|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Пермский филиал федерального государственного автономного  образовательного учреждения высшего образования  “Национальный исследовательский университет  “Высшая школа экономики” | | | | |
|  | | | | |
| **Факультет профессиональной переподготовки** | | | | |
|  | | | | |
| **Разработка системы «Сравнение компьютеров»** | | | | |
|  |  | | |  |
|  | | | | |
| Проектная работа по направлению подготовки  «Объектно-ориентированное программирование» | | | | |
|  | | | | |
|  | | Слушателя | Калашникова А. Д. | |
|  | | Группы | ООПД-17 | |
|  | | | | |
|  | | Руководитель проектной работы | | |
|  | |  | | |
|  | | Старший преподаватель кафедры информационных технологий в бизнесе, Лебедев В. В. | | |
|  | | | | |
| Пермь, 2018 | | | | |

**АННОТАЦИЯ**

Автор — Калашников Андрей Дмитриевич

Тема работы - «Разработка информационной системы подбора компьютерных комплектующих»

Год издания — 2018

Издательство — Факультет профессиональной переподготовки НИУ ВШЭ — Пермь

Количество глав — 5

В работе рассмотрены вопросы, связанные с анализом, проектированием и созданием информационной системы подбора компьютерных комплектующих. Рассмотрены аналоги, их преимущества и недостатки.

Работа содержит 52 страницы основного текста, состоит из пяти глав и одного приложения.

**Оглавление**

[Введение 5](#_Toc514712118)

[Глава 1. Анализ предметной области 6](#_Toc514712119)

[1.1. Оценка производительности компьютера 7](#_Toc514712120)

[1.2. Обзор существующих решений 8](#_Toc514712121)

[1.2.1. Программы оценки производительности конфигурации компьютера 8](#_Toc514712122)

[1.2.2. Сервисы сравнения конфигураций компьютеров 10](#_Toc514712123)

[1.3. Формирование требований к разрабатываемой системе 12](#_Toc514712124)

[Глава 2. Проектирование программного решения 14](#_Toc514712125)

[2.1. Диаграмма прецедентов 14](#_Toc514712126)

[2.1.1. Прецедент «Вход в систему» 15](#_Toc514712127)

[2.1.2. Прецедент «Конструирование конфигурации» 16](#_Toc514712128)

[2.1.3. Прецедент «Выбор центрального процессора» 16](#_Toc514712129)

[2.1.4. Прецедент «Выбор оперативной памяти». 17](#_Toc514712130)

[2.1.5. Прецедент «Выбор платы видеоускорителя» 17](#_Toc514712131)

[2.1.6. Прецедент «Выбор системы хранения данных» 18](#_Toc514712132)

[2.1.7. Прецедент «Обновление конфигурации» 18](#_Toc514712133)

[2.1.8. Прецедент «Поиск единицы оборудования базе данных» 19](#_Toc514712134)

[2.1.9. Прецедент «Сравнение конфигураций» 19](#_Toc514712135)

[2.1.10. Прецедент «Удаление конфигурации» 20](#_Toc514712136)

[2.2. Диаграммы активностей 20](#_Toc514712137)

[2.2.1 Вход в систему 20](#_Toc514712138)

[2.2.2. Конструирование конфигурации 21](#_Toc514712139)

[2.2.3. Обновление конфигурации 22](#_Toc514712140)

[2.2.4. Поиск единицы оборудования в базе 23](#_Toc514712141)

[2.2.5 Сравнение конфигураций 24](#_Toc514712142)

[2.2.6. Удаление конфигурации 25](#_Toc514712143)

[2.3. Статическая структура системы 27](#_Toc514712144)

[2.3.1. Вход в систему 28](#_Toc514712145)

[2.3.2. Конструирование конфигурации 28](#_Toc514712146)

[2.3.3. Обновление конфигурации 28](#_Toc514712147)

[2.3.4. Поиск единицы оборудования в базе данных 28](#_Toc514712148)

[2.3.5. Сравнение конфигураций 29](#_Toc514712149)

[2.3.6. Удаление конфигурации 29](#_Toc514712150)

[2.4. Связи таблиц базы данных 29](#_Toc514712151)

[Глава3. Имя главы3 30](#_Toc514712152)

[3.1. Подзаголовок главы 3 30](#_Toc514712153)

[3.1.1. Подподзаголовок главы 3 30](#_Toc514712154)

[Приложение А. Диаграммы классов 31](#_Toc514712155)

# Введение

Современное общество невозможно представить без компьютеров. Будь то обычная сим-карта или дата-центр в несколько десятков гектар площадью, смартфоны, настольные решения, планшеты, игровые автоматы, сложные системы управления технологическими линиями — всё это представляет собой компьютер в том или ином виде. Компьютеризация несомненно затронула все сферы жизнедеятельности человечества. Вычислительные машины являются мощным инструментом, который упрощает нашу жизнь. Компьютеру не нужен отдых, а вышедшие из строя детали легко заменить. В последние годы складывается тенденция к развитию слабой форме искусственного интеллекта — когда компьютер начинает делать выводы по решаемой задаче самостоятельно, что уже применяется в области медицины и астрономии.

Среди массового потребителя очень популярно решение в виде стационарного домашнего компьютера, который позволяет выполнять игровые и мультимедийные функции. Чаще всего представляет собой совокупность нескольких компонентов: системный блок, монитор, манипуляторы ввода (клавиатура, мышь), колонки. Подобная конфигурация позволяет легко заменить любой из компонентов самостоятельно, не обладая специфичными знаниями или инструментом. Компоненты системного блока так же представляют из себя отдельные аппаратные решения, которые можно заменить. Если выбор периферийных устройств не представляет сложную задачу, то выбор компонентов системного блока — задача требующего особых знаний.

Цель данной работы: разработать информационную систему сравнения компьютеров. Основная задача, которая будет решаться: предоставить техническому специалисту средство, которое позволит сравнивать конфигурации компьютеров на основании объективной оценки.

# Глава 1. Анализ предметной области

Слово «компьютер» является англицизмом. В своей изначальной форме «computer» является производным от слова «compute», что означает «вычислять». Таким образом устройство, что позволяет производить вычисления является своеобразным компьютером. К первым устройствам подобного типа чаще всего относят счёты, которые появились примерно за три тысячи лет до нашей эры.

В конце второй мировой войны произошёл сильный скачок в развитии компьютеров в связи с необходимостью больших расчётов в рамках разработки ядерного вооружения (Об этом подробнее см. Таненбаум Э., Бос Х. «Современные операционные системы.» 4-е изд. — СПб.). В данный период появляются компьютеры первого поколения. Это были огромные вычислительные комплексы, основанные на электронных лампах. Второе поколение компьютеров относят к промежутку 1955 — 1965 г., с момента начала использования транзисторов. Третье поколение компьютеров (1965 — 1980 г.) появилось благодаря внедрению технологии интегральных схем. В данный период было создано легендарное семейство компьютеров IBM/360. Данное семейство стало популярным благодаря разделению архитектуры и реализации. Архитектура данного семейства, в свою очередь, стала промышленным стандартом и используется по сей день. Четвёртое поколение компьютеров (с 1980 г. по наши дни) связывают с появлением больших интегральных схем. В данный период компьютер начинает быть персонализированным устройством. Именно четвёртое поколение компьютеров и будет рассмотрено в рамках данной работы.

Компьютеры любого предшествующего поколения требовали большого штата специально обученных специалистов. При чём не только для возможности технического обслуживания, но и для ввода информации. На сегодняшний день собрать и настроить персональный компьютер может любой желающий. Чаще всего потребитель предпочитает переложить ответственность за качество конечного устройства на «плечи» продавца. Подобное разграничение ответственности даже привело к созданию организаций, которые занимаются подобной работой профессионально. Например, Dell.

Сегодня сердцем персональной компьютерной системы является системный блок. Эволюция электронных компонентов системного блока привела к тому, что на данный момент можно выделить следующие классы устройств:

* Центральный процессор
* Материнская плата
* Оперативная память
* Плата графического ускорителя (видеокарта)
* Звуковая карта
* Сетевая карта
* Накопитель постоянной памяти (жёсткий или твердотельный диск)
* Привод оптических дисков
* Устройство считывания флеш-карт

Почти все компоненты системного блока влияют на производительность компьютера, но можно выделить основные узлы, которые принимают на себя основную вычислительную нагрузку: центральный процессор, оперативная память, плата графического ускорителя, жёсткий диск. Если обобщить выбранные компоненты, то это память и чипы, производящие все основные вычисления.

## 1.1. Оценка производительности компьютера

Для объективной оценки производительности компьютера было создано специальное программное обеспечение — бенчмарки. Как правило бенчмарки представляют собой систему тестов производительности компьютера в повседневных задачах:

* Сжатие файлов архиватором
* Обработка мультимедиа контента
* Обработка объёмных сцен 3D-графики

Бывают и более специфичные тесты производительности:

* Тесты, проверяющие скорость записи и чтения в оперативной и постоянной памяти
* Тесты на скорость обработки математических вычислений

В ходе проведения теста программа формирует количественную оценку, которая обычно выражается в условных единицах. На усмотрение разработчика программы данная оценка может заносится в базу данных, расположенной на сервере разработчика. Впоследствии программа может предоставить возможность сравнения текущей компьютерной конфигурации с конфигурациями из данной базы данных.

Существует так же количественная оценка вычислительной мощности компьютера — количество операций с числами с плавающей точкой в секунду (флопс; FLOPS англ.). Данная оценка, как правило, важна для сложных математических операций и операций моделирования.

## 1.2. Обзор существующих решений

Сравнить конфигурации компьютеров возможно благодаря не только бенчмаркам, которые уже включает в функционал возможность сравнения текущей конфигурации с другими, но и с помощью открытых баз данных, составленных энтузиастами, или сервисом сравнения конфигураций, предоставляемым интернет-магазином.

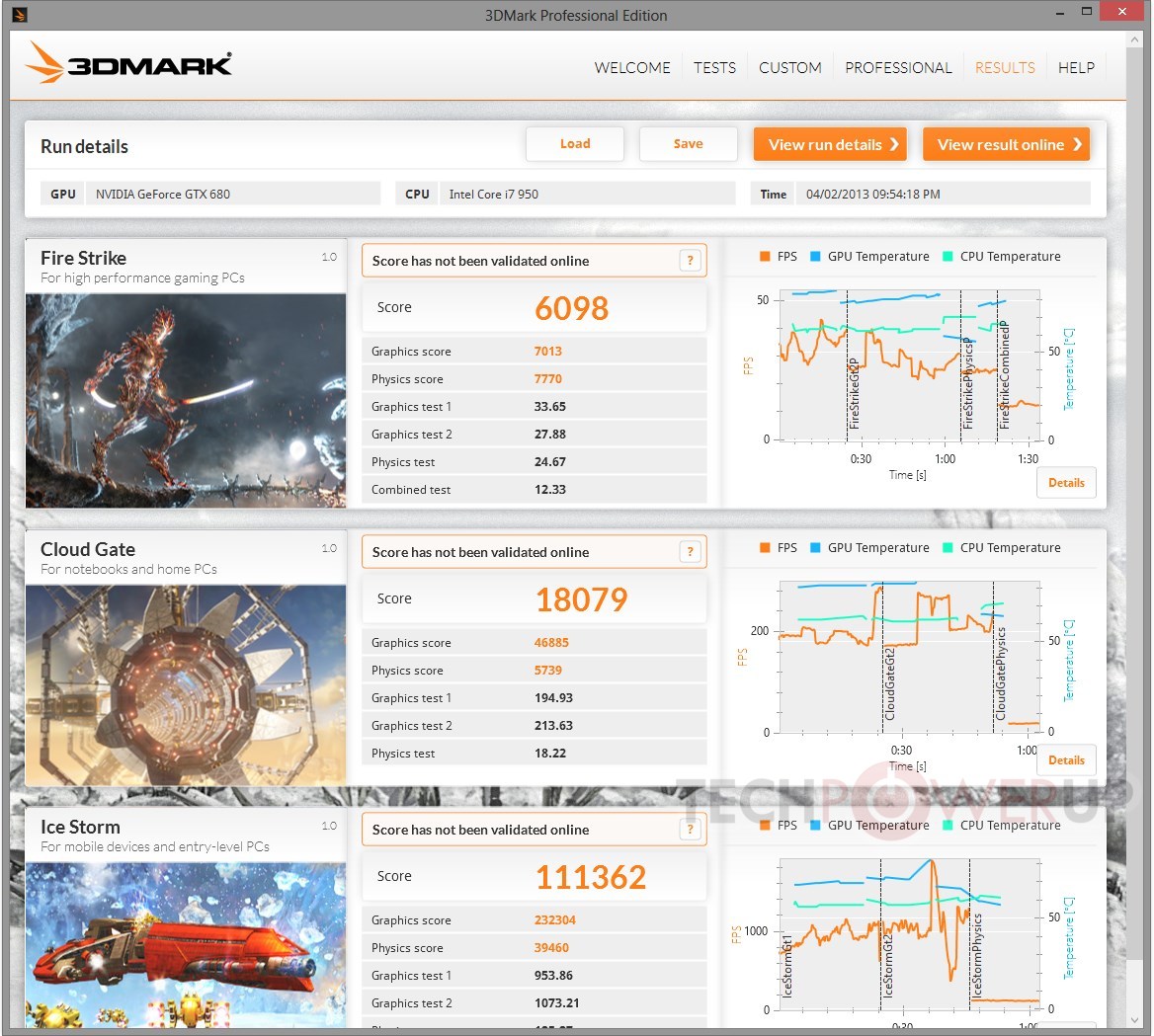
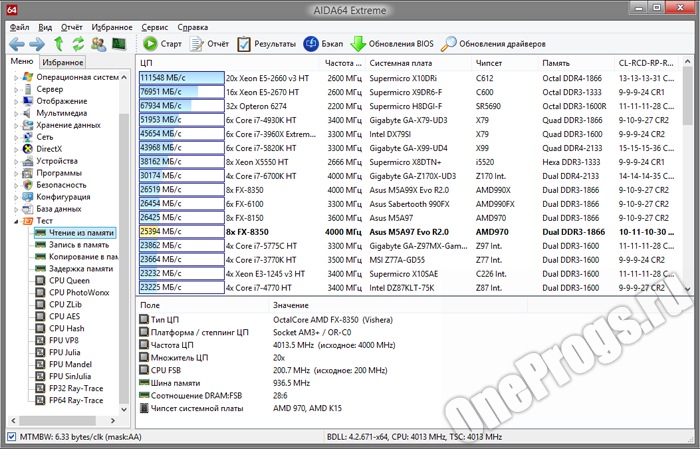
### 1.2.1. Программы оценки производительности конфигурации компьютера

Вот небольшой список бенчмарков, которые производят оценку производительности компьютера:

* 3DMark (см. рис. 1.1)
* PCMark
* AIDA64 (см. рис. 1.2)
* Fraps
* Jbenchmark

Программы для комплексного теста производительности компьютера, как правило, обеспечивают всесторонний тест, учитывающий все основные вычислительные узлы компьютера. По мере прохождения теста производительности на том или ином узле — программа выставляет оценку. После завершения теста пользователю предлагается оценить результат в виде итоговой оценки по всем проведённым тестам. По данной оценке можно сравнить производительность текущего компьютера с производительностью компьютеров других людей. Например, в сети интернет на сайтах компаний, производящий данный тест, или специализированных форумах. Если тест выдаёт количественную оценку производительности компьютера, выраженную в числах, то в разговорной речи употребляют термин «попугаи», поскольку не существует единицы измерения данной величины.

Подобная оценка является объективной, поскольку выполняется по заранее заложенному алгоритму и на результат влияет только быстродействие компонентов компьютера.



**Рисунок 1.2. Окно программы «AIDA64» на вкладке тестирования оперативной памяти**

**Рисунок 1.1. Основное окно программы "3DMark"**

С другой стороны, при использовании подобных программ возникает очень много факторов, которые влияют на конечную оценку:

* Изношенность компонентов системы
* Проблемы в системе охлаждения
* Используемая операционная система
* Запущенные программы на компьютере в момент теста

Как правило оценку в данных программах делают несколько раз. Даже один «проход» занимает продолжительное время, а совокупность подобных тестов увеличивает ожидание многократно.

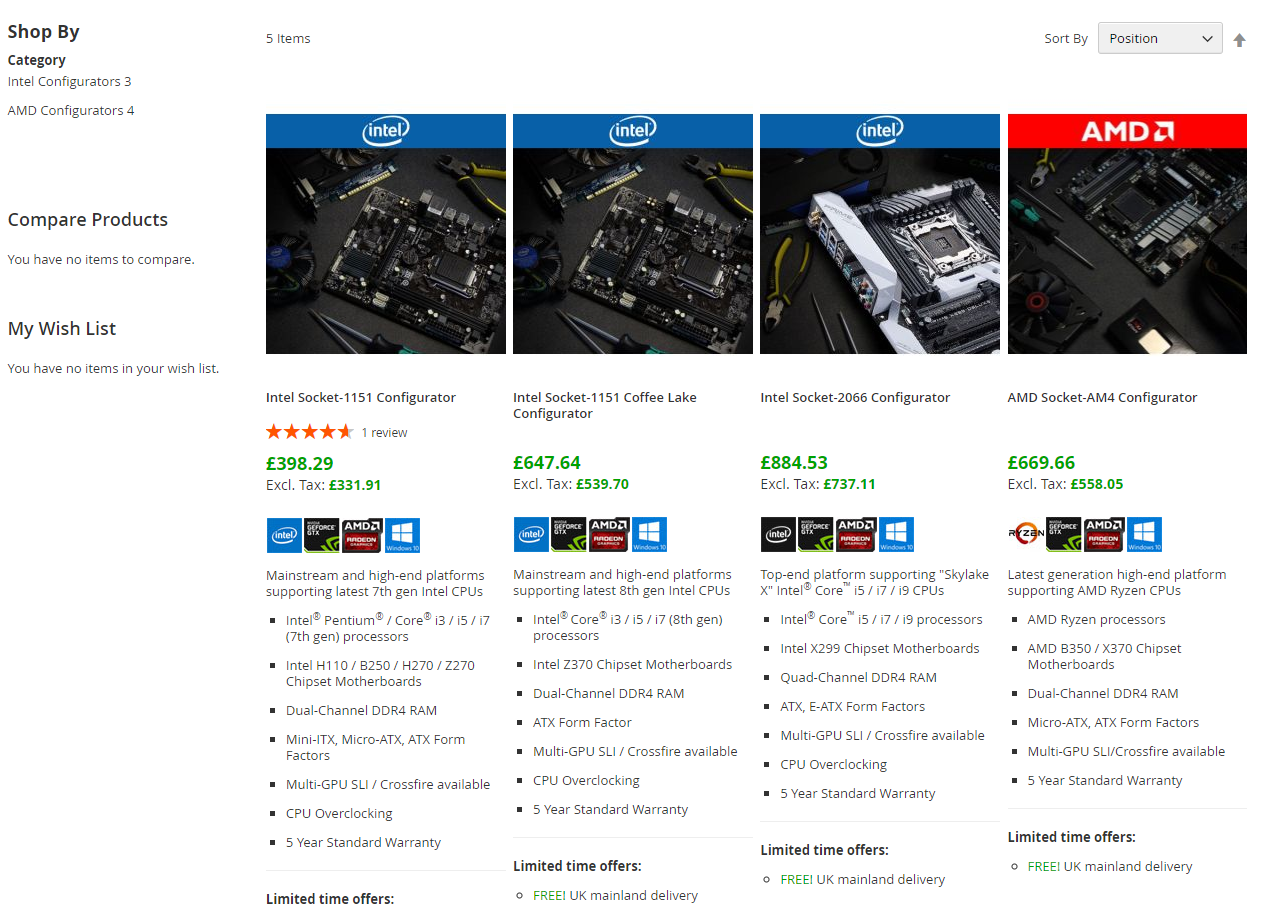
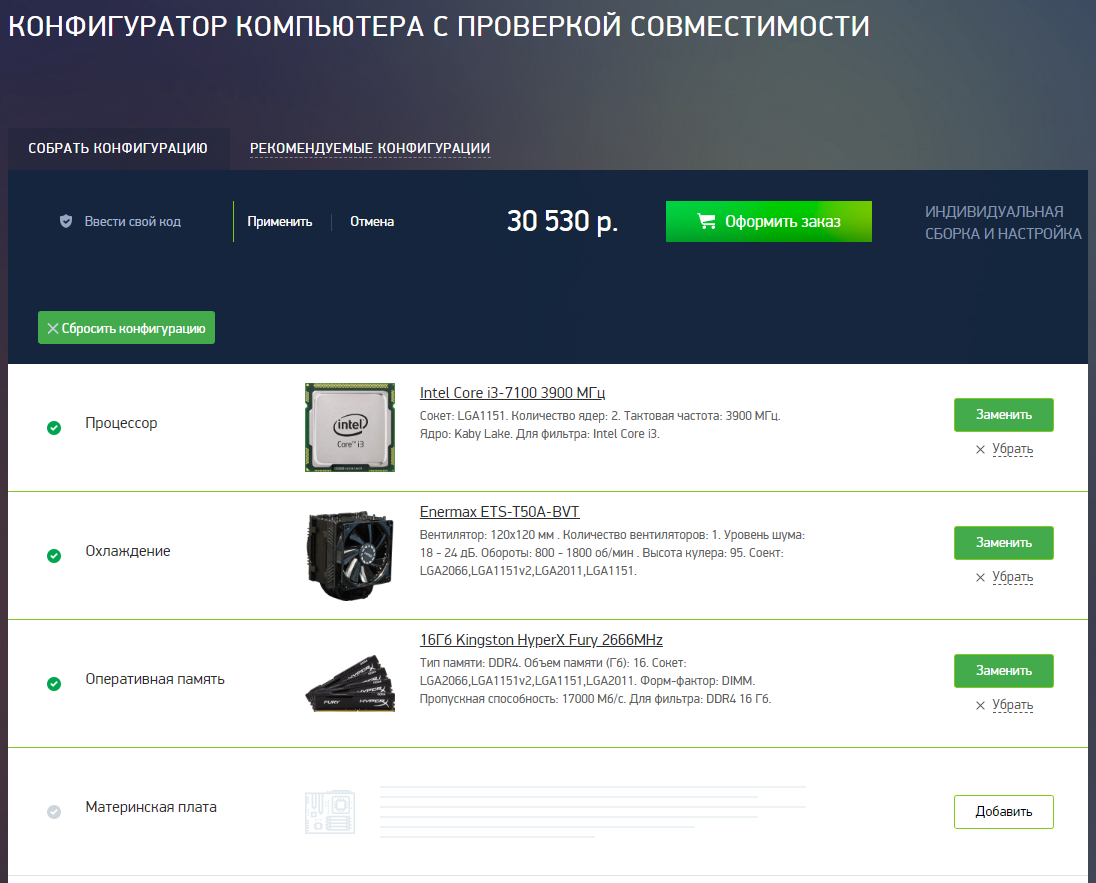
Использование одной и той же программы накладывает зависимость оценки от алгоритмов тестирования. Участники, которые хотят получить объективное сравнение, должны пользоваться одной и той же программой. Данное обстоятельство делает людей зависимыми от базы сравнения производительности компьютеров производителя данной программы. Если производитель данной программы по каким-либо причинам перестаёт поддерживать базу и данные в ней теряют актуальность — подобная оценка становится не показательной.

Качественное программное обеспечение (далее ПО) для тестирования компьютеров является платным и цены сравнительно высокие. Например, за одну лицензию профессионального издания программы PCMark было необходимо (на момент написания работы) заплатить порядка полутора тысячи долларов.

### 1.2.2. Сервисы сравнения конфигураций компьютеров

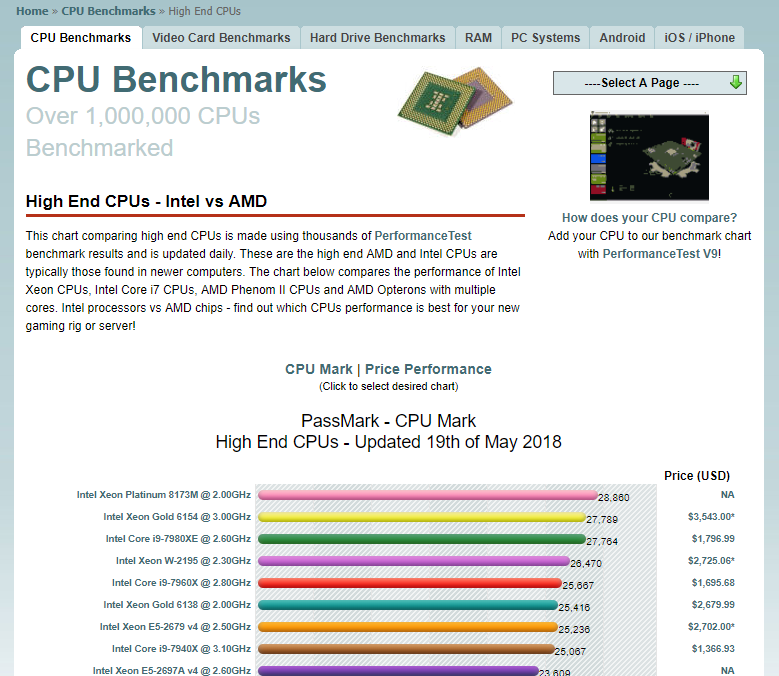
Существует множество сервисов, которые позволяют конструировать электронную начинку будущего компьютера, а также сравнивать по объективной оценке различные конфигурации компьютера. Сервисы представлены как отечественными разработчиками, так и зарубежными:

* https://www.dinopc.com/online-pc-configurator/ (см. рис. 1.3)
* https://edelws.ru/constructor/ (см. рис. 1.4)
* http://www.ironbook.ru/constructor/
* https://www.cpubenchmark.net/ (см. рис. 1.5)



**Рисунок 1.4. Страница конструирования конфигурации компьютера сайта «edelws.ru»**

**Рисунок 1.3. Страница выбора конфигурации компьютера сайта « dinopc.com»**

Подобные сервисы, как правило, предоставляют магазины, чтобы покупатель после подбора интересующего оборудования мог сразу его купить, что, несомненно, является плюсом. Основным недостатком подобных сервисов является отсутствие возможности оценить производительность получаемой конфигурации в сравнении с другими конфигурациями. Также при отсутствии интернет-соединения сервис будет недоступен.

**Рисунок 1.5. Страница таблицы результатов бенчмарков центральных процессоров на сайте «cpubenchmark.net»**

## 1.3. Формирование требований к разрабатываемой системе

На основании всего вышеизложенного функциональные требования к разрабатываемой системе будут следующими:

* Разрабатываемая система должна предоставлять возможность сравнивать получаемые конфигурации системных блоков на основании объективной оценки производительности.
* Оценка производительности узлов компьютера должна производится на основании числового значения, полученного с помощью специализированного ПО, либо из пополняемой актуальной общедоступной базы данных подобных оценок.
* Разрабатываемая система должна иметь связь с существующей базой доступного оборудования на складе предприятия.
* Доступ к системе должен осуществляться вне зависимости от программной платформы (операционной системы)

# Глава 2. Проектирование программного решения

Этап проектирования программного решения позволит сформулировать и структурировать основные решаемые задачи в ходе разработки, выделить основные группы пользователей системы, описать поведение системы в разных состояниях.

В качестве инструмента для проектирования будет использоваться унифицированный язык моделирования (англ. universal modeling language, UML). Язык представляет собой набор правил, которые позволяют создать диаграммы, отражающие суть решаемых задач и проблем, взаимодействие компонентов системы и т.д.

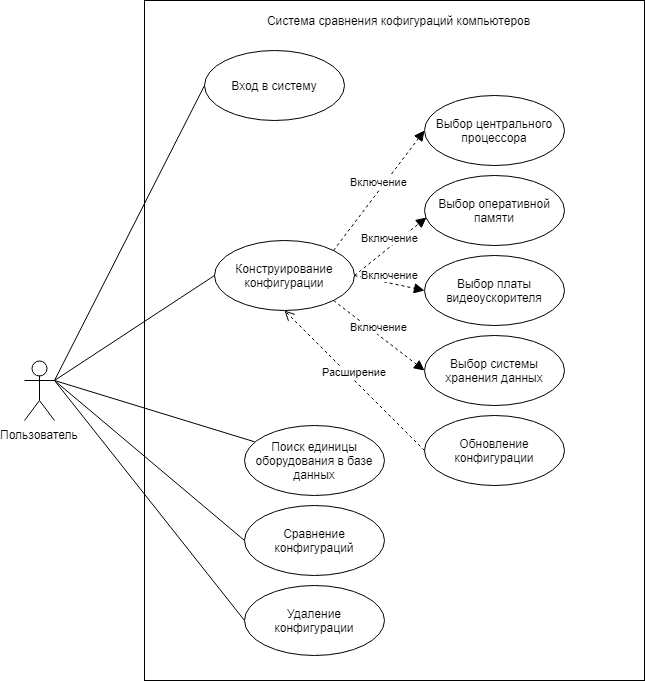
## 2.1. Диаграмма прецедентов

Диаграмма прецедентов показывает основных пользователей будущего программного обеспечения и основные варианты использования. Для разрабатываемой системы диаграмма прецедентов приведена ниже (см. рис. 2.1).

Пользователем системы является технический специалист, который может войти в данную систему и производить различные манипуляции с конфигурациями:

* Создавать
* Сохранять
* Сравнивать одни конфигурации с другими
* Удалять конфигурации
* Обновлять сохранённые ранее конфигурации

Конструирование конфигурации, в свою очередь, представляет из себя последовательность операций выбора компонентов.



**Рисунок 2.1. Диаграмма прецедентов**

### 2.1.1. Прецедент «Вход в систему»

Название: вход в систему.

Актор: пользователь.

Краткое описание: пользователь авторизуется в системе.

Триггер: открытие главного окна программы.

Таблица 3.1. Вход в систему

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Заходит в систему | Формирует страницу авторизации |
| Вводит имя пользователя и пароль (E1) | Предоставление доступа к основной странице программы |

Альтернативные потоки:

Е1: В случае некорректного ввода имени пользователя или пароля система перенаправляет пользователя на повторный ввод данных.

### 2.1.2. Прецедент «Конструирование конфигурации»

Название: конструирование конфигурации.

Актор: пользователь.

Краткое описание: создание, изменение, просмотр и удаление конфигурации компьютера.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы

Таблица 3.2. Конструирование конфигурации

|  |  |
| --- | --- |
| Действие актора | Действие актора |
| Актор последовательно производит выбор единиц оборудования для составления конфигурации. | Система фиксирует выбор пользователя. |
| Актор посылает сигнал на сохранение конфигурации. | Сохраняет созданную конфигурацию (Е1, E2). |

Подпотоки: отсутствуют.

Альтернативные потоки:

E1: Если пользователь не выбрал единицу оборудования из представленных позиций — возникает предупреждение о необходимости заполнить соответствующее поле.

E2: Если количество возможных сохранённых конфигураций превышает заданное — система выдаёт предупреждение о невозможности сохранить текущую собранную конфигурацию.

### 2.1.3. Прецедент «Выбор центрального процессора»

Название: выбор центрального процессора.

Актор: пользователь.

Краткое описание: пользователю предоставляется возможность выбора модели центрального процессора.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.3. Выбор центрального процессора

|  |  |
| --- | --- |
| Действие актора | Действие актора |
| Актор посылает сигнал на предоставление возможности выбора модели центрального процессора. | Актор посылает сигнал на предоставление возможности выбора модели центрального процессора. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.4. Прецедент «Выбор оперативной памяти».

Название: выбор оперативной памяти.

Актор: пользователь.

Краткое описание: пользователю предоставляется возможность выбора модели оперативной памяти.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.4. Выбор оперативной памяти

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает сигнал на предоставление возможности выбора модели оперативной памяти. | Система делает запрос к базе данных и предоставляет пользователю выборку моделей оперативной памяти. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.5. Прецедент «Выбор платы видеоускорителя»

Название: выбор платы видеоускорителя.

Актор: пользователь.

Краткое описание: пользователю предоставляется возможность выбора модели платы видеоускорителя.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.5. Выбор оперативной памяти

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает сигнал на предоставление возможности выбора модели платы видеоускорителя. | Система делает запрос к базе данных и предоставляет пользователю выборку моделей платы видеоускорителя. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.6. Прецедент «Выбор системы хранения данных»

Название: выбор системы хранения данных.

Актор: пользователь.

Краткое описание: пользователю предоставляется возможность выбора системы хранения данных.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.6. Выбор оперативной памяти

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает сигнал на предоставление возможности выбора системы хранения данных. | Система делает запрос к базе данных и предоставляет пользователю выборку систем хранения данных. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.7. Прецедент «Обновление конфигурации»

Название: обновление конфигурации.

Актор: пользователь.

Краткое описание: пользователю предоставляется возможность изменить выбранную конфигурацию.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.7. Обновление конфигурации

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает сигнал на предоставление возможности обновления конфигурации. | Система формирует ответ в виде формы конструирования конфигурации с подставленными значениями из выбранной конфигурации. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.8. Прецедент «Поиск единицы оборудования базе данных»

Название: поиск единицы оборудования базе данных.

Актор: пользователь.

Краткое описание: поиск единицы оборудования в текущей базе данных с заданными критериями.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.8. Поиск единицы оборудования

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор вводит критерии поиска. | Система формирует запрос к базе данных с заданными критериями. |
| Актор посылает сигнал о начале поиска. | Система обрабатывает сигнал и предоставляет пользователю соответствующую выборку. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.9. Прецедент «Сравнение конфигураций»

Название: сравнение конфигураций.

Актор: пользователь.

Краткое описание: система предоставляет возможность просмотра всех составленных конфигураций для анализа пользователем.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.9. Сравнение конфигураций

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает запрос на вывод информации о текущих составленных конфигурациях. | Система формирует подробную выборку о составленных конфигурациях и предоставляет сформированный ответ для пользователя. |

Подпотоки: отсутствуют.

Альтернативные потоки: отсутствуют.

### 2.1.10. Прецедент «Удаление конфигурации»

Название: удаление конфигурации.

Актор: пользователь.

Краткое описание: система предоставляет возможность удалить выбранную конфигурацию.

Триггер: пользователь выполнил вход в систему и перешёл на соответствующую вкладку программы.

Таблица 3.10. Сравнение конфигураций

|  |  |
| --- | --- |
| Действие актора | Отклик системы |
| Актор посылает запрос на удаление выбранной конфигурации. | Система фиксирует выбранную конфигурацию и формирует запрос на подтверждение проведения операции. |
| Актор посылает сигнал о подтверждении(Е1). | Система удаляет выбранную конфигурацию из общего списка. |

Подпотоки: отсутствуют.

Альтернативные потоки:

Е1: Если актор изменил своё решение и посылает сигнал об отмене операции удаления — система перенаправляет актора на страницу сравнения конфигураций.

## 2.2. Диаграммы активностей

Диаграммы активностей демонстрируют процесс взаимодействия пользователя и системы, а наглядное представление упрощает восприятие информации и систематизацию процессов, происходящих в системе.

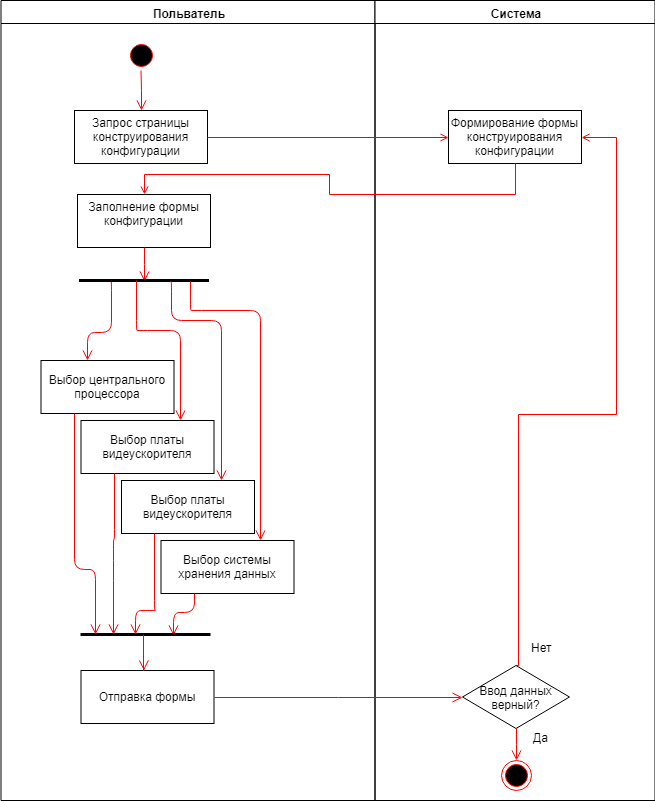
### 2.2.1 Вход в систему

Диаграмма активностей «Вход в систему» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.2)

### 2.2.2. Конструирование конфигурации

**Рисунок 2.2 Диаграмма «Вход в систему»**

Диаграмма активностей «Конструирование конфигурации» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.3).



**Рисунок 2.3 Диаграмма «Конструирование конфигурации»**

### 2.2.3. Обновление конфигурации

Диаграмма активностей «Обновление конфигурации» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.4).

### 2.2.4. Поиск единицы оборудования в базе

**Рисунок 2.4. Диаграмма «Обновление конфигурации»**

Диаграмма активностей «Поиск единицы оборудования в базе» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.5).

### 2.2.5 Сравнение конфигураций

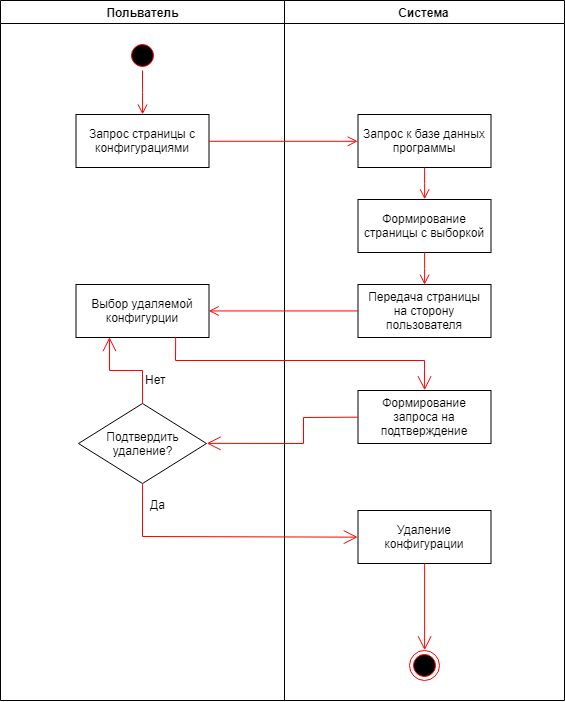
**Рисунок 2.5. Диаграмма «Поиск единицы оборудования в базе»**

Диаграмма активностей «Сравнение конфигураций» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.6).

### 2.2.6. Удаление конфигурации

**Рисунок 2.6. Диаграмма «Сравнение конфигураций»**

Диаграмма активностей «Удаление конфигурации» отображает порядок действий пользователя системы и системы. Диаграмма представлена на рисунке ниже (см. рис. 2.7).



**Рисунок 2.7. Диаграмма “Удаление конфигурации”**

## 2.3. Статическая структура системы

Разрабатываемая система будет представлять из себя веб-приложение. Приложения подобного типа содержат в себе две части:

* Фронтенд (frontend англ. - передняя часть) — представляющий из себя веб-страницы.
* Бэкенд (backend англ. - задняя часть) — представляющий из себя совокупность программ, отрабатывающих всю основную логику приложения на стороне сервера.

Фроненд может содержать в себе сложную логику, но она, как правило, относится к тому — каким образом информация будет отображаться в браузере пользователя. Фронтенд определяется стеком трёх технологий: HTML – гипертекстовый язык разметки страниц; CSS – набор каскадных стилей для определения стиля объектов на странице; JavaScript – интерпретируемый язык программирования, который предоставляет возможность написания сложной логики взаимодействия элементов на странице. В рамках данного проекта фронтенд определяет пользовательский интерфейс, который будет описан в соответствующем разделе.

Бэкенд составляет «скелет» проекта, который выполняется на стороне сервера и осуществляет всю основную работу по обработке данных: взаимосвязь с базой данных, взаимосвязь со сторонними модулями и программами, подготовка данных для передачи их в контекст страницы и т. д. Для того, чтобы разделить ответственность выполнения тех или иных операций широко применяется паттерн проектирования приложений «MVC».

Model-View-Controller (MVC, «Модель-Представление-Контроллер», «Модель-Вид-Контроллер») — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

В рамках данного проекта модель будет представлять логику, которая отвечает за взаимосвязь с базой данных. Описывая объект модели, фактически, мы описываем и сущность таблицы, в которую будут заносится данные для приложения.

Объекты представления — это страницы с гипертекстовой разметкой со специальными управляющими символами, обрабатывая которые, сервер подставляет требуемую информацию.

Объекты контроллера — сущности, которые соединяют логику работы с объектами модели и представления. Данные объекты отвечают за обращение к объектам модели, чтобы те в свою очередь предоставили информацию из базы данных. Обрабатывают данную информацию в соответствии с требуемой логикой работы приложения и передают результат в контекст объектов представления.

В соответствии с принятым паттерном и диаграммой прецедентов составим схемы взаимодействия объектов в нотации диаграмм классов UML.

### 2.3.1. Вход в систему

Связь сущностей представляет собой реализацию ограничение прав доступа пользователя на страницу. Основные классы контроллеров наследуют свойства базового класса аутетификации, который реализует в себе основную логику. Диаграмма классов представлена на рисунке в приложении (см. прил. А, рис.А.1).

### 2.3.2. Конструирование конфигурации

Связь сущностей представляет собой реализацию возможности добавления конфигурации компьютера к сравнению. При этом главный объект модели связан с другими моделями и хранит в себе лишь ссылку на объекты других моделей. Диаграмма классов представлена на рисунке в приложении (см. прил. А, рис.А.2).

### 2.3.3. Обновление конфигурации

Связь сущностей представляет собой реализацию возможности обновления конфигурации компьютера. При этом главный объект модели связан с другими моделями и хранит в себе лишь ссылку на объекты других моделей. Диаграмма классов представлена на рисунке в приложении (см. прил. А, рис.А.3).

### 2.3.4. Поиск единицы оборудования в базе данных

В проекте используются четыре типа аппаратного обеспечения компьютера:

* Центральный процессор
* Оперативная память
* Жёсткий диск
* Видеоускоритель

Рационально разделить поиск единицы оборудования по каждой из категории и для каждой категории сформировать свою отдельную таблицу в базе данных, а следовательно, предусмотреть отдельную модель. Ранее было показано, что относительно каждой из моделей будет существовать связь с моделью сравнения конфигураций. В связи с этим и принятым соглашением об использовании паттерна MVC ниже будут представлены следующие диаграммы связей:

* Диаграмма классов «Поиск в БД проекта моделей центральных процессоров» (см. прил. А, рис.А.4).
* Диаграмма классов «Поиск в БД проекта моделей оперативной памяти» (см. прил. А, рис.А.5).
* Диаграмма классов «Поиск в БД проекта моделей жёстких дисков» (см. прил. А, рис.А.6).
* Диаграмма классов «Поиск в БД проекта моделей видеоускорителей» (см. прил. А, рис.А.7).

### 2.3.5. Сравнение конфигураций

Связь сущностей представляет собой реализацию возможности сравнения ранее добавленных конфигураций компьютера. При этом главный объект модели связан с другими моделями и хранит в себе лишь ссылку на объекты других моделей. Диаграмма классов представлена на рисунке в приложении (см. прил. А, рис.А.8).

### 2.3.6. Удаление конфигурации

Связь сущностей представляет собой реализацию возможности удаления ранее добавленных конфигураций компьютера. При этом главный объект модели связан с другими моделями и хранит в себе лишь ссылку на объекты других моделей. Диаграмма классов представлена на рисунке в приложении (см. прил. А, рис.А.9).

## 2.4. Связи таблиц базы данных

ыыавыа

# Глава3. Имя главы3

Текст текст текст

Текст текст текст

Текст текст текст

## 3.1. Подзаголовок главы 3

Текст текст текст

Текст текст текст

Текст текст текст

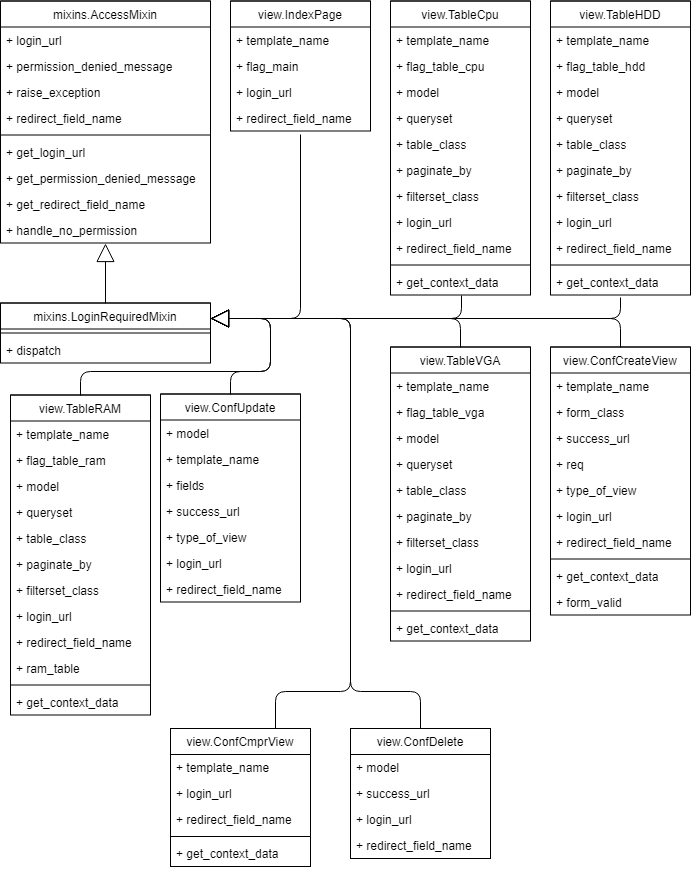
### 3.1.1. Подподзаголовок главы 3

Текст текст текст

Текст текст текст

Текст текст текст

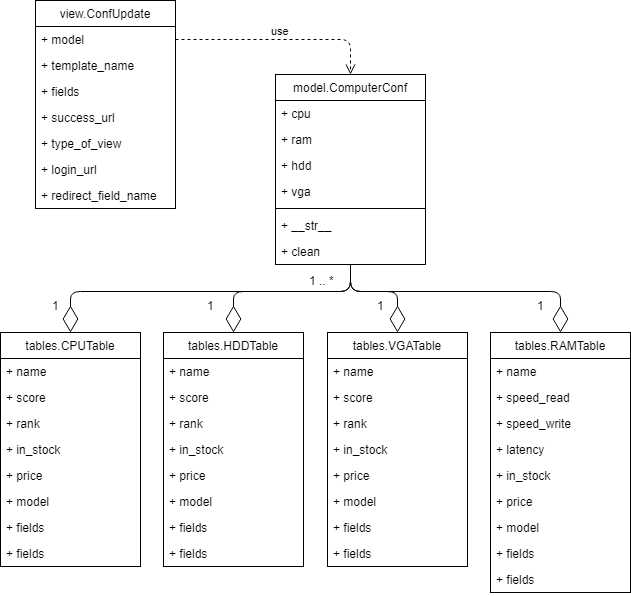
# Приложение А. Диаграммы классов



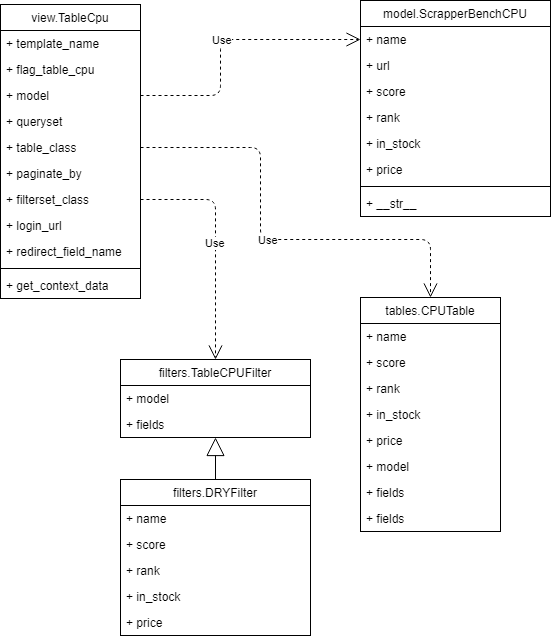
**Рисунок А.1. Диаграмма классов «Вход в систему»**



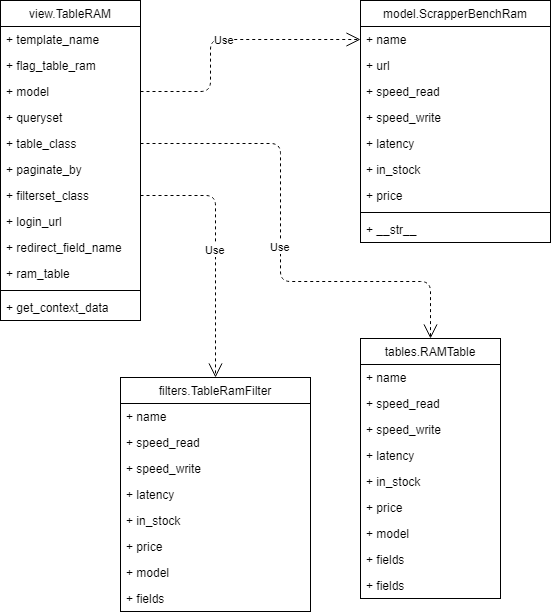
**Рисунок А.2. Диаграмма классов «Конструирование конфигурации»**



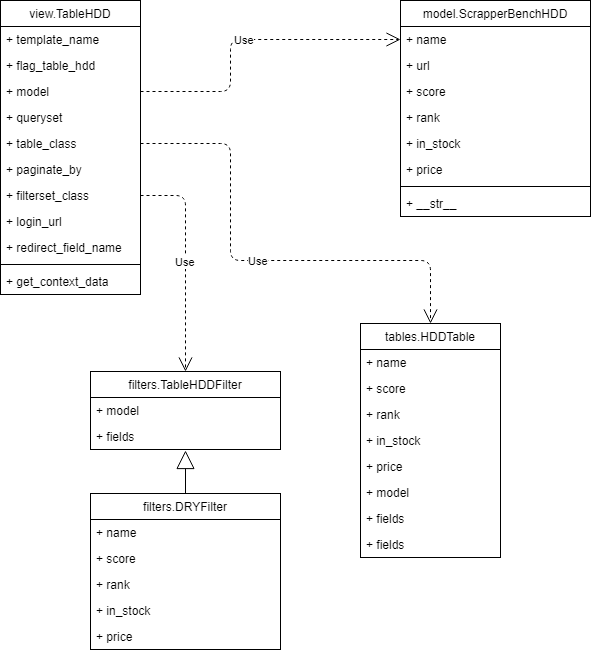
**Рисунок А.3. Диаграмма классов «Обновление конфигурации»**



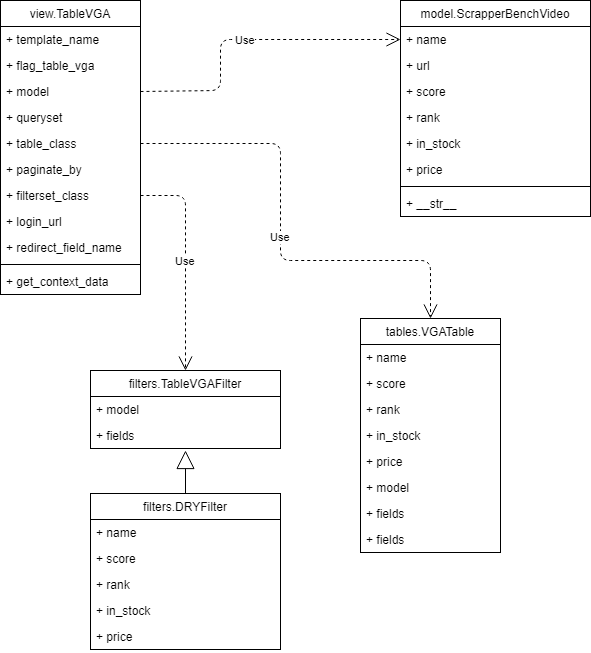
**Рисунок А.4. Диаграмма классов «Поиск в БД проекта моделей центральных процессоров»**



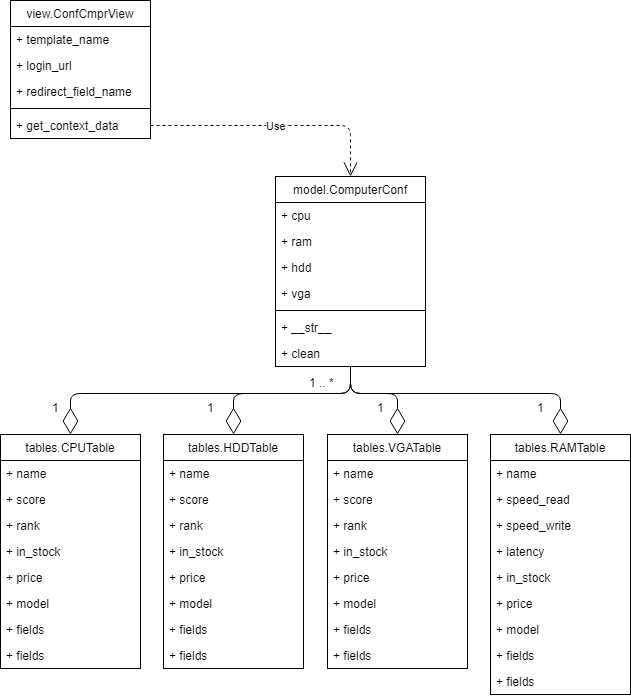
**Рисунок А.5. Диаграмма классов «Поиск в БД проекта моделей оперативной памяти»**



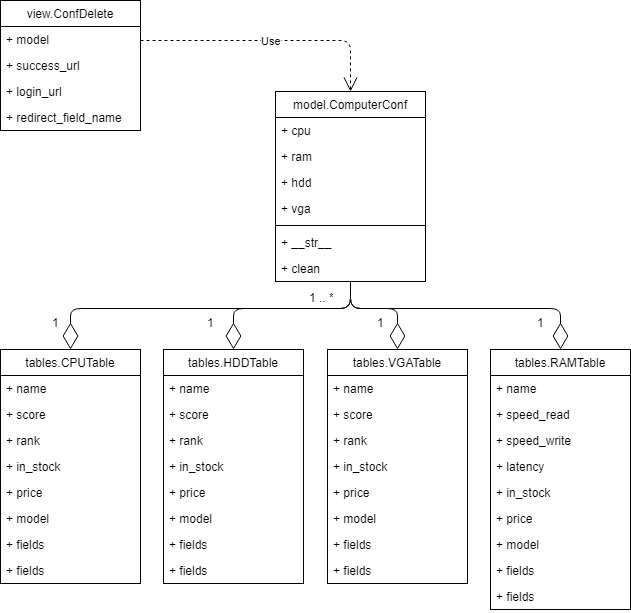
**Рисунок А.6. Диаграмма классов «Поиск в БД проекта моделей жёстких дисков»**



**Рисунок А.7. Диаграмма классов «Поиск в БД проекта моделей видеоускорителей»**



**Рисунок А.8. Диаграмма классов «Сравнение конфигураций»**



**Рисунок А.9. Диаграмма классов «Удаление конфигурации»**