Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

Тамбовский государственный технический университет

Кафедра

Пояснительная записка

к курсовой работе по дисциплине

«Автоматизация конструкторского и технологического проектирования»

Выполнил студент гр. –41 Д. Ю.

Проверил: Ю. В.

Тамбов

**Содержание**

|  |  |
| --- | --- |
|  | Введение 3 |
| 1. | Описание алгоритма решения задачи автоматизированного проектирования технологического процесса 3 |
| 2. | Информационное обеспечение разрабатываемой подсистемы 9 |
| 3. | Описание диалоговой подсистемы 12 |
| 4. | Примеры полученной технологической информации 13 |
|  | Заключение 14 |
|  | Список используемых источников 15 |
|  | Приложение А 16 |

**Введение.**

Целью данной работы по дисциплине «Автоматизация конструкторского и технологического проектирования» является приобретение навыков разработки системы составления маршрутной документации на стадии планирования будущего производства.

Заданием работы было поставлено следующее: «Разработать АСТП подготовительного производства полимерпесчанной продукции методом нового планирования».

Необходимо создать удобную диалоговую подсистему формирования части маршрутной карты, где входными данными является результат ручного или автоматически подобранного списка оборудования проектируемой производственной линии.

1. **Описание алгоритма решения задачи автоматизированного проектирования технологического процесса.**

В решении поставленной задачи было использовано несколько вариантов. Первый, наиболее простой заключается в следующем, пользователь в диалоговом режиме выбирает необходимые элементы производственной линии, которые заключены в единую базу данных, внутри разбиты на отдельные каталоги для каждого вида оборудования. Все элементы, выбранные ранее, формируют последовательность технологического производства. Каждый элемент последовательности имеет свои уникальные характеристики, взятые из базы ранее. После стадии формирования линии следует заполнение таблицы операций.

В режиме автоматического подбора оборудования создаваемой линии программа выполняет поиск элементов в базе по критерию, в данном случае был взят критерий максимальной стоимости. Отсутствие необходимого компонента в базе создаст исключительную ситуацию, при которой будет сформирован запрос пользователю на принятие каких-либо действий, пропустить операцию или выполнить вставку временного элемента с последующей правкой его на конкретное оборудование из базы. По окончанию следует заполнение таблицы операций.

Операции закреплены за типом оборудования, логика следующая, например в погрузочном контейнере можно что-либо перевозить, загружать\разгружать, выполнить операцию измельчения в нем невозможно, следовательно итогом будет операция загрузки.

Следующим шагом предстоит заполнение всех полей раздела А, Б, О, Т карты [3].

Раздел А:

|  |  |
| --- | --- |
| Обозначение служебного символа | Содержание информации вносимой в графы, расположенные на строке |
| A | Номер цеха, участка, рабочего места, где выполняется операция, номер операции, код и наименование операции, обозначение документов, применяемых при выполнении операции |

Строки для заполнения следующие:

|  |  |  |
| --- | --- | --- |
| №п/п | Наименование (условное обозначение) графы | Содержание информации |
| 1 | Цех | Номер (код) цеха, в котором выполняется операция |
| 2 | Уч. | Номер (код) участка, конвейера, поточной линии и т.п. |
| 3 | РМ | Номер (код) рабочего места |
| 4 | Опер. | Номер операции (процесса) в технологической последовательности изготовления или ремонта изделия (включая контроль и перемещение) |
| 5 | Код, наименование операции | Код операции по технологическому классификатору, наименование операции. |
| 6 | Обозначение документа | Обозначение документов [4], инструкций по охране труда, применяемых при выполнении данной операции. Состав документов следует указывать через разделительный знак «;» с возможностью, при необходимости, переноса информации на последующие строки |

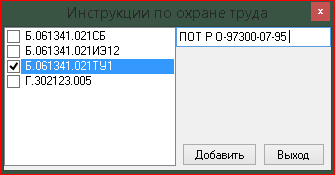
Все поля открыты для заполнения вручную. Строки 1-5 заполняются программой при выборе очередной операции из соответствующей таблицы. Строка 4 заполняется последовательно с промежутком в пять пунктов, это гарантирует резерв на будущее, если произойдут изменения или правки технологического процесса. Строка 6 выполняет роль диалога с отображение окна, в котором пользователь вправе выбрать нужный ему документ, либо, при отсутствии таковых, внести свой задав его название в отдельной графе и нажав кнопку «Добавить» (Рисунок 1).

Рисунок Выбор документов ОТ

Раздел Б:

|  |  |
| --- | --- |
| Обозначение служебного символа | Содержание информации вносимой в графы, расположенные на строке |
| Б | Код, наименование оборудования и информация по трудозатратам |

Строки для заполнения:

|  |  |  |
| --- | --- | --- |
| №п/п | Наименование (условное обозначение) графы | Содержание информации |
| 1 | Код, наименование оборудования | Код оборудования по классификатору, краткое наименование оборудования, его инвентарный номер |
| 2 | СМ | Степень механизации (код степени механизации) |
| 3 | Проф. | Код профессии по классификатору ОКПДТР |
| 4 | Р | Разряд работы, необходимый для выполнения операции |
| 5 | УТ | Код условий труда по классификатору ОКПДТР и код вида нормы |
| 6 | КР | Количество исполнителей, занятых при выполнении операции |
| 7 | КОИД | Количество одновременно изготавливаемых (обрабатываемых, ремонтируемых) деталей (сборочных единиц) при выполнении одной операции |
| 8 | ЕН | Единица нормирования, на которую установлена норма расхода материала или норма времени |
| 9 | ОП | Объем производственной партии в штуках. На стадиях разработки предварительного проекта и опытного образца допускается графу не заполнять. |
| 10 | Кшт | Коэффициент штучного времени при многостаночном обслуживании |
| 11 | Тпз | Норма подготовительно-заключительного времени на операцию |
| 12 | Тшт | Норма штучного времени на операцию |

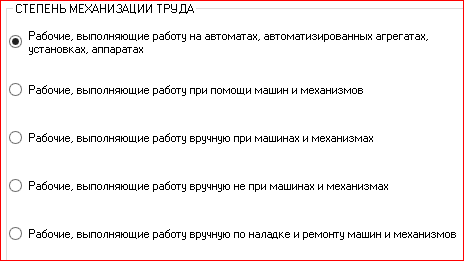
Строки 1, 10-12 заполняются автоматически. Строка 2 предоставляет выбор степени механизации в появившемся справа поле выбора (Рисунок 2)[2].

Рисунок Степень механизации

Строка 3 заполняется выбором соответствующего кода профессии (Рисунок 3)[2].

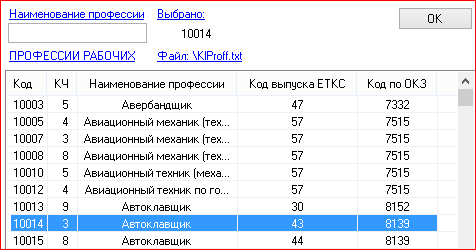
Данные загружаются из специально составленной базы – текстового файла, внутри которого каждое поле разделяется пробелом.

Рисунок Классификатор профессий

Строки 4, 5 заполняются выбором пунктов из полей справа. Строки 6-9 предназначены для ввода чисел пользователем.

Строки 10, 11, 12 рассчитываются по соответствующим формулам:

Штучная норма времени– Тшт. рассчитывается по формуле[6]:

Время на отдых и личные надобности может устанавливаться в минутах и в процентах от оперативного времени:

отсюда

, где Котд. – коэффициент отдыха, %

Время технического обслуживания – это время, расходуемое на действия рабочего по уходу за рабочим местом, на смену инструмента, на подналадку оборудования и т.д. Оно часто устанавливается в процентах от оперативного времени, т.е.

отсюда

, где Кобс. – коэффициент обслуживания рабочего места, %.

Вспомогательное время – Твсп. – это время, расходуемое на действия, обеспечивающие выполнение технологической части операции. Это время на установку детали на станок и снятие ее после обработки, на подвод и отвод инструмента, пуск и остановку станка и т.д.

Оперативное время – Топ. – это  время, необходимое на непосредственное выполнение операции. Оно состоит из основного и дополнительного времени.

, где Д – длина обработки, мм. П – подача, мм/об. С – скорость резания, об/мин

Норматив вспомогательного времени определяется по нормативам для технического нормирования выполняемых работ (например для токарных Нвсп=40%)

Коэффициент штучного времени - отношение затрат на выполнение рабочим-многостаночником технологической операции к сумме затрат рабочего времени по всем операциям, выполняемым на рабочем месте при многостаночном обслуживании[7].

Коэффициент штучного времени (Кшт) при многостаночном обслуживании вычисляют по формуле:

сумма норм штучного времени технологических операций, выполняемых на рабочем месте;

n - количество операций, выполняемых на рабочем месте;

i - порядковый номер технологической операции по технологическому процессу.

Расчёт подготовительно-заключительного времени. Норматив подготовительно-заключительного времени зависит от времени на наладку станка, определяемого способом установки детали и количеством инструментов, участвующих при выполнении операции и времени, затрачиваемого в случаях работы с каким-либо дополнительным нерегулярно встречающимся в работе приспособлением или устройством, предусмотренным технологическим процессом на данную операцию[8].

Подготовительно-заключительное время в нормативах рассчитано на организационные условия производства, при которых доставка технической документации, нарядов, инструмента и приспособлений к рабочему месту производится вспомогательным обслуживающим персоналом.

, где Тпз1 – затраты по времени (приемов), это время на получение наряда, чертежа, технологической документации на рабочем месте в начале работы и на сдачу в конце смены;

Тпз2 – затраты по времени на дополнительные работы;

Тпз3 – затраты по времени на пробную обработку детали;

Раздел О - Содержание операции (перехода) заполняется автоматически с возможностью внесения правок в описание. Зависит от типа операций.

Раздел Т - Информация о применяемой при выполнении [операции технологической](http://www.gosthelp.ru/text/Metodicheskierekomendacii319.html) оснастке заполняется пользователем.

1. **Информационное обеспечение разрабатываемой подсистемы.**

Часть базы классификации профессий:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код | КЧ | Наименование профессии | Код выпуска ЕТКС | Код по ОКЗ |
| 10003 5 Авербандщик 47  10005 4 Авиационный механик (техник) по планеру 57  и двигателям  10007 3 Авиационный механик (техник) по приборам 57  и электрооборудованию  10008 8 Авиационный механик (техник) по 57  радиооборудованию  10010 5 Авиационный техник (механик) по 57  парашютным и аварийно-спасательным  средствам  10012 4 Авиационный техник по горюче-смазочным 57  материалам  10013 9 Автоклавщик 30  10014 3 Автоклавщик 43  10015 8 Автоклавщик 44  10016 2 Автоклавщик литья под давлением 02  10017 7 Автоклавщик на запарке брикетов 04 | | | | 7332  7515  7515  7515  7515  7515  8152  8139  8139  8122  8112 |

База данных Агломератор:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | производительность | Размер гранулы | чвротор | мощность | габариты | загрузпроем | ножи | масса |
| АГМТ30 | 570000 | agm90s.jpg | 100 | 8 | 1500 | 30 | 1800 х 950 х 1465 | 200 | 12 | 400 |
| АГМ-15 | 300000 | agglomerator.jpg | 70 | 9 | 1500 | 15 | 1400х650х1250 | 510 | 4 | 600 |
| АГМ-75 | 450000 | agm55.gif | 250 | 5 | 2000 | 75 | 2200х1000х1900 | 820 | 6 | 1750 |
| OULI-300 | 645000 | ouli-200.jpg | 300 | 5 | 2000 | 37 | 2000 x 1000 x 1400 | 300 | 9 | 1400 |

База данных Дробилка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | производительность | мощность | загрузпроем | ротацнож | стационнож | габариты | масса |
| HSS200 | 80000 | hss\_a.jpg | 100 | 5.5 | 250х400 | 20 | 2 | 795х770х1300 | 500 |
| HSS 230 A | 84368 | hss\_a.jpg | 200 | 4 | 230x160 | 6 | 2 | 1080x760x1200 | 500 |
| HSS 300 A | 84368 | hss\_a.jpg | 300 | 5.5 | 300x210 | 9 | 2 | 1310x930x1310 | 540 |
| HSS 400 A | 84368 | hss\_a.jpg | 550 | 7.5 | 400x250 | 12 | 2 | 1420x1070x1530 | 550 |
| HSS 500 A | 84368 | hss\_a.jpg | 580 | 11 | 500x250 | 15 | 2 | 1530x1170x1530 | 500 |
| HSS 600 A | 84368 | hss\_a.jpg | 600 | 15 | 600x310 | 18 | 4 | 1610x1290x1820 | 550 |
| HSS 700 A | 100000 | hss\_a.jpg | 700 | 18.5 | 700x310 | 21 | 4 | 1610x1390x1820 | 550 |
| HSS 800 A | 100000 | hss\_a.jpg | 800 | 22 | 800x420 | 24 | 4 | 1860x1610x2280 | 560 |
| HSS 800 C | 130000 | hss\_c.jpg | 800 | 22 | 800x420 | 39 | 4 | 1860x1610x2280 | 1850 |
| HSS 300 B | 193343 | hss\_b.jpg | 50 | 5.5 | 300x210 | 3 | 2 | 1310х930х1310 | 428 |
| WT 30-80 | 124600 | 1x\_shreder.jpg | 500 | 30 | 1280х880 | 42 | 4 | 2466x1677x1700 | 1950 |

База данных Экструдер:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | производительность | скоростьшнека | регулиртемпература | погрешностьтемператур | мощность | габариты |
| SJSZ | 1900000 | shekkovuy\_ekstruder\_iz\_kitaya.jpg | 500 | 25 | от 100 до 400 | 0.1 | 37 | 4200 х 680 х 1200 |

База данных Контейнер:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | описание | цена | фото | температура | размеры | грузоподьемность | вес | толщина | конструкция | окраска | цвет | обьем |
| КГ-12 | Тара (контейнер металлический) предназначен для хр... | 23873 | i-423211-01.jpg | от -40С до +40С | 1000х1700х1000 | 1200 | 220 | 3 | сварная | ПФ-эмаль | серый | 100 |
| КГ-01 | • Грузовой контейнер типа КГ предназначен для тран... | 5283 | kg01\_a\_big.jpg | от +40С до -40С | 450x600x400 | 500 | 22 | 3 | сварная | ПФ-эмаль | серый или синий | 40 |
| КГ-17 | • Металлический контейнер типа КГ предназначен для... | 13600 | kg17\_big.jpg | от +40С до -40С | 1050х1600х1000 | 800 | 106 | 3 | сварная | ПФ-эмаль | серый или синий | 1300 |
| КС-03 | • Контейнер металлический типа КС предназначен для... | 6796 | ks03\_big\_2.jpg | от +40С до -40С | 600х600х800 | 1200 | 33 | 4 | сетчатый | ПФ-эмаль | серый или синий | 180 |

База данных Конвеер:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | ширина | длина | скорость | наклон | привод | дипривод | динатяжного | дироликов |
| КЛС4 | 90000 | tr7\_300.jpg | 400 | 200 | 0.2-2.2 | 0-18 | Мотор-редуктор | 220-420 | 220-420 | 51-108 |
| КЛС5 | 90000 | tr7\_300.jpg | 500 | 250 | 0.2-2.5 | 0-18 | Мотор-редуктор | 220-530 | 220-530 | 51-127 |
| КЛС6 | 105000 | 99e8b20e608b86227f56703b58bffb34.jpg | 650 | 300 | 0.2-2.5 | 0-18 | Мотор-редуктор | 220-530 | 220-530 | 51-159 |
| КЛС8 | 105000 | 99e8b20e608b86227f56703b58bffb34.jpg | 800 | 500 | 0.25-2.5 | 0-18 | Мотор-редуктор | 220-530 | 220-530 | 76-159 |
| КЛС10 | 113500 | 99e8b20e608b86227f56703b58bffb34.jpg | 1000 | 500 | 0.25-3.2 | 19-45 | Мотор-редуктор | 320-530 | 320-530 | 76-159 |
| КЛС12 | 120000 | 99e8b20e608b86227f56703b58bffb34.jpg | 1200 | 500 | 0.25-3.5 | 19-45 | Мотор-редуктор | 320-530 | 320-530 | 76-159 |
| КЛС14 | 120000 | 99e8b20e608b86227f56703b58bffb34.jpg | 1400 | 500 | 0.25-3.5 | 19-45 | Мотор-редуктор | 420-630 | 420-630 | 108-159 |
| КЛС16 | 125000 | 99e8b20e608b86227f56703b58bffb34.jpg | 1600 | 700 | 0.4-3.5 | 19-45 | Цилиндро-конический | 420-630 | 420-630 | 108-159 |
| КЛС18 | 125000 | 99e8b20e608b86227f56703b58bffb34.jpg | 1800 | 700 | 0.4-3.5 | 19-45 | Планетарный | 420-830 | 420-830 | 127-159 |
| КЛС20 | 125000 | 99e8b20e608b86227f56703b58bffb34.jpg | 2000 | 700 | 0.4-3.5 | 0-45 | Мотор-барабан | 530-830 | 530-830 | 127-159 |
| КЛС22 | 130000 | 99e8b20e608b86227f56703b58bffb34.jpg | 2200 | 700 | 0.6-3.5 | 45-90 | Мотор-барабан | 530-830 | 530-830 | 127-159 |
| КЛС11ф | 130000 | 99e8b20e608b86227f56703b58bffb34.jpg | 2100 | 750 | 0.6-3.5 | 19-45 | Цилиндро-конический | 530-630 | 530-630 | 76-159 |
| КЛ-600-7 | 240000 | 104857807\_w640\_h640\_500\_produce\_1370329021.jpg | 600 | 700 | 0.6 | 35 | Мотор-барабан | 220-420 | 220-420 | 51-108 |

База данных Магнит:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | габариты | ширина | глубина | масса |
| СМПА 500 | 15000 | 12ed41a4e08de10d7b6696134afb454a.png | 1115х1073х380 | 500 | 300 | 450 |
| СМПА 650 | 15000 | Снимок.PNG | 1320х1073х380 | 650 | 350 | 600 |
| СМПА-ТМ 1200 | 25000 | 12ed41a4e08de10d7b6696134afb454a.png | 2185х1950х490 | 1200 | 450 | 2700 |
| СМПА-М 1600 | 25000 | 12ed41a4e08de10d7b6696134afb454a.png | 2521х1748х490 | 1600 | 450 | 2150 |

База данных Пескосушилка:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| имя | цена | фото | производительность | фракциявыход | влажностьвыход | чвротор | мощность | температуракамеры | времянагревакамеры | времясушки | загрузка | габариты | масса |
| Р-03 | 150000 | 422.jpg | 300 | 3 | 1 | 4 | 9 | 120 | 40 | 12 | 50 | 2475 х 950 х1465 | 1000 |
| Р-07л | 119000 | 2090109200.jpg | 250 | 3.8 | 1 | 4 | 9 | 130 | 45 | 11 | 40 | 2475 х 950 х1465 | 1000 |

1. **Описание диалоговой подсистемы.**

Диалоговая подсистема выполнена в виде меню-диалог, заполнения бланка. Пользователь может взаимодействовать с пунктами описания операций, заполнять поля ввода, править итоговую таблицу карты.

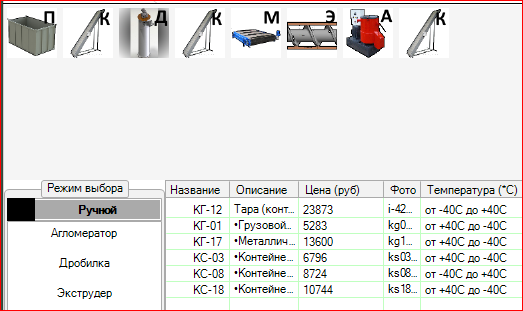
Пользователю предоставляется выбрать оборудование из таблиц и сформировать итоговые последовательности линии.

Рисунок Выбор оборудования

1. **Примеры полученной технологической информации.**

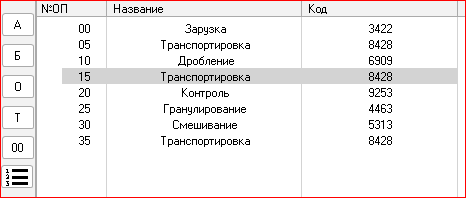
Этапы построения линии (Рисунок 4) сведены в таблицу ниже (Рисунок 5):

Рисунок Таблица операций

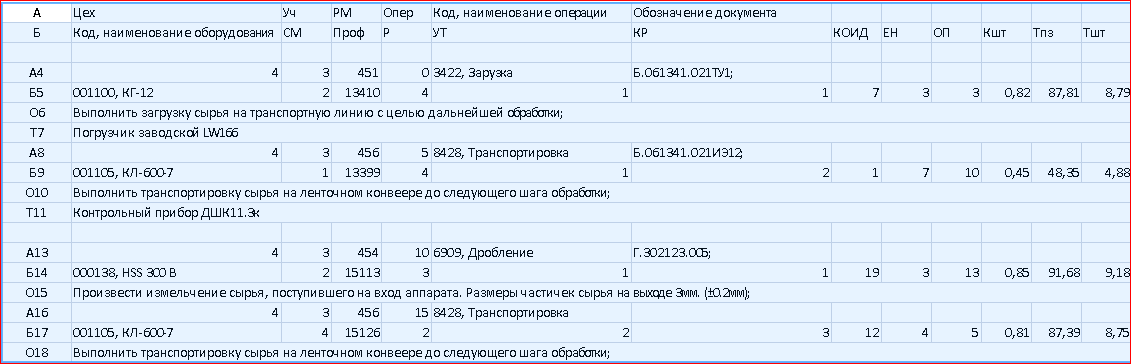
Результирующей таблицей служит маршрутная карта (Рисунок 6)

Рисунок Маршрутная карта

**Заключение.**

По итогам выполнения данной работы было разработано приложение, выполняющее роль помошника, способного подобрать оборудование, составить технологическую цепочку из этапов операций, и как результат, формирование маршрутной документации – карты. Удобство в использовании также обеспечивает внутренняя справочная система помощи.

**Список используемых источников.**

1. Комплектность и правила заполнения бланков технологических документов: Методическое пособие для самостоятельной работы, интернет ресурс: <http://vunivere.ru/work54400/page8>;
2. Общероссийский классификатор профессий рабочих, должностей служащих и тарифных разрядов (ОКПДТР), интернет ресурс: <http://www.kadrovik.ru/docs/okpdtr.htm>;
3. ГОСТ 3.1118-82 ЕСТД. Формы и правила оформления маршрутных карт, интернет ресурс: <http://www.gosthelp.ru/text/GOST3111882ESTDFormyiprav.html>;
4. ГОСТ 2.201-80 ЕСКД. Обозначение изделий и конструкторских документов, интернет ресурс: <http://www.gosthelp.ru/text/GOST220180ESKDOboznacheni.html>;
5. В.В. Быков, И.Г. Голубев, В.В. Каменский, «Проектирование технологических процессов восстановления деталей транспортных и технологических машин», Москва – 2013;
6. Техническое нормирование труда, интернет ресурс: <http://okolovesna.yomu.ru/3b.htm>;
7. Р 50-72-88 Рекомендации. Единая система технологической документации, порядок оформления документов, применяемых при нормировании технологических процессов (операций), интернет ресурс: <http://gostrf.com/normadata/1/4293850/4293850380.htm>;
8. Р. Г. Гришин, Н. В. Лысенко, Н. В. Носов, «Нормирование станочных работ. Определение вспомогательного времени при механической обработке заготовок», учебное пособие, интернет ресурс: <http://tm.samgtu.ru/sites/tm.samgtu.ru/files/normirovanie.pdf>;

**Приложение А**

Общие сведения

Наименование программы: MapTech, написана на языке программирования С# в среде разработки Microsoft Visual Studio 2008 для Windows Desktop версии 6.3.9600.00.

Функциональное назначение

Программа выполняет построение технологической линии производства полимерпесчанной смеси с заполнением полей маршрутной карты.

Описание логической структуры

Программа делится на 3 подпрограммы:

1. Создание диалоговой подсистемы выбора оборудования.
2. Формирование таблицы операций.
3. Вывод результатов в виде бланка-таблицы.

Используемые технические средства

Программа была разработана в среде Microsoft Visual Studio 2008 для Windows Desktop. На компьютере с операционной системой Windows 8, Intel core i-7,ОЗУ 8 GB, ПЗУ 500 GB.

Вызов и загрузка программы

Запуск программы осуществляется через файл MapTech.exe.

После старта программы произойдет загрузка содержимого баз данных с исходной информацией, оборудовании. Отобразится окно выбора режимов проектирования.

Входные данные

БД агрегатов Агломератор, . Дробилка, Конвеер, Контейнер, Металлодетектор, Пескосушилка, Экструдер. А также таблица Klassifikator.

Выходные данные

Маршрутная карта с полями А, Б, О, Т.

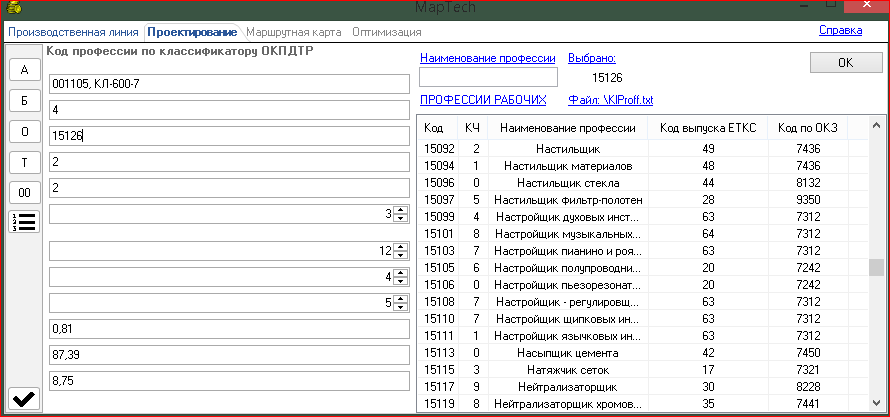


Рисунок Раздел Б

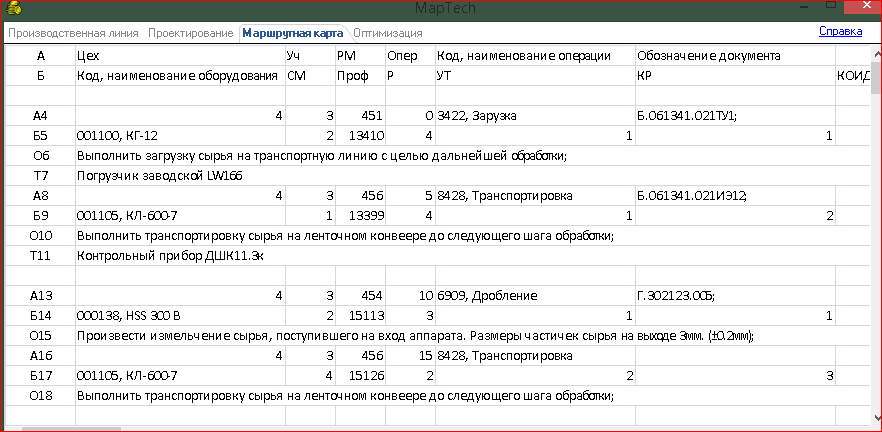
Код программы

Рисунок Маршрутная карта

//ГОСТ 3.1104-81 http://www.gosthelp.ru/text/GOST3111882ESTDFormyiprav.html

**namespace** MapTech

**{**

**public** partial class Form1 **:** Office2007Form

**{**

/// <summary>

/// Помощь

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

**private** void linkLabel\_Help\_LinkClicked**(object** sender**,** LinkLabelLinkClickedEventArgs e**)**

**{**

superTabControl\_КонтейнерВкладок**.**SelectedTabIndex **=** 4**;**

superTabControl\_КонтейнерВкладок**.**SelectedTab **=** superTabItem1\_HelpTab**;**

superTabControl\_КонтейнерВкладок**.**SelectedPanel **=** superTabControlPanel1\_Help**;**

pageSlider1**.**SelectedPageIndex **=** 0**;**

index **=** 1**;**

**}**

/// <summary>

/// Печать Маршрутной Карты

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

**private** void ПечатьtoolStripMenuItem\_Click**(object** sender**,** EventArgs e**)**

**{**

PrinterNull print **=** **new** PrinterNull**();**

print**.**ShowDialog**();**

**}**

//Для исключения вылетов гасим контрол при выходе

**private** void Form1\_FormClosed**(object** sender**,** FormClosedEventArgs e**)**

**{**

reoGridControl\_МаршрутнаяКарта.Dispose();

}

private int index = 1;

//Кнопка Следующий слайд

private void buttonX1\_NextSlideHelp\_Click(object sender, EventArgs e)

{

if (index > pageSlider1.PageCount - 1) index = 0;

pageSlider1.SelectedPageIndex = index++;

}

public Form1()

{

InitializeComponent();

var sheet = reoGridControl\_МаршрутнаяКарта.CurrentWorksheet;

//Отключить отображение Шапки и Нумерации

sheet.DisableSettings(WorksheetSettings.View\_ShowRowHeader);

sheet.DisableSettings(WorksheetSettings.View\_ShowColumnHeader);

\_clTablica.СоздатьШапкуКарты(reoGridControl\_МаршрутнаяКарта);

\_clTablica.ПустаяСтрока(reoGridControl\_МаршрутнаяКарта);

очиститьПолеToolStripMenuItem\_Click("", null);

//Pan();

}

private clTablica \_clTablica = new clTablica();

/// <summary>

/// Раскрытие формы

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void Form1\_Resize(object sender, EventArgs e)

{

//развернуть компонент нечеткой логики

ucNeLogik1.Width = groupPanel1\_КритерииАвтоВыбора.Width-10;

ucNeLogik1.Height = groupPanel1\_КритерииАвтоВыбора.Height-10;

}

private void radioButton1\_НечеткоеПравило\_CheckedChanged(object sender, EventArgs e)

{

if (radioButton1\_МаксимальнаяСтоимость.Checked)

ucNeLogik1.Visible = !radioButton1\_МаксимальнаяСтоимость.Checked;

else ucNeLogik1.Visible = !radioButton1\_МаксимальнаяСтоимость.Checked;

//Запуск алгоритма автопостроения

НачатьАвтоПостроениеЛинии();

}

private clDBExec \_clDBexec = new clDBExec();

/// <summary>

/// Выбран один из 7 пунктов

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonItem1\_7\_Click(object sender, EventArgs e)

{

//Надо достать опознаватель:

string opozn = (sender as ButtonItem).Tag.ToString();

\_structImg.\_tag = opozn;

//определить пункт

//выполнить запрос в базу+заголовки подготовить

//вывести результат в таблицу Adv

var masHd = \_clDBexec.GetHeaderTable(opozn);

LoadHeader(masHd);

\_clDBexec.Select(opozn);

AddSampleData();

}

/// <summary>

/// Заголовки зависят от выбранной Таблицы

/// </summary>

private void LoadHeader(IEnumerable<ColumnHeader> massivH)

{

//чистим предыдущее

advTree1.Columns.Clear();

foreach (var columnHeader in massivH)

{

advTree1.Columns.Add(columnHeader);

}

}

/// <summary>

/// Записать инфу с БД (SELECT) в AdvTree и вывести

/// </summary>

private void AddSampleData()

{

//Заполнить таблицу SELECT

int index = 0;

if (\_clDBexec.ListSelectFull == null) return;

//Чистим предыдущие

advTree1.Nodes.Clear();

foreach (AdvTree tree in \_clDBexec.ListSelectFull)

{

while (index < Convert.ToInt32(\_clDBexec.Count))

{

advTree1.Nodes.Add(tree.Nodes[index]);

index++;

}

}

}

private string \_tagActivDrgDrop;

/// <summary>

/// Двойной клик по ноде - выполнить Бросить компонент на Flow

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void advTree1\_NodeDoubleClick(object sender, TreeNodeMouseEventArgs e)

{

if(e.Button != MouseButtons.Left) return;

\_tagActivDrgDrop = e.Node.SelectedCell.Parent.FullPath;

//MessageBox.Show(\_tagActivDrgDrop);

AddPanelToFlowLayoutPanel();

//Выполнить заполнение Таблицы Операций -> Проектирование\Проектирование ТМ

ЗаполнитьТаблицуОпераций(\_structImg.\_tag, \_tagActivDrgDrop);

}

private Panel \_panel;

private clDBExec.PanelImage \_structImg = new clDBExec.PanelImage("null");

/// <summary>

/// Добавить очередрую панель на поле

/// </summary>

private void AddPanelToFlowLayoutPanel(params string[] param)

{

reflectionLabel1\_НетКомпонентов.Visible = false;

\_panel = new Panel

{

Size = new Size(50, 50),

BackgroundImage =

param.Length == 0

? \_structImg.GetPanelImageForKey(\_structImg.\_tag)

: \_structImg.GetPanelImageForKey(param[0]),

BackgroundImageLayout = ImageLayout.Stretch

};

flowLayoutPanel\_ВизуальныйКонтейнер.Controls.Add(\_panel);

}

private void очиститьПолеToolStripMenuItem\_Click(object sender, EventArgs e)

{

//удалить все панели с контейнера

flowLayoutPanel\_ВизуальныйКонтейнер.Controls.Clear();

//показать ПУСТО лейбл

reflectionLabel1\_НетКомпонентов.Visible = true;

//Затереть \_OP

\_OP = 00;

//Очистить содержимое в таблице AdvTree

advTree2\_ТаблОпераций.ClearAndDisposeAllNodes();

//Почистить формы ввода от старых заполнителей

ОчиститьВсеПоляВвода();

//Затереть массив с ИМЕНАМИ станков

\_clDBexec.НазванияОборудованияПоСтоимости = new List<string>();

//Повысить номер цеха

\_НомерЦеха++;

//Чистим массив случайных времен Тшт

\_Тшт = new List<double>();

//Чистим массив случайных времен Тпз

\_Тпз = new List<double>();

}

private int \_НомерЦеха = 01;

/// <summary>

/// Переключатель с АВТО\РУЧНОЙ

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void switchButton\_РучнойАвто\_ValueChanged(object sender, EventArgs e)

{

//Активировать ручной режим составления линии

//деактивировать панель

itemPanel\_СемьЭлементов.Enabled = advTree1.Visible = !switchButton\_РучнойАвто.Value;

groupPanel1\_КритерииАвтоВыбора.Visible = switchButton\_РучнойАвто.Value;

//Запуск алгоритма подбора оборудования в автоматическом режиме

НачатьАвтоПостроениеЛинии();

}

/// <summary>

/// Запуск алгоритма подбора

/// </summary>

private void НачатьАвтоПостроениеЛинии()

{

if (!switchButton\_РучнойАвто.Value) return;//для Ручной

//Чистим пано

очиститьПолеToolStripMenuItem\_Click("", null);

//для Авто

//по максимальной стоимости:

if(radioButton1\_МаксимальнаяСтоимость.Checked)

foreach (string tag in \_clDBexec.GetAllCost())

{

AddPanelToFlowLayoutPanel(tag);

//Выполнить заполнение Таблицы Операций -> Проектирование\Проектирование ТМ

ЗаполнитьТаблицуОпераций(tag, "");

}

//Заполнить Поля Маршрутной карты

//foreach (string s in \_clDBexec.НазванияОборудованияПоСтоимости)

//{

// \_clTablica.ТstrokaSet(new clTablica.Т(s, reoGridControl\_МаршрутнаяКарта));

// \_clTablica.ПустаяСтрока(reoGridControl\_МаршрутнаяКарта);

//}

}

private int \_OP = 00;

/// <summary>

/// Последовательность операций

/// </summary>

private void ЗаполнитьТаблицуОпераций(string modul, string name)

{

//Очистить содержимое в таблице AdvTree//advTree2\_ТаблОпераций.ClearAndDisposeAllNodes();

//NOTE: Подобрать очередному модулю его операции

{

var node = new Node(string.Format("{0:0#}", \_OP));

switch ((clDBExec.Table)Enum.Parse(typeof(clDBExec.Table), modul))

{

case clDBExec.Table.aglomerator:

node.Cells.Add(new Cell(clTablica.OperationEnum.Смешивание.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Смешивание).ToString()));

break;

case clDBExec.Table.drobilka:

node.Cells.Add(new Cell(clTablica.OperationEnum.Дробление.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Дробление).ToString()));

break;

case clDBExec.Table.ekstruder:

node.Cells.Add(new Cell(clTablica.OperationEnum.Гранулирование.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Гранулирование).ToString()));

break;

case clDBExec.Table.konteiner:

node.Cells.Add(new Cell(clTablica.OperationEnum.Зарузка.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Зарузка).ToString()));

break;

case clDBExec.Table.konveer:

node.Cells.Add(new Cell(clTablica.OperationEnum.Транспортировка.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Транспортировка).ToString()));

break;

case clDBExec.Table.magnit:

node.Cells.Add(new Cell(clTablica.OperationEnum.Контроль.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Контроль).ToString()));

break;

case clDBExec.Table.peskosuwilka:

node.Cells.Add(new Cell(clTablica.OperationEnum.Сушка.ToString()));

node.Cells.Add(new Cell(((int)clTablica.OperationEnum.Сушка).ToString()));

break;

}

advTree2\_ТаблОпераций.Nodes.Add(node);

//Имя пишем в массив//Стол//Код берем из Char первых символов имяни

if(!string.IsNullOrEmpty(name))

\_clDBexec.НазванияОборудованияПоСтоимости.Add(string.Format("{1:000000}, {0}", name, (int)name[0] + (int)name[name.Length - 1]));

}

\_OP += 5;

}

/// <summary>

/// Клик меню по кнопке А/Б/О/Т

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void buttonX\_АБОТ\_Click(object sender, EventArgs e)

{

var name = ((ButtonX) sender).Text;

//Показать поля для заполнений

switch (name)

{

case "А":

superTabControl1\_АБОТ.SelectedTabIndex = 1;

break;

case "Б":

superTabControl1\_АБОТ.SelectedTabIndex = 2;

break;

case "О":

superTabControl1\_АБОТ.SelectedTabIndex = 3;

break;

case "Т":

superTabControl1\_АБОТ.SelectedTabIndex = 4;

break;

case "00":

\_clTablica.ПустаяСтрока(reoGridControl\_МаршрутнаяКарта);

break;

case ""://Таблица Операций

superTabControl1\_АБОТ.SelectedTabIndex = 0;

break;

case ""://Подтвердить

if(advTree2\_ТаблОпераций.Nodes.Count==0) break;

ДобавитьДанныеСФормы();

//Открыть Таблицу Операций

superTabControl1\_АБОТ.SelectedTabIndex = 0;

break;

}

}

//Клик по строке в Таблице Операций

private void advTree2\_ТаблОпераций\_NodeClick(object sender, TreeNodeMouseEventArgs e)

{

//Чистим все поля

ОчиститьВсеПоляВвода();

var nodeClick = e.Node;

//Заполнить строки //Опер.

textBoxX\_НОперации.Text = nodeClick.Cells[0].Text;

//Код, наименование операции

textBoxX\_КодОперации.Text = string.Format("{1}, {0}", nodeClick.Cells[1].Text, nodeClick.Cells[2].Text);

//Код, наименование оборудования

textBoxX\_Код.Text = \_clDBexec.НазванияОборудованияПоСтоимости[nodeClick.Index];

//Номер (код) цеха, в котором выполняется операция

textBoxX\_Цех.Text = string.Format("{0:00}", \_НомерЦеха);

//Номер (код) участка, конвеера, линии

textBoxX\_Участок.Text = string.Format("{0:00}", \_НомерЦеха >= 1 ? \_НомерЦеха - 1 : \_НомерЦеха);

//Номер (код) рабочего места

textBoxX\_РМесто.Text = string.Format("{0}{1:0}", \_НомерЦеха, (int)nodeClick.Cells[2].Text[0]);

//Коэфф штучного времени

textBoxX\_Кшт.Text = string.Format("{0:0.##}", КоэффШтучногоВремени(nodeClick.Index));

//Норма подготовительно-заключительного времени

textBoxX\_Тпз.Text = string.Format("{0:0.##}", НПодгельноЗаклогоВремНаОперю(nodeClick.Index));

//Заполнение поля Описание

var описание = "";

\_clTablica.OpisanieOperationEnum.TryGetValue(

(clTablica.OperationEnum) Enum.Parse(typeof (clTablica.OperationEnum), nodeClick.Cells[1].Text),

out описание);

textBoxX1\_O.Text = описание;

//Переключится на вкладку заполнения маршрутной карты//кнопка -А

buttonX\_АБОТ\_Click(new ButtonX() { Text = "А" }, e);

}

/// <summary>

/// Чистим все поля ввода АБОТ

/// </summary>

private void ОчиститьВсеПоляВвода()

{

foreach (var control in superTabControlPanel1\_А.Controls)

{

try{((TextBoxX)control).Text = "";}

catch (InvalidCastException){continue;}

}

foreach (var control in superTabControlPanel2\_Б.Controls)

{

try { ((TextBoxX)control).Text = ""; }

catch (InvalidCastException)

{

try{((DevComponents.Editors.IntegerInput)control).Text = "";}

catch (InvalidCastException){continue;}

}

}

textBoxX1\_O.Text = "";

textBoxX\_Т.Text = "";

label\_СтрокаСостояний\_А.Text =

label\_СтрокаСостояний\_Б.Text =

label\_CтрокаСостояний\_О.Text = label\_СтрокаСостояния\_Т.Text = "Строка состояний";

}

#region Вкладка А операций

fОхранаТруда \_fOxranaTruda = new fОхранаТруда();

//Выбрать из окна нужные документы охран труда

private void textBoxX6\_ДокОхранаТруда\_ButtonCustomClick(object sender, EventArgs e)

{

//Открыть окно с выбором из базы нормативных документов охраны труда

\_fOxranaTruda.ShowDialog();

var docResult = \_fOxranaTruda.\_chekedDoc;

textBoxX6\_ДокОхранаТруда.Text = "";

foreach (var dstr in docResult)

textBoxX6\_ДокОхранаТруда.Text += dstr;

}

/// <summary>

/// Кнопка ПОДТВЕРДИТЬ - вписать поля в Таблицу карты

/// </summary>

private void ДобавитьДанныеСФормы()

{

\_clTablica.АtrokaSet(new clTablica.A(textBoxX\_Цех.Text, textBoxX\_Участок.Text, textBoxX\_РМесто.Text

, textBoxX\_НОперации.Text, textBoxX\_КодОперации.Text,

textBoxX6\_ДокОхранаТруда.Text, reoGridControl\_МаршрутнаяКарта));

\_clTablica.БstrokaSet(new clTablica.Б(textBoxX\_Код.Text, textBoxX\_СМ.Text, textBoxX\_Проф.Text

, textBoxX\_Р.Text, textBoxX\_УТ.Text, integerInput\_\_КР.Text,

integerInput\_КОИД.Text, integerInput\_ЕН.Text,

integerInput\_ОП.Text, textBoxX\_Кшт.Text, textBoxX\_Тпз.Text,

textBoxX\_Тшт.Text, reoGridControl\_МаршрутнаяКарта));

if (!string.IsNullOrEmpty(textBoxX1\_O.Text)) \_clTablica.ОstrokaSet(new clTablica.О(textBoxX1\_O.Text, reoGridControl\_МаршрутнаяКарта));

if (!string.IsNullOrEmpty(textBoxX\_Т.Text)) \_clTablica.ТstrokaSet(new clTablica.Т(textBoxX\_Т.Text, reoGridControl\_МаршрутнаяКарта));

}

//Cтрока состояний заполняем тут

private void textBoxX6\_ДокОхранаТруда\_Click(object sender, EventArgs e){label\_СтрокаСостояний\_А.Text = ((DevComponents.DotNetBar.Controls.TextBoxX) sender).WatermarkText;}

#endregion

#region Вкладка Б операций

//Enter//Click

private void textBoxX\_Проф\_Click(object sender, EventArgs e)

{

VisibleUCAll();

switch (int.Parse(((DevComponents.DotNetBar.Controls.TextBoxX)sender).Tag.ToString()))

{

case 2: ucСтепеньМеханизац\_2.Visible = true; break;

case 3: ucКодПроф\_3.Visible = true; break;

case 4: ucРазрядРаботы1.Visible = true; break;

case 5: ucКодУсловийТруда1.Visible = true; break;

}

var ctrl = ((DevComponents.DotNetBar.Controls.TextBoxX)sender).WatermarkText;

label\_СтрокаСостояний\_Б.Text = ctrl;

}

private void textBoxX\_Проф\_Leave(object sender, EventArgs e)

{

switch (int.Parse(((DevComponents.DotNetBar.Controls.TextBoxX)sender).Tag.ToString()))

{

case 2: ucСтепеньМеханизац\_2.Visible = ucСтепеньМеханизац\_2.ContainsFocus; break;

case 3: ucКодПроф\_3.Visible = ucКодПроф\_3.ContainsFocus; break;

case 4: ucРазрядРаботы1.Visible = ucРазрядРаботы1.ContainsFocus; break;

case 5: ucКодУсловийТруда1.Visible = ucКодУсловийТруда1.ContainsFocus; break;

}

}

//Погасить все панели СПРАВО

private void VisibleUCAll()

{

foreach (Control cntrl in panel1\_ДляНастроекСправо.Controls)

cntrl.Visible = false;

}

//Выбрал Код Професии

private void ucКодПроф1\_ИзмененияВЛинке(object sender, EventArgs e){textBoxX\_Проф.Text = ucКодПроф\_3.КодПроф;}

//Выбрал Код Механизации

private void ucСтепеньМеханизац1\_КодСтепеньМеханизацииEvent(object sender, EventArgs e){textBoxX\_СМ.Text = ucСтепеньМеханизац\_2.КодСтепеньМеханизации.ToString();}

//Выбрал Разряд Работ

private void ucРазрядРаботы1\_РазрядРабот\_Event(object sender, EventArgs e){textBoxX\_Р.Text = ucРазрядРаботы1.РазрядРабот.ToString();}

//Выбрал Код Условий труда

private void ucКодУсловийТруда1\_КодУсловийТруда\_Event(object sender, EventArgs e){textBoxX\_УТ.Text = ucКодУсловийТруда1.КодУсловийТруда.ToString();}

private void integerInput\_ОП\_Click(object sender, EventArgs e)

{

try

{

var ctrl = ((DevComponents.DotNetBar.Controls.TextBoxX)sender).WatermarkText;

label\_СтрокаСостояний\_Б.Text = ctrl;

}

catch (InvalidCastException)

{

var ctrl = ((DevComponents.Editors.IntegerInput)sender).WatermarkText;

label\_СтрокаСостояний\_Б.Text = ctrl;

}

}

/// <summary>

/// Рассчитать Кшт=[0...1]

/// </summary>

/// <returns></returns>

private double КоэффШтучногоВремени(int index)

{

var rnd = new Random();

var Тшт = 0.0;

//Кшт = Тшт\(Тшт+Тшт+..)

var result = 0.0;

//Если индекс есть в \_Тшт[i] то берем его, иначе сгенерить новое и записать \_Тшт.Add(Тшт);

if (index < \_Тшт.Count)

Тшт = \_Тшт[index];

else

for (int i = 0; i < advTree2\_ТаблОпераций.Nodes.Count; i++)

{

//от 0 до 10

Тшт = rnd.NextDouble() \* (10.0 - 0.1) + 0.1;

\_Тшт.Add(Тшт);

}

textBoxX\_Тшт.Text = string.Format("{0:0.##}", Тшт);

var SumТшт = 10.75;

result = Тшт/SumТшт;

return result;

}

List<double> \_Тшт = new List<double>();

List<double> \_Тпз = new List<double>();

/// <summary>

/// Расчет Тпз=[0...100]

/// </summary>

/// <returns></returns>

private double НПодгельноЗаклогоВремНаОперю(int index)

{

var rnd = new Random();

//Тпз = Тпз1 + Тпз2 + Тпз3

var Тпз = 0.0;

if (index < \_Тпз.Count)

Тпз = \_Тпз[index];

else

for (int i = 0; i < advTree2\_ТаблОпераций.Nodes.Count; i++)

{

//от 0 до 100

Тпз = rnd.NextDouble() \* (100.0 - 0.1) + 0.1;

\_Тпз.Add(Тпз);

}

return Тпз;

}

#endregion

#region Вкладка О операций

//Строка состояния

private void textBoxX1\_O\_Click(object sender, EventArgs e)

{

label\_CтрокаСостояний\_О.Text = ((DevComponents.DotNetBar.Controls.TextBoxX) sender).WatermarkText;

}

#endregion

#region Вкладка Т операций

//Строка состояния

private void textBoxX\_Т\_Click(object sender, EventArgs e)

{

label\_СтрокаСостояния\_Т.Text = ((DevComponents.DotNetBar.Controls.TextBoxX) sender).WatermarkText;

}

#endregion

}

}