следующие значения:

* Enum TypeVideocard – Перечисление видеокарт
* Double Power – Производительность видеокарты (ТераФлопсы)

Notebook – класс ноутбуков. Может принимать следующие значения:

* Double Diagonal – диагональ ноутбука
* Double Duration – время работы ноутбука

Оба класса имеют конструктор, метод GetInfo() для вывода информации.

1. **Интерфейс IActions**

Интерфейс IActions описывает следующие методы:

* Used() –Использовать компьютер
* Upgrade() – Улучшить компьютер

Данные методы реализованы в базовом классе Computer.

1. **Класс WareHouse**

Класс WareHouse создан на основе класса Storage<Computer>. В классе описаны следующие методы:

* AddComputer(Computer) - метод добавления компьютера.
* DeleteComputer (Computer) - удаления компьютера из склада
* ViewGameByTypeVideocard(TypeVideocard) – выборка игровых компьютеров по их типу видеокарт
* ViewNotebookByDuration(Duration)– выборка ноутбуков по их времени работы

1. **Класс Clients**

Класс принимает единственное значение FullName - Имя. Используется как источник для операций. Реализует интерфейс IActions.

1. **Класс Operation**

В начале перечисляются типы операций TypeOperation, в них входит:

* On – включить компьютер
* Off – выключить компьютер
* StartUse– начать пользование компьютером
* StopUse – завершить пользование компьютером

Класс используется для формирования операций. Он принимает следующие значения:

* Clients – клиенты, источник события
* Computer – компьютер, с чем взаимодействуют
* Enum TypeOperation – тип операции
* DateTime timeop – фиксирует время операции
* String Message – сообщение с типом операции

Класс имеет конструктор и метод ToString().

1. **Класс ActionsClient**

Данный класс формирует события взаимодействия читателей с книгами.

Принимает и формирует следующие значения:

* Clients – клиент, источник события
* EventClient(Operation) – делегат обработчика событий
* WareHouseWithEvents – склад с событиями
* Computer – компьютер, с которой взаимодействуют
* Int Timing – целое число, которое обозначает длительность операций
* Random rnd – генератор случайных чисел

Имеет два конструктора, пустой и полный, а также следующие методы:

* InitEvent() – регистрация обработчика событий OnEventComputer();
* Use(Clients, Computer, int interval) – описание действия использования компьютера. Создает операцию, выполняет два ее типа StartUse и StopUse с заданным временным интервалом между ними.
* Off(Clients, Computer) – описание действия выключения компьютера. Создает операцию и выполняет ее тип Off.
* On(Clients, Computer) - описание действия включения компьютера.Создает операцию и выполняет ее тип On.
* DoActions() – генерация событий. Сначала используют метод On(), затем Use(), и в конце Off(). Между методами поток спит указанный отрезок времени.

1. **Класс WareHouseWithEvents**

Класс является дочерним классом WareHouse. Принимает следующие типы значений:

* List<Operation> - журнал операций

Значения, связанные с БД:

* SqlConnection – подключение к базе данных SQL сервера
* SqlDataAdapter – заполняет DataSet данными из базы данных
* SqlCommandBuilder - генерирует однотабличные команды, которые используются для согласования изменений, внесенных в DataSet, со связанной базой данных SQL
* DataSet – хранилище таблиц DataTable, с которыми можно работать независимо от подключения
* DataTable – одна таблица данных в памяти

Класс имеет следующие методы:

* OnEventComputer(Operation) – обработчик событий. В нем записываются операции в журнал событий. Также из данных операций заполняются таблицы DataTable Computer, Clients и Operations
* GetOperationsWithTimeInterval(beginDataTime, endDataTime) – выборка операций по заданному временному интервалу.
* GetJournalForClient(Clients) – выборка операций по выбранному читателю
* WritetoXml\_Journal(NameFile) – создает XML файл, записывает туда значения операций
* ReadXml\_Journal(NameFile) – Просмотр файла
* ID\_TypeComputer(Computer) – метод позволяет узнать тип компьютера

Методы, связанные с БД:

* InitDB() – Присоединяет SQL сервер. Заполняем DataSet через SqlDataAdapter с помощью метода Fill()
* QuitDB() – с помощью метода Update() обновляем DataAdapter и закрываем соединение.
* ViewDS(DataSet) – просмотр таблиц DataSet и вызов метода ViewDataTable()
* ViewDataTable(DataTable) – просмотр все ячеек таблиц DataTable
* ListDB() – Вывод отчета по операциям в базе данных. Вместо id для наглядности отчета подставляются значения.
* ListDBForClient(Clients) - Вывод отчета по операциям в базе данных для отдельного клиента
* ListDBInterval(DateTime) – Вывод отчета по операциям за определенный интервал времени
* CleanOperations() – очистка таблицы операций на определенное количество строк

1. **База данных**

Создаем локальный сервер MS SQL CompDB.mdf

Создаем следующие таблицы данных:

* Computer – таблица со следующими полями:
* IdComp – int
* IdTC - int
* Maker – nchar(10)
* Processor – nchar(10)
* Events – таблица со следующими полями:
* IdEv - int
* Name – nchar(10)
* Operations – таблица со следующими полями:
* IdOp – int
* IdComp– int
* IdCl – int
* IdEv – int
* DateTime – datetime
* Message – nchar(20)
* Clients – таблица со следующими полями:
* IdCl - int
* Name – nchar(20)
* TypeComputer – таблица со следующими полями:
* IdTC - int
* Type – nchar(10)

1. **Program**

В начале программы мы вводим следующие значения:

* cs[] – массив с книгами Book
* cl[] – массив с читателями Reader

Далее мы создаем WareHouseWithEvents - Wh, Thread – thArray[cl.length] и ActionsReader – acts[cl.length]

Затем мы используем следующие методы:

* Wh .InitDB() – запускаем работу с БД
* acts .InitEvent() – регистрируем обработчик событий
* thArray.Start() – начинаем поток действий ActionsClient, начинается выполнение операций.
* Wh.GetJournalForClient(cl[i]) – Выводим журнал для клиентов
* Wh.WriteToXml\_journal(“out.xml”) – Записываем журнал в файл XML
* Wh.ReadToXml\_journal(“out.xml”) – Выводим в консоль файл XML
* Wh.ListDB() – выводим журнал событий из базы данных
* Wh.ListDBForClient(cl[i]) – выводим журнал событий из базы данных для определенных клиентов
* Wh.ListDBInterval – выводим журнал событий из базы данных для определенного интервала времени
* Wh.QuitDB() – закрываем и записываем изменения в базу данных.

# **Литература**

1. Дрейер М. Д73 C# для школьников: Учебное пособие / М. Дрейер. Перевод с англ. под ред. В. Биллига—М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2009. — 128 с.: ил.,

Глава 2 «Учимся общаться с компьютером», Глава 4 «Базы данных и XML»

1. Книга. Троелсен А., ЯЗЫК ПРОГРАММИРОВАНИЯ C# 5 И ПЛАТФОРМА .NET 4.5, Глава 8 «Работа с интерфейсами», глава 10 «Делегаты и события» и глава 24 «Введение в LINQ to XML»
2. Cтиллмен Э. «Изучаем С#» 3-e изд.
3. Сайт. <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/> Справочник по C#
4. Cайт. <https://metanit.com/sharp/tutorial>
5. Сайт. <https://itproger.com/course/csharp>