

Промежуточный отчёт
о ходе выполнения учебной вычислительной практики
на 2 июля 2024 г.

ФИО исполнителя: Майсеенко Матвей Андреевич

Индивидуальное задание: «Создание автоматизированной системы для расчета себестоимости оборудования»

1. План работы:

- 1.1 Изучить инструментальные средства для разработки проекта.
- 1.2 Описать инструментальные средства для разработки проекта.
- 1.3. Разработать и реализовать проект по индивидуальному заданию.
 - 1.3.1 Изучить предметную область.
 - 1.3.2 Обосновать выбор контейнера для реализации проекта.
 - 1.3.3 Описать входные данные проекта.
 - 1.3.4 Реализация выбранных контейнеров и подходящих алгоритмов.
 - 1.3.5 Организация валидации данных, вводимых пользователем.
- 1.4 Описать файловую структуру проекта.
- 1.5 Создание руководства пользователя к проекту.
- 1.6 Оформление отчетной документации
 - 1.6.1 Создание шаблона отчета практики.
 - 1.6.2 Создание шаблона презентации практики.
 - 1.6.3 Оформление итогового отчета по практике.
 - 1.6.4 Создание презентации по итогам практики.

2. На 03.07.2024 выполнены следующие задачи:

№ раздела (в соответствии с планом)	Описание результатов
1.1	<p>Изучена стандартная библиотека шаблонов (STL) и ее компоненты:</p> <ul style="list-style-type: none">1. Контейнер (англ. container) — хранение набора объектов в памяти.2. Итератор (англ. iterator) — обеспечение средств доступа к содержимому контейнера.3. Алгоритм (англ. algorithm) — определение вычислительной процедуры.4. Адаптер (англ. adaptor) — адаптация компонентов для обеспечения различного интерфейса.5. Функциональный объект (англ. functor) — сокрытие функции в объекте для использования другими компонентами. <p>Источник : Стандартная библиотека шаблонов — Википедия</p>

	(wikipedia.org)
1.2	<p>Библиотека стандартных шаблонов (STL) (англ. Standard Template Library) — набор согласованных обобщённых алгоритмов, контейнеров, средств доступа к их содержимому и различных вспомогательных функций в C++.</p> <p>https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2</p> <p>Контейнер - это объект, используемый для хранения других объектов. Контейнер берет на себя управление всей памятью, которую эти объекты занимают. Есть 4 типа: последовательные, ассоциативные, контейнеры-адаптеры, псевдоконтейнеры. https://learn.microsoft.com/ru-ru/cpp/standard-library/stlcontainers?view=msvc-170</p> <p>Итераторы обеспечивают доступ к элементам контейнера и представляют реализацию распространенного паттерна объектноориентированного программирования "Iterator". С помощью итераторов очень удобно перебирать элементы. https://metanit.com/cpp/tutorial/7.3.php</p> <p>Алгоритм STL это специальные шаблонные функции, которые занимаются разного рода обработкой данных в указанных им интервалах. http://dev.mindillusion.ru/algoritmyi-stl/</p> <p>Адаптеры - шаблонные классы, которые обеспечивают отображения интерфейса. https://studfile.net/preview/9596529/page:4/ Функциональный объект (англ. function object), также функтор, функционал и функционид — распространённая в программировании конструкция, позволяющая использовать объект как функцию.</p> <p>https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%B0%D1%8B%D0%B9_%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82</p> <p>В библиотеке для реализации тех или иных задач используются различные коллекции. Для использования коллекции в своем коде используется следующая директива: <code>#include<T></code> , где T — название коллекции</p> <p>Наиболее часто используются:</p>

	<ul style="list-style-type: none"> • vector — коллекция элементов, сохраненных в массиве, изменяющегося по мере необходимости размера (обычно, увеличивающегося); • list — коллекция, хранящая элементы в виде двунаправленного связанного списка; • map — коллекция, сохраняющая пары вида , т.е. каждый элемент — это пара вида <ключ, значение>, при этом однозначная (каждому ключу соответствует единственное значение), где ключ — некоторая характеризующая значение величина, для которой применима операция сравнения; пары хранятся в отсортированном виде, что позволяет осуществлять быстрый поиск по ключу, но за это, естественно, придется заплатить: придется так реализовывать вставку, чтобы условие отсортированности не нарушилось; • set — это отсортированная коллекция одних только ключей, т.е. значений, для которых применима операция сравнения, при этом уникальных — каждый ключ может встретиться во множестве (от англ. set — множество) только один раз; • multimap — map, в котором отсутствует условие уникальности ключа, т.е. если вы произведете поиск по ключу, то получите не единственное значение, а набор элементов с одинаковым значением ключа; для использования в коде используется #include ; • multiset — коллекция с тем же отличием от set'a, что и multimap от map'a, т.е. с отсутствием условия уникальности ключа; для подключения: #include .
1.3.1	<p>Была изучена возможность работы с таблицами в формате csv.</p> <p>https://stackoverflow.com/questions/1120140/how-can-i-read-and-parse-csv-files-in-c</p> <p>Был изучен вопрос: “из чего состоит себестоимость оборудования?”</p>
1.3.2	<p>Обосновать выбор контейнера для реализации проекта: Для хранения параметров деталей и оборудования используются следующие структуры</p> <pre> struct Equipment { //оборудование int id; //id оборудования string Name; //название оборудования map<string, int> DetailsCount; }; struct Detail { //Деталь int id; //id детали string Name; //название детали double costs; //цена детали }; </pre> <p>map - так как названия компонентов оборудования являются уникальными, то можно производить поиск нужной структуры по</p>

	<p>названию детали/оборудования.</p> <p>vector - так как является простым и удобным контейнером для хранения однотипной информации</p>																																																																																																	
1.3.3	<p>Входные данные состоят из двух таблиц в формате csv:</p> <p>1) Таблица деталей:</p> <table><tr><th></th><th>A</th><th>B</th><th>C</th></tr><tr><td>1</td><td>11</td><td>Брусok</td><td>100</td></tr><tr><td>2</td><td>12</td><td>Гвоздь</td><td>15</td></tr><tr><td>3</td><td>13</td><td>Болт</td><td>20</td></tr><tr><td>4</td><td>14</td><td>Гайка</td><td>15</td></tr><tr><td>5</td><td>15</td><td>Железная пластина</td><td>120</td></tr><tr><td>6</td><td>16</td><td>Корпус механизма</td><td>300</td></tr><tr><td>7</td><td>17</td><td>Лампочка</td><td>150</td></tr><tr><td>8</td><td>18</td><td>Плата</td><td>1000</td></tr><tr><td>9</td><td>19</td><td>Процессор</td><td>1300</td></tr><tr><td>10</td><td>110</td><td>Системный блок</td><td>500</td></tr><tr><td>11</td><td>111</td><td>Жесткий диск</td><td>900</td></tr><tr><td>12</td><td>112</td><td>Железная балка</td><td>200</td></tr></table> <p>Которая составляется по принципу: первый столбец – id детали второй столбец – название детали третий столбец – цена за единицу детали</p> <p>2) Таблица оборудования:</p> <table><tr><th></th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th><th>G</th><th>H</th></tr><tr><td>1</td><td>21</td><td>Компьютер</td><td>Жесткий диск</td><td>1</td><td>Системный блок</td><td>1</td><td>Процессор</td><td>1</td></tr><tr><td>2</td><td>22</td><td>Стул</td><td>Гвоздь</td><td>6</td><td>Брусok</td><td>2</td><td></td><td></td></tr><tr><td>3</td><td>23</td><td>Стол</td><td>Брусok</td><td>6</td><td>Гвоздь</td><td>4</td><td></td><td></td></tr><tr><td>4</td><td>24</td><td>Лампа</td><td>Железная пластина</td><td>3</td><td>Железная балка</td><td>1</td><td>Лампочка</td><td>1</td></tr></table> <p>Которая составляется по принципу: первый столбец – id оборудования второй столбец – название оборудования все последующие нечетные столбцы – необходимая деталь все последующие четные столбцы – необходимое количество соответствующей детали</p>		A	B	C	1	11	Брусok	100	2	12	Гвоздь	15	3	13	Болт	20	4	14	Гайка	15	5	15	Железная пластина	120	6	16	Корпус механизма	300	7	17	Лампочка	150	8	18	Плата	1000	9	19	Процессор	1300	10	110	Системный блок	500	11	111	Жесткий диск	900	12	112	Железная балка	200		A	B	C	D	E	F	G	H	1	21	Компьютер	Жесткий диск	1	Системный блок	1	Процессор	1	2	22	Стул	Гвоздь	6	Брусok	2			3	23	Стол	Брусok	6	Гвоздь	4			4	24	Лампа	Железная пластина	3	Железная балка	1	Лампочка	1
	A	B	C																																																																																															
1	11	Брусok	100																																																																																															
2	12	Гвоздь	15																																																																																															
3	13	Болт	20																																																																																															
4	14	Гайка	15																																																																																															
5	15	Железная пластина	120																																																																																															
6	16	Корпус механизма	300																																																																																															
7	17	Лампочка	150																																																																																															
8	18	Плата	1000																																																																																															
9	19	Процессор	1300																																																																																															
10	110	Системный блок	500																																																																																															
11	111	Жесткий диск	900																																																																																															
12	112	Железная балка	200																																																																																															
	A	B	C	D	E	F	G	H																																																																																										
1	21	Компьютер	Жесткий диск	1	Системный блок	1	Процессор	1																																																																																										
2	22	Стул	Гвоздь	6	Брусok	2																																																																																												
3	23	Стол	Брусok	6	Гвоздь	4																																																																																												
4	24	Лампа	Железная пластина	3	Железная балка	1	Лампочка	1																																																																																										
1.3.4	<p>Для чтения данных из таблиц были созданы следующие алгоритмы:</p> <pre>void readDetailsFile(istream& DetailFile, GlobalList& List) {</pre>																																																																																																	

	<pre> int DetailCounter = 0; string curDetailInfo; while (getline(DetailFile, curDetailInfo)) { istringstream iss(curDetailInfo); Detail curDetail; string curId; getline(iss, curId, ','); curDetail.id = stoi(curId); string curName; getline(iss, curName, ','); curDetail.Name = curName; string detailCost; getline(iss, detailCost); curDetail.costs = stod(detailCost); DetailsNameMap[curName] = curDetail; DetailList.push_back(curDetail); DetailCounter++; } } void readEquipmentFile(istream& EquipmentFile, GlobalList& List) { int EquipmentCounter = 0; while (!EquipmentFile.eof()) { Equipment curEquipment; string curId; getline(EquipmentFile, curId, ','); curEquipment.id = stoi(curId); string curName; getline(EquipmentFile, curName, ','); curEquipment.Name = curName; string curInfo; getline(EquipmentFile, curInfo); istringstream iss(curInfo); string detail; string detailCount; while (getline(iss, detail, ',') && !detail.empty()) { getline(iss, detailCount, ','); int NeedCount = stoi(detailCount); curEquipment.DetailsCount[detail] = NeedCount; } cout << endl << endl; EquipmentNameMap[curName] = curEquipment; </pre>
--	--

	<pre> EquipmentList.push_back(curEquipment); EquipmentCounter++; } EquipmentCount = EquipmentCounter; } </pre>
1.3.5	в разработке...
1.4	
1.5	
1.6.1	
1.6.2	
1.6.3	
1.6.4	