

**Паспорт технологического решения в рамках**

**кейс – чемпионата PAPER CUP**

|  |  |
| --- | --- |
| **Название кейса** | Разработка программы для планирования участка резки |
| **Название команды** | Re::Code⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ |
| **ФИО и контактные данные**  **Наставника/ лидера команды** | Ивутин Алексей Николаевич⠀⠀⠀  +7-963-227-11-88⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀  alexey.ivutin@yandex.ru⠀⠀⠀⠀⠀⠀⠀ |
| **Образовательная организация** | Тульского государственного университета (ТулГУ) ⠀⠀⠀⠀⠀⠀⠀ |
| **Название Проектной идеи** | KartCut⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀⠀ |

Тула 2024

1. **Аннотация проекта**

Цель: снижение простоев навивных станков, вызванных отсутствием подготовленного нарезанного картона. Достижение этой цели позволит повысить общую эффективность производственного процесса и создать более устойчивую операционную среду.

Задачи:

1) разработка программы очередности резки картона (создание алгоритма, который будет учитывать специфику времени, необходимого для резки картона, а также приоритеты в очередности выполнения заказов). Это позволит оптимально настраивать график нарезки материала в зависимости от его актуальности и сроков исполнения заказов.

2) анализ данных (сбор и обработка данных о времени резки, производственных возможностях и текущих заказах). Важно учесть статистические данные о прошлом поведении процессов, а также предсказать потребности на основе исторической информации.

Ожидаемые результаты: снижение простоев навивных станков, вызванных ожиданием нарезки картона, на 10%. Это приведет к более эффективному использованию оборудования и ресурсов, а также повысит общую производительность предприятия.

Область применения: участок резки, где производится нарезка картона для последующего использования на навивных станках. Улучшение организации работы данного участка поможет существенно сократить время ожидания и повысит качество конечного продукта.

Потенциальные потребительские сегменты: производственные предприятия, занимающиеся изготовлением изделий из картона, включая упаковку и бумажную продукцию.

1. **Описание проблемы**

Проблема:

- в связи с отсутствием автоматизированной системы для решения данной задачи планирование резки картона может приводить к простоям технологического оборудования, что приводит к снижению эффективности производства.

1. **Технология, лежащая в основе продукта**

В основе разработки данного продукта лежит комплексный процесс анализа сменного задания на трех участках навивки, который служит основой для формирования сменного задания для участка резки. Основной целью данного подхода является минимизация простоев на навивных станках и предотвращение образования излишнего количества картона, вызванного недостоверной оценкой объема роликов.

Кроме этого, продукт включает несколько ключевых аспектов, которые способствуют повышению его эффективности:

* учет остаточных объемов картона для каждого навивного станка. Это позволяет оптимизировать использование сырья и своевременно реагировать на дефицит материала;
* закладывание в сменное задание необходимости подготовки запасов картона для обеспечения бесперебойной работы в следующей смене. Это способствует повышению производительности и снижению риска простоя;
* на каждом станке установлено максимально допустимое количество размещения за одну завесу. Это ограничение необходимо для эффективного распределения нагрузки и оптимизации производственного процесса;
* ширина полос картона может варьироваться в зависимости от диаметра изготавливаемой шпули. Это позволяет гибко адаптировать процесс резки в соответствии с конкретными требованиями к продукции;
* каждый типоразмер полос обрабатывается с определенной скоростью, что также учитывается в алгоритме управления. Это позволяет оптимизировать производственный цикл и повысить качество конечной продукции.
* деление сырья на классы: эконом, стандарт, стандарт усиленный и экстра. Эта классификация позволяет более точно управлять процессами, а также оптимизировать затраты на материалы в зависимости от требований к производству.

1. **Научно-техническое решение**

В рамках данного научно-технического решения рассматривается оптимизация процесса навивки на специализированных машинах, что включает в себя различные аспекты управления производственными ресурсами и повышения эффективности процессов. Способствующая данному улучшению система базируется на определенных классах и алгоритмах, позволяющих управлять данными и оптимизировать производственный цикл:

1. машина навивки представляется в качестве класса, содержащего следующие поля (в соответствии с пунктом 7.3):

* индекс (номер машины): уникальный идентификатор, присваиваемый каждой машине для ее дальнейшего отслеживания и управления;
* список задач: представлен в виде массива данных (класс «*Изделие*»), который фиксирует все текущие задачи, поставленные перед машиной навивки. Это позволяет систематизировать рабочие процессы и упрощает планирование;
* список процесса: представлен в виде массива данных (класс «*Изделие*»), содержащий информацию о тех изделиях, которые уже находятся в стадии обработки. Данный список необходим для отслеживания статуса выполнения заказов и управления последовательностью работы;
* резерв машины: представлен в виде массива данных (класс «*Картон*»), который охватывает наличные запасы картона, доступные для обработки. Это предоставляет возможность оперативно восполнять запасы и управлять ресурсами более эффективно.

1. первоочередное определение общего количества необходимых роликов для резки: важным элементом является начальная оценка количества роликов, необходимых для резки. Это позволяет минимизировать отходы и оптимизировать ресурсное обеспечение;
2. переход от веса изделий к количеству: Исходя из текущих требований (в соответствии с пунктом 7.1 «Определение количества изделий исходя из их веса»), предусмотрен переход от измерения в весовых единицах к количествам изделий. Это изменение упрощает процесс учета и управления запасами;
3. алгоритм индексации «важности» ширины резки картона: в рамках данного решения используется алгоритм, который позволяет определять порядок вычисления количества роликов на смену (в соответствии с пунктом 7.1 «Определение индекса ширины полосы»). Это обеспечивает большую точность в настройках и улучшает качество конечного продукта;
4. использование алгоритма поэтапной обработки сменного задания: этап состоит из двух частей: определение станка навивки с наименьшим временем сработки и определение следующего изделия у данного станка (в соответствии с пунктом 7.2);
5. использование дополнительного условного резерва: картон не использованный для изготовления изделия «запоминается» программой и ждет своего последующего использования, а не заполняет один из станков;
6. приоритетное использование резервов: если присутствует возможность взять картон из резервов (как станка так и условного), то данный вариант имеет более высокий приоритет по сравнению с резкой;
7. контроль времени работы: в рамках решения предусмотрено внедрение механизма контроля за временем работы, что содействует повышению общей производительности и управлению временными ресурсами;
8. разделение обработки сметы: обработка сметы должна делиться на несколько категорий, включая сегодняшнюю, завтрашнюю, 150-х полос и с дополнительного производства, что помогает лучше отслеживать производственные расходы и планировать будущие затраты.

Данное научно-техническое решение имеет за цель экономически обоснованное и эффективное управление производственными процессами, что, в свою очередь, способствует повышению общей производительности и конкурентоспособности предприятия.

1. **Существующие альтернативы**

В современных условиях задача планирования участка резки осуществляется путем приблизительного определения последовательности операций, исходя из опыта мастера.

1. **Обоснование реализуемости идеи**

В условиях современного рынка, характеризующегося высокой конкурентоспособностью и динамичными изменениями до спроса, обоснование реализуемости предлагаемых научно-технических решений требует комплексного анализа различных факторов, которые могут быть определены как конкурентные преимущества. Данные преимущества служат основой для успешной реализации идеи и исходят из уникальных ресурсов и возможностей, которые имеются у нашей компании.

Конкурентные преимущества:

1. Уникальные решения и технологии:

- Использование алгоритмов и патентованных методик для оптимизации процессов резки.

- Внедрение интеллектуальных систем планирования и автоматизации.

2. Дефицит:

- На рынке ощущается нехватка интегрированных решений, способных обеспечить бесперебойную работу производственных линий.

- Существующие системы управления зачастую не учитывают специфику конкретных производств, что приводит к непредвиденным простоям.

3. Уникальность:

- Проект предлагает комплексное решение, включающее не только технологическую оптимизацию, но и полную интеграцию процессов с учетом специфики предприятия.

- Уникальные разработанные алгоритмы и системы позволяют достичь наивысшей точности и качества резки, что существенно улучшает конечный продукт.

1. **Основные технические параметры**
2. **Математическое обеспечение**

* Определение количества изделий исходя из их веса:

, (1)

где:

W – вес изделий;

w – ширина ролика;

T – ширина одной полосы;

c – количество полос, требуемое для 1 изделия.

* Определение индекса ширины полосы:

Представим результат резки ролика на полосы в виде таблицы:

Таблица 1 – Результат резки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 70 | 100 | 150 | 166 |
| 70 | N(70,70) | N(70,100) | N(70,150) | N(70,166) |
| 100 | N(100,70) | N(100,100) | N(100,150) | N(100,166) |
| 150 | N(150,70) | N(150,100) | N(150,150) | N(150,166) |
| 166 | N(166,70) | N(166,100) | N(166,150) | N(166,166) |

Производится суммирование строк таким образом что, если N(\_, \_) имеет значение отличное от 0, то данной ширине увеличивается индекс на 1.

Таблица 2 – Пример

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 70 | 100 | 150 | 166 |
| 70 | 7 | 3 | 0 | 0 |
| 100 | 0 | 8 | 0 | 0 |
| 150 | 0 | 0 | 7 | 0 |
| 166 | 0 | 1 | 0 | 4 |

В соответствии с таблицей 2, получим, что *70* имеет индекс – 2, *100* имеет индекс – 1, *150* имеет индекс – 1, *166* имеет индекс – 2.

Отсортировав ширины по индексу получим следующий порядок резки: 100 → 150 → 70 → 166.

Применяется данный способ в связи с возможностью получения дополнительных полос совсем других ширин при резки, из-за чего при линейном определении количества роликов может возникать остаток картона.

1. **Алгоритмическое обеспечение**

* Метод “*machine\_create*”

1. Считать data\_, width\_roller;

2. keys = [], index = 0, machine = 0, step = “”;

3. Переберать элементы data\_ и добавить ключи в список keys.

4. Создать список machines, содержащий три экземпляра data.Machine;

5. Переберать элементы в первом наборе данных (по ключу keys[0]).

- для каждого элемента увеличить счетчик machine.

- установить переменную step равной keys[0].

- инициализировать переменную index.

6. Обработать каждое изделие (workpiece)

- для каждого workpiece в значении текущего элемента:

- увеличить счетчик index.

- преобразовать третий элемент workpiece в целое число и сохраните в переменной \_type\_.

- проверить, находится ли \_type\_ в заданном множестве (70, 100, 166).

- проверить, что первый элемент workpiece присутствует в data.thickness\_count.

- получить количество \_count\_ из data.thickness\_count.

- проверить, является ли первый элемент workpiece допустимым классом (можно использовать список допустимых значений).

- преобразовать вторую часть workpiece в float, заменяя запятую на точку, и сохранить в \_thickness\_.

- выполнить математические расчеты в соответствии с формулой 1, в результате которых создать экземпляры data.Workpiece и добавить их в список estimate\_workpiece соответствующей машины.

7. Обработка данных второго ключа

- сбросить счетчик machine на 0.

- переберать элементы во втором наборе данных (по ключу keys[1]).

- для каждого элемента снова увеличить счетчик machine.

- установить step равным keys[1].

- инициализируйте переменную index.

- обработать каждый элемент (card) в значении текущего элемента:

- увеличить index.

- создать экземпляр data.Cardboard и добавьте его в список reserve соответствующей машины.

8. Вернуть список machines.

* Метод «*rollers*»

1. Создать два объекта cr и cr\_out класса data.Reels().
2. Для каждой машины в списке machines подсчитать количество роликов и обновить соответствующие значения в cr.
3. Пройти в цикле по отсортированным значениям индекса (сортировка производится в порядке возрастания значений второго элемента кортежа).
4. Для каждого ключа выполнить SQL-запрос для получения данных о количестве роликов разных размеров (count\_70, count\_100, и так далее) для конкретного типа роликов и ширины.
5. Рассчитать необходимое количество стандартных и эконом роликов для каждой категории роликов с округлением до целого числа в большую сторону.
6. Обновить значения в cr\_out с учетом подсчитанного количества стандартных и эконом роликов.
7. Обновить значения в cr, вычитая из текущих значений рассчитанные количества в зависимости от полученных данных запросом (count\_reel).
8. Вернуть объект cr\_out, содержащий финальное количество роликов для стандартной и эконом после всех расчетов.

* Метод «*winding*»

1. Инициализировать переменные: резерв материалов (reserve\_w), общее время работы (time), а также объект roller, который управляет количеством роликов различных типов.
2. Создать два выходных списка: один для резки (out\_cut), другой для намотки (out\_wind).
3. Выполнять пока у машин есть время работы, которое не равно максимальному значению (сравнение с sys.maxsize).
4. Определить машину с наименьшим временем до завершения текущей работы (с использованием функции min\_time).
5. Проверить наличие нужного картона:

- если для работы машины есть нужный картон, проверить его наличие как в резерве самой машины (machine.reserve), так и в общем резерве (reserve\_w).

- если картон есть, корректировать его количество в резервах, и машина переходит к намотке. Карточку изделия переместить из списка ожидаемых изделий (estimate\_workpiece) в список произведенных (production\_workpiece).

1. Добавить информацию о произведенном изделии в список намотки (out\_wind).
2. Если нужного картона нет или его недостаточно, то:

- из базы данных запросить информацию о доступных катушках и выполнить расчет необходимого количества роликов.

- корректировать запасы роликов (для эконом или стандартных видов картона).

- рассчитать время резки и, при необходимости, время переналадки станка, если используется ролик другого типа.

- обновлять время, исходя из количества роликов и времени их резки, а также времени переналадки (если тип ролика сменился).

1. Если время работы машины превышает время первой детали в производственном списке, то эта деталь нужно удалить, а время машины обновить.
2. Если были использованы катушки, обновить их остатки в резервах, и резерв перезаписать с новыми данными.
3. В конце функции возвратить списки резки и намотки, а также обновленный резерв.

* Метод «*cut\_150*»

1. Преобразовать входной список картонов с помощью функции card\_create, которая создает объекты с информацией о каждом картоне.
2. Инициализировать пустой список out для хранения результатов резки.
3. Обработать картон:

- для каждого картона из списка:

- проверить класс картона:

- если класс — «standart» (стандартный):

- использовать список катушек reels.standart для работы.

1. Если класс — другой (например, эконом):

- использовать список катушек reels.econom для работы.

1. Для каждой катушки (по каждой паре ключ-значение в словарях катушек):

- выполнить SQL-запрос, чтобы получить информацию о количестве материала на катушке для резки шириной 150 мм.

- извлечь результат из запроса (значение материала для катушки).

- умножить это значение на количество катушек данного типа.

- вычесть полученное значение из текущего количества картона.

1. После обработки каждой карточки картона добавить её в список out.
2. Вернуть список out, который содержит обновленные объекты картона с учетом выполненной резки.

* Метод «*reserve*»

1. Создать список из трёх новых объектов машин machines, где каждая машина инициализируется с пустыми списками для заготовок и резервов.
2. В цикле для каждой из трёх машин:
3. Скопировать резерв машин из списка machinesO в соответствующую машину из списка machinesN.
4. Для каждой машины в списке machinesN:

- для каждой заготовки (workpiece) из списка estimate\_workpiece машины:

- выполнить SQL-запрос для получения максимального размера резерва для данной машины.

1. Если нужный картон для заготовки есть в общем резерве res\_w и текущий резерв машины меньше допустимого размера:
2. Если картон, соответствующий заготовке, уже есть в резерве машины:

- увеличить количество картона в резерве машины на количество из общего резерва res\_w.

1. Если картона в резерве машины нет:

- переместить картон из общего резерва res\_w в резерв машины.

1. Для каждой машины:
2. Если минимальное время выполнения работы среди всех машин не достигло максимального значения:

- выполнить SQL-запрос для получения максимального размера резерва для данной машины.

1. Для каждой завершённой заготовки из списка, возвращаемого функцией winding:
2. Если заготовка была завершена на текущей машине:
3. Если резерв машины не превышает допустимый размер:
4. Если соответствующий картон уже есть в резерве машины:

- увеличить количество картона на значение завершённой заготовки.

1. Если такого картона нет:

- добавить картон из завершённой заготовки в резерв.

1. Если размер резерва превышен, прекратить добавление заготовок.
2. Для каждой машины:

- добавить текущий картон из резерва машины в список estimate\_workpiece каждой машины.

1. Вернуть обновлённый список машин machines, каждая из которых содержит заготовки, готовые для дальнейшей обработки.
2. **Диаграмма классов**

Диаграмма классов представлена в соответствии с рисунком 1.

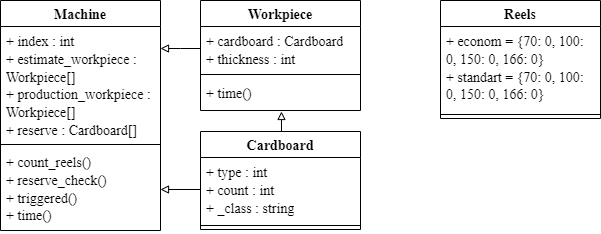


Рисунок 1 – диаграмма классов

1. **Оценка требований к вычислительным ресурсам (база данных)**

Первым этапом в оценке требований идёт определение примерного объёма данных.

Данные вычисления можно произвести в соответствии с формулой (2).

, (2)

где:

li- длина записи в i-й таблице (в байтах);

Ni - максимально возможное количество записей в i-й таблице;

Na - количество записей в архиве i-й таблицы.

1. Таблица “machine”.

Длина записи вычисляется путём суммирования веса типов столбцов, которые присутствуют в данной таблице.

В таблице присутствует 2 столбца с типом integer.

Вес данных типов в PSQL:

Integer – 4 байта.

Исходя из вышеперечисленных данных, можно посчитать длину записи, которая будет равна – 8 байт.

Следующим этапом идёт вычисление максимального количества записей в таблице.

В данный момент на производстве присутствует 3 навивных станка, следовательно и записей будет 3.

В связи с тем, что архив для данной таблицы не предусматриваем его количество примем за 0.

Подставив конечные данные в формулу, получим:

что таблица с водителями займёт 24 байт памяти.

1. Таблица winding.

В таблице присутствует 4 столбца с типами: integer (3 столбца), text.

Вес данных типов в PSQL:

Integer – 4 байта;

Text – 65536 байт.

Исходя из вышеперечисленных данных, можно посчитать длину записи, которая будет равна:

Возможных комбинаций класса, ширины полосы и толщины изделия всего 16, следовательно и записей будет 16.

В связи с тем, что архив для данной таблицы не предусматриваем его количество примем за 0.

Подставив конечные данные в формулу, получим, что таблица займёт байт памяти.

1. Таблица cutting.

В таблице присутствует 6 столбцов с типом integer.

Вес данных типов в PSQL:

Integer – 4 байта.

Исходя из вышеперечисленных данных, можно посчитать длину записи, которая будет равна

Возможных комбинаций ширины полосы и ширины ролика всего 8, следовательно и записей будет 8.

В связи с тем, что архив для данной таблицы не предусматриваем его количество примем за 0.

Подставив конечные данные в формулу, получим, что таблица займёт:

1. Таблица cutting\_time.

В таблице присутствует 3 столбца с типами: integer, text, double.

Вес данных типов в PSQL:

Integer – 4 байта;

Double – 17 байт;

Text – 65536 байт.

Исходя из вышеперечисленных данных, можно посчитать длину записи, которая будет равна

Возможных комбинаций ширины полосы и класса всего 8, следовательно и записей будет 8.

В связи с тем, что архив для данной таблицы не предусматриваем его количество примем за 0.

Подставив конечные данные в формулу, получим, что таблица займёт:

1. Таблица retooling.

В таблице присутствует 2 столбца с типами: integer, double.

Вес данных типов в PSQL:

Integer – 4 байта;

Double – 17 байт.

Исходя из вышеперечисленных данных, можно посчитать длину записи, которая будет равна

Всего ширин полос 4, следовательно и записей будет 4.

В связи с тем, что архив для данной таблицы не предусматриваем его количество примем за 0.

Подставив конечные данные в формулу, получим, что таблица займёт:

В конечном итоге, вся БД займёт в памяти:

Динамика роста объёма данных.

Рост данных, в случае с автопарком будет напрямую зависеть от двух факторов: роста населения города и роста самого города.

1. **Аналитическое решение**

Возьмем следующее сменное задание:

Таблица 3 – Сменное задание для станка 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Изделия | Класс | Толщина | Ширина полосы | Вес |
| Шпуля 1 | Стандарт | 3.5 | 70 | 200 |
| Шпуля 2 | Стандарт | 4 | 70 | 240 |

Таблица 4 – Сменное задание для станка 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Изделия | Класс | Толщина | Ширина полосы | Вес |
| Шпуля 3 | Стандарт | 7 | 100 | 628.57 |
| Шпуля 5 | Эконом | 8 | 701 | 320 |

Таблица 5 – Сменное задание для станка 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № Изделия | Класс | Толщина | Ширина полосы | Вес |
| Шпуля 5 | Стандарт | 5 | 166 | 758.85 |

В качестве ширины ролика для резки примем 1050, со следующими показателями:

Таблица 6 – Результат резки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ширина полосы | Кол-во 70 полос | Кол-во 100 полос | Кол-во 150 полос | Кол-во 166 полос |
| 70 | 12 | 0 | 1 | 0 |
| 100 | 0 | 10 | 0 | 0 |
| 150 | 0 | 0 | 7 | 0 |
| 166 | 1 | 0 | 1 | 5 |

Рассчитаем количество роликов исходя из следующей формулы:

, (3)

где:

W – вес изделий;

w – ширина ролика;

T – ширина одной полосы;

Количество картона требуемого для первой шпули:

Количество картона требуемого для второй шпули:

Количество картона требуемого для третьей шпули:

Количество картона требуемого для четвертой шпули:

Количество картона требуемого для пятой шпули:

Итого, для данного заказа требуется 19 полос 70 ширины класса стандарт, 11 полос 150 ширины класса эконом и 8 полос 166 ширины класса Стандарт.

Аналитически рассмотрим работу станка:

1. Определим станок с наименьшим временем сматывания (в связи с тем, что резерв мы не рассматриваем, выбираем первый) – 1.
2. В связи с тем, что первой позицией стоит шпуля из 70 полос класса стандарт, режем ролик на 150 полосы, в соответствии с рисунком 2.

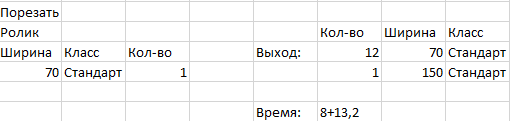


Рисунок 2 – Этап резки

1. В результате получаем следующую ситуацию на станках сматывания, в соответствии с рисунком 3.

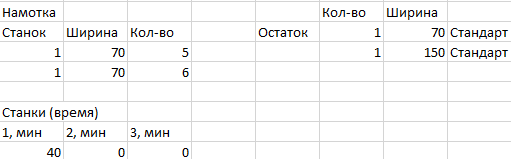


Рисунок 3 – Этап навивки

1. Снова определяем станок с наименьшим временем сматывания и прогоняем предыдущие шаги, в соответствии с рисунком 4.

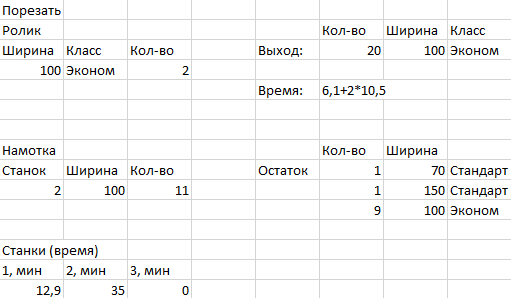


Рисунок 4 – Обработка второго станка

1. Снова определяем станок с наименьшим временем сматывания и прогоняем предыдущие шаги, в соответствии с рисунком 5.

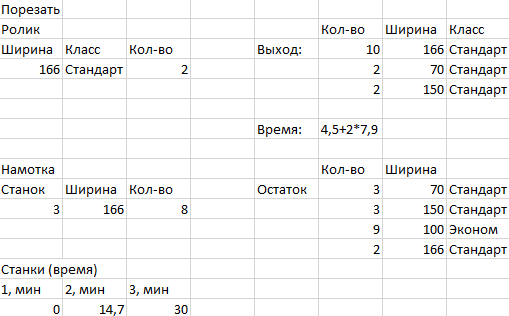


Рисунок 5 – Обработка третьего станка

1. Снова определяем станок с наименьшим временем сматывания и прогоняем предыдущие шаги, в соответствии с рисунком 6.

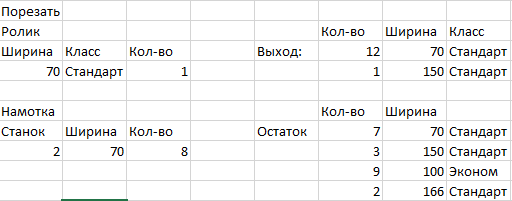


Рисунок 6 – Обработка первого станка

Таким образом, смета на станок резки будет выглядеть следующим образом, в соответствии с рисунком 7.

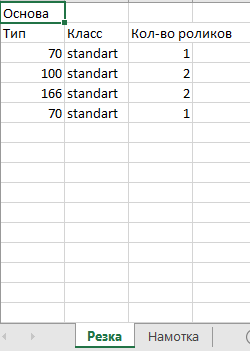


Рисунок 7 – Задание на резку

1. **Тестирование**

Проведем тестирование программы и сравним результат с аналитическим решением, в соответствии с рисунками 8-13.

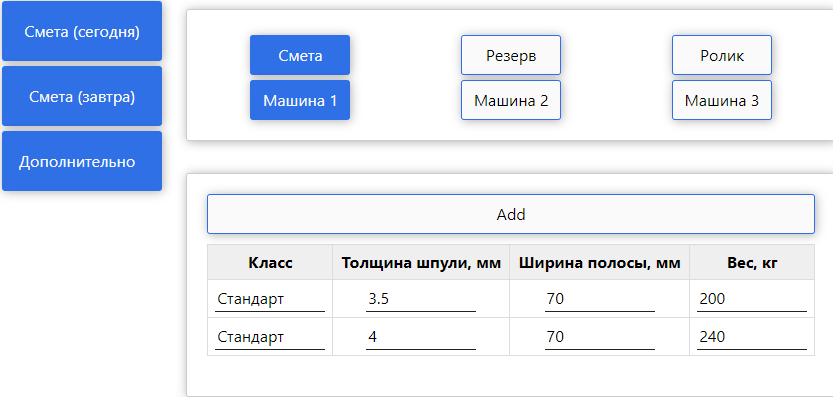


Рисунок 8 – Станок навивки №1

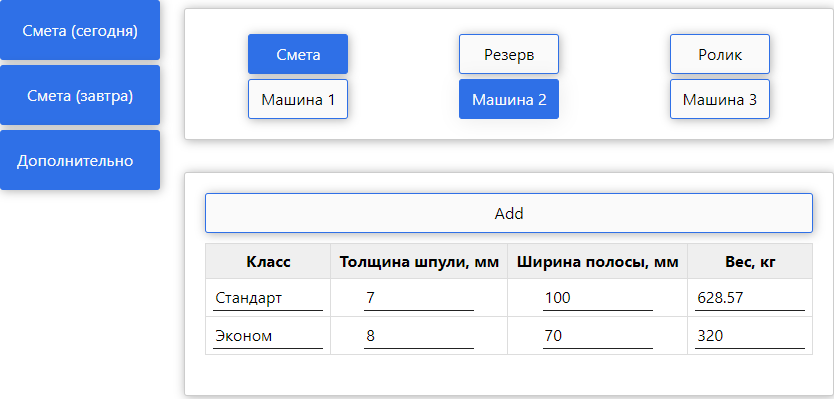


Рисунок 9 – Станок навивки №2

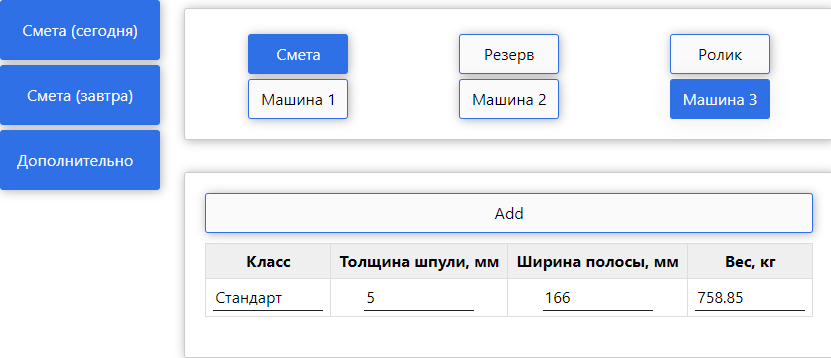


Рисунок 10 – Станок навивки №3



Рисунок 11 – Подсчет количества роликов

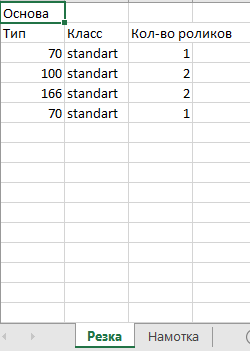


Рисунок 12 – Задание на резку

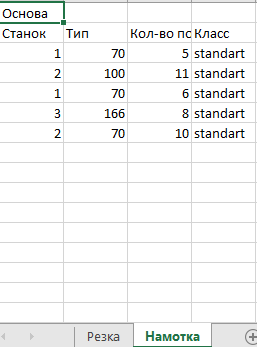


Рисунок 13 – Задание на навивку

Результат аналитического решения и работы программы совпал, на основании чего можно сделать вывод о том, что программа работает корректно.