

Министерство науки и высшего образования Российской Федерации

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

ОТЧЕТ

по дисциплине «Интеллектуальные системы»

на тему «Экспертная система, рекомендующая конфигурацию персонального компьютера»

Выполнили студенты команды №1

Квашнина А., Мирошник С., Киреев Д.,
Ковалева Н., Маслова Е., Урбановский Е.

1. Идентификация проблемной области

- 1.1. Экспертная система помогает обычному пользователю ПК принять решение о покупке новой или замене текущей комплектации персонального компьютера. Система получает входные параметры, включающие в себя список задач, решение которых требуется пользователю. После, на основе заданных параметров, выдаёт подходящую под задачи пользователя комплектацию ПК.

Сфера применения: ЭС может применяться как лично пользователем для самостоятельного подбора комплектации, так и системными администраторами для ускорения поиска комплектующих и упрощения процесса подбора конфигурации персонального компьютера. Также система может использоваться в магазинах цифровой техники для ускорения консультации покупателей по текущему ассортименту, а следовательно, сокращения затрат на персонал.

Эксперты:

- Специалист по сборке и ремонту ПК
- Менеджер по работе с клиентами в магазине цифровой техники
- Сервисный инженер

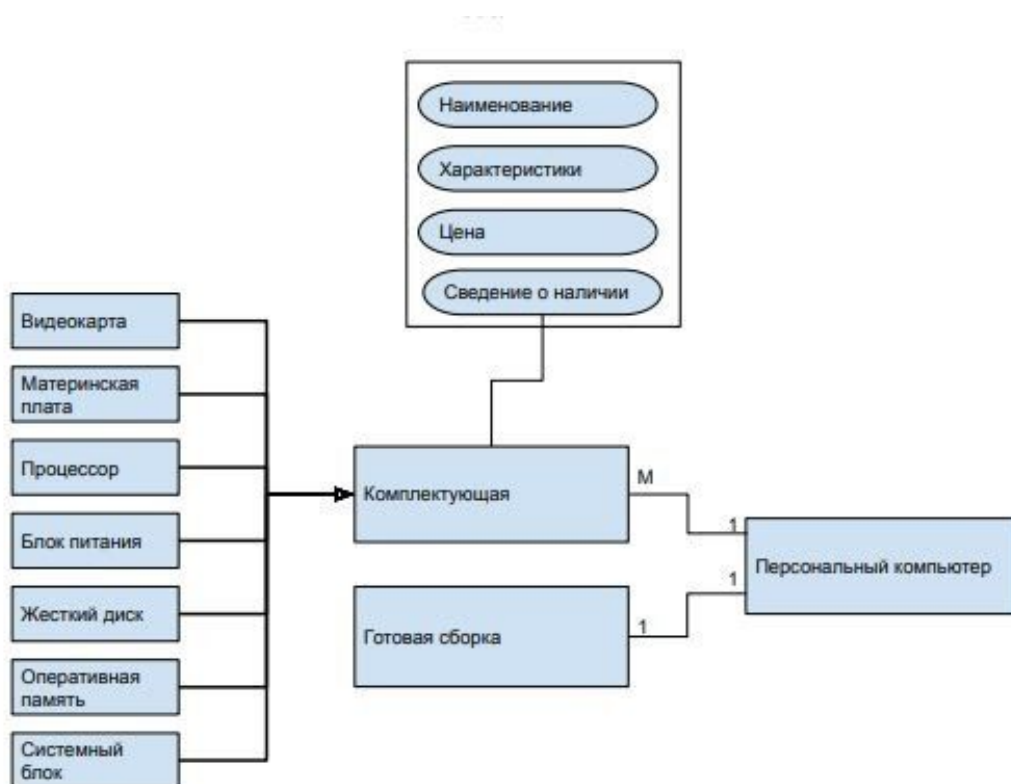
Требования к системе: система должна в кратчайшее время составить конфигурацию из существующих/имеющихся в наличии комплектующих, отвечающих требованиям пользователя и по возможности минимизирующая затраты.

- 1.2. Цель работы: разработка экспертной системы, нацеленной на предоставление информации о комплектации персонального компьютера и помощи пользователю в выборе и коррекции конфигурации персонального компьютера.

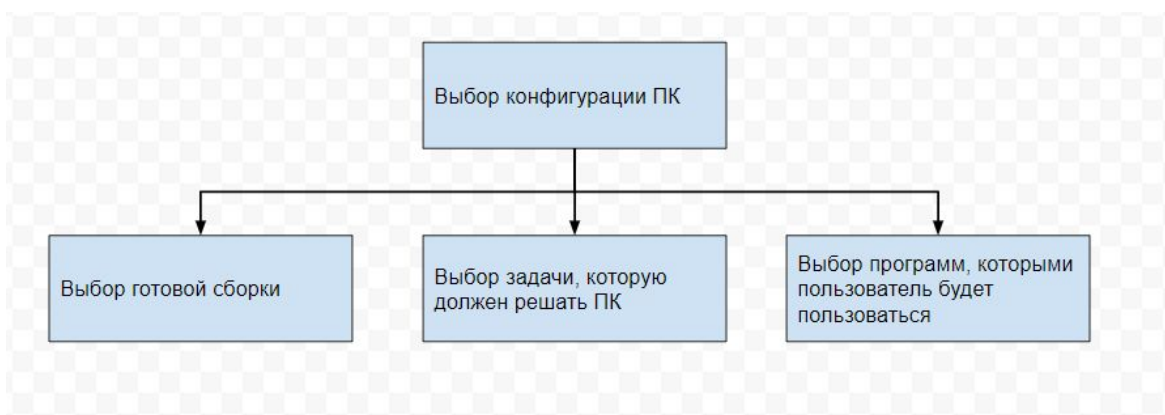
Постановка задачи: Исходя из цели работы, необходимо проработать теоретический материал по экспертным системам, изучить предметную область, выбрать язык программирования и платформу для реализации, разработать экспертную систему, подбирающую конфигурацию компьютера, и протестировать ее работу.

2. Концептуализация проблемной области

2.1. Построение объектной области



2.2. Построение функциональной модели



2.3. Построение поведенческой модели

Вывод конфигурации

Событие	Поведение	Состояние
Оформляется	Выбор решаемых задач Выбор характеристик	Оформлен
Фильтрация	Проверка соответствий требованиям Проверка наличия	Отфильтрован
Вывод	Вывод найденных комплектующих Вывод найденные готовые сборки	Выведено

3. Формализация базы знаний

3.1. Построение логической модели

\exists комплектующая
указана(характеристика_комплектующей(комплектующая, характеристика))

\exists сборка \forall характеристика \forall комплектующая (входит(комплектующая, сборка) \wedge характеристика_комплектующей(комплектующая, характеристика) \rightarrow удовлетворяет (сборка, характеристика))

\exists задача \exists программа \exists сборка соответствует(сборка, задача) \vee подходит(сборка, программа)

3.2. Построение продукционной модели

П1: ЕСЛИ “характеристика - указана”

ТО “Комплектующая с характеристиками не ниже требуемых”

П2: ЕСЛИ “необходимая комплектующая - указана”

ТО “Запрос характеристик”

П3: ЕСЛИ “Программа - указана” ИЛИ “Задача - указана”

ТО “требуемые характеристики”

П4: ЕСЛИ “Сборка”

ТО “проверка совместимости”

4. Этап реализации ЭС

4.1. Описание метода решения задачи

Для решения поставленной задачи необходимо выполнить следующие действия:

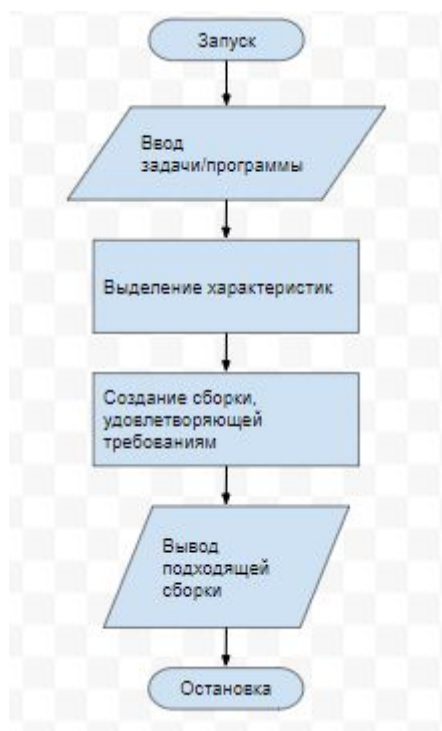
1. Ввести необходимые задачи или программы
2. Выполнить поиск подходящих конфигураций
3. Вывести подходящие конфигурации

Для успешной реализации этих действий программа должна состоять из следующих функциональных модулей:

1. Хранилище данных
2. Модуль поиска компонент по характеристикам

4.2. Структурная схема алгоритма

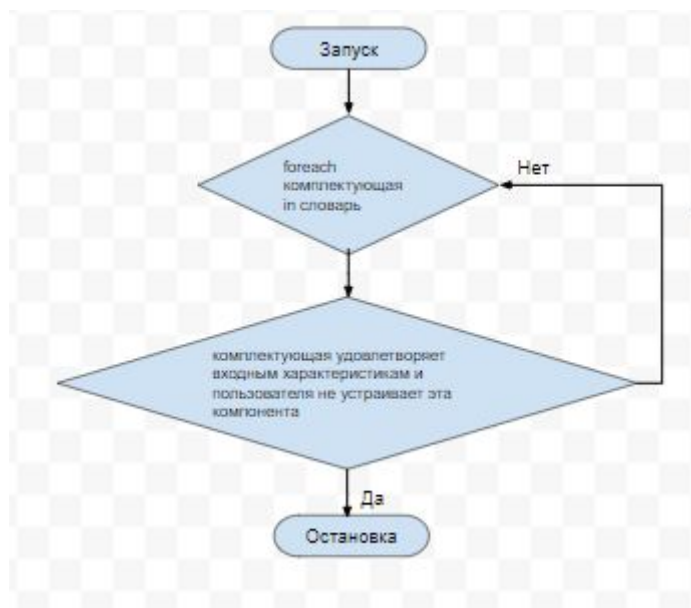
Блок-схема работы метода составления сборки



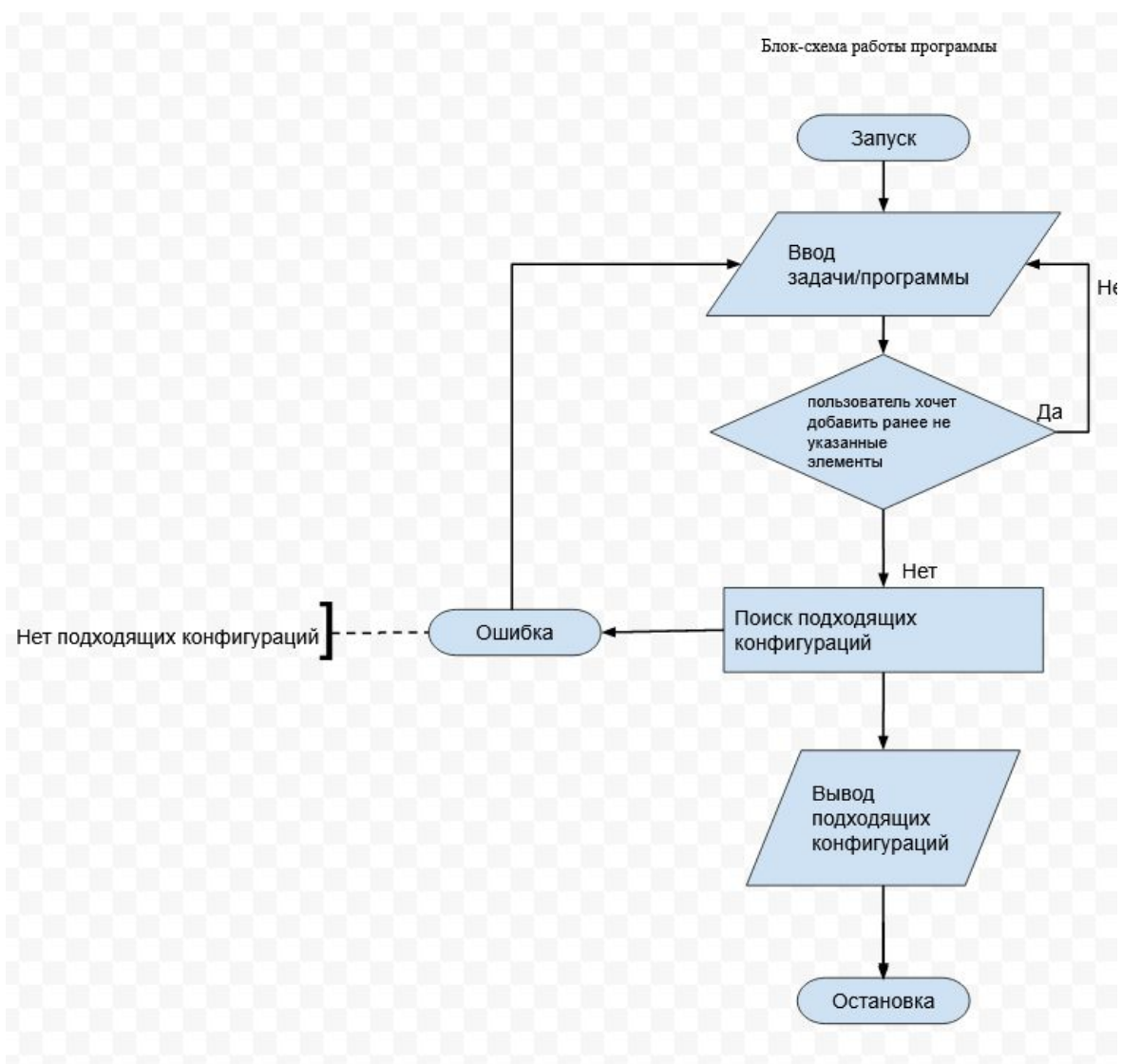
Блок-схема работы метода составления сборки, удовлетворяющей требованиям



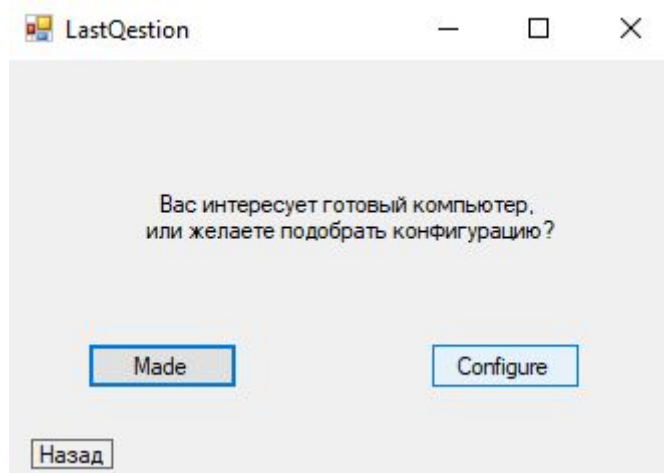
Блок-схема работы метода поиска комплектующей



4.3. Блок-схема работы программы



4.4. Скриншоты работы программы



MadeForm

Money

Цели использования

Сменить

для рядового пользователя
видеоредактор
программист
датасайентист
фоторедактор/художник

Подобрать

Назад

MadeForm

Money

Цели использования

Сменить

Visual Studio 2019
Photoshop
Sony vegas pro
Notepad++

Подобрать

Назад

4.5. Листинг программы

см. Приложение

5. Этап тестирования

5.1. Оценка точности и полезности работы

Для тестирования входным параметром была выбрана цель использования - “для рядового пользователя” и вариант минимально возможной сборки.

The screenshot shows a window titled "MadeForm" with a light gray background. On the left, a text box lists hardware specifications: "Видеокарта: MSI GeForce GT 710 Silent LP 1 19 2850", "Жесткий диск: Seagate HDD BLUE 1000 3199 6,8", "Матринская плата: Biostar Micro-ATX H110MDE LGA 1151 Intel H110 DDR4 2 3699", "Блок питания: HIPER HP-250SFX 250 1399", "Процессор: Intel Celeron G4930 LGA 1151-v2 3200 2 54 3050", and "Оперативная память: Patriot Signature 2 DDR2 1,8 1350". Below this list is a "Назад" button. On the right, there is a "Money" input field with the value "16000". Below it is a section titled "Цели использования" with a "Сменить" button. A list of options is shown: "для рядового пользователя" (highlighted in blue), "видеоредактор", "программист", "датасайтист", and "фоторедактор/художник". At the bottom right is a "Подобрать" button.

Результаты теста соответствуют ожиданиям, и экспертная система работает верно.

5.2. Вывод

В результате работы предложена концепция экспертной системы по рекомендации конфигурации персонального компьютера. Разработан алгоритм, позволяющий подобрать оптимальные подходящие конфигурации на основе данных, введенных пользователем.

Предлагаемая концепция доказательна, так как основана на системном подходе, а также имеет большой потенциал для практического применения, так как может быть использована обычным пользователем ПК.

Список литературы

1. Решитов А.А., Абдулаева У.Б. Сравнительная характеристика конфигураторов персональных компьютеров в функциональном аспекте // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. - 2018. - №3(21).
2. Джарратано Дж., Райли Г. Экспертные системы: принципы разработки и программирование. 4-е издание. — М.: Вильямс, 2007. — 1152 с
3. Поляков В. И. Интеллектуальные системы. - Томский государственный университет, 2010.
4. Евменов В. П. Интеллектуальные системы управления / В. П. Евменов. - Москва : ЛИБРОКОМ, 2009. - 300 с.: ил., табл.

ПРИЛОЖЕНИЕ

```
1 class Motherboard
2 {
3     public string firm;
4     public string modelName;
5     public int price;
6     public string formFactor;
7     public string socket;
8     public string chipset;
9     public string supportedMT;
10    public string numberOfSlots;
11    public int power;
12    public string[] all = new string[9];
13
14    Ссылка: 0
15    public Motherboard(string firm,
16                        string formFactor,
17                        string modelName,
18                        string socket,
19                        string chipset,
20                        string supportedMT, //тип поддерживаемой озу
21                        string numberOfSlots,
22                        int power,
23                        int price)
24    {
25        this.formFactor = all[0] = formFactor;
26        this.firm = all[1] = firm;
27        this.modelName = all[2] = modelName;
28        this.socket = all[3] = socket;
29        this.chipset = all[4] = chipset;
30        this.supportedMT = all[5] = supportedMT;
31        this.numberOfSlots = all[6] = numberOfSlots;
32        this.price = price;
33        all[7] = price.ToString();
34        this.power = power;
35        all[8] = power.ToString();
36    }
37
38    Ссылка: 0
39    public string getStats()
40    {
41        return string.Join(" ", this.all);
42    }
43 }
```

```
C:\Users\sermi\Downloads\Assembly.cs - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Assembly.cs  C6opka.cs
1  class Assembly
2  {
3      private double cap = 0;
4      private Processor poleProc;
5      private Motherboard poleMoth;
6      private MemoryDisc poleMem;
7      private RandomAccessMemory poleRAM;
8      private GraphicsCard poleGCard;
9      private PowerSupply polePower;
10     private void findProcessor(Processor theorProc)
11     {
12         foreach (Processor proc in Storage.processors)
13         {
14             if (proc.coreAmount >= theorProc.coreAmount) & (proc.frequency >= theorProc.frequency) &
15             {
16                 poleProc = proc;
17                 break;
18             }
19         }
20     }
21
22
23
24
25     private void findMother()
26     {
27         foreach (Motherboard moth in Storage.motherboards)
28         {
29             if (poleProc.chipset == moth.chipset) & (poleMoth != moth)
30             {
31                 poleMoth = moth;
32                 break;
33             }
34         }
35     }
36
37
38
39     private void findMem(MemoryDisc theorMem)
40     {
41         foreach (MemoryDisc mem in Storage.memorydiscs)
42         {
43             if ( mem.storageVolume >= theorMem.storageVolume) & (poleMem != mem)
44             {
45                 poleMem = mem;
46                 break;
47             }
48         }
49     }
50 }
```

11 lines, 273 characters selected

Spaces: 4 C#