## ЗАДАНИЕ 2021\_А2. Цех

**Матвеев, Перехрест**

Имеется конвертерный цех (рис. 1). В цех с интенсивностью K подается жидкий чугун с помощью передвижного миксера емкостью 7600 тонн. Далее чугун поступает в один из трех конвертеров (конвертерных печей). Доля чугуна для плавки составляет 75% от объема конвертерных печей. В конвертере происходит выплавака стали; далее разливной ковш по очереди принимает сталь из конвертеров и перевозит ее в устройства непрерывного разлива стали (УНРС). Сталь находится в конвертере до тех пор, пока не освободиться хотя бы одно устройство УНРС. Устройства непрерывного разлива стали формируют на выходе единицы слябов (1 единица за один период работы устройства).

Необходимо определить оптимальную интенсивность поступления чугуна K, при которой достигаются: 1) максимальные загрузки агрегатов цеха (конвертеров, разливного ковша, агрегатов УНРС); 2) максимальная производительность цеха, определяемая в количестве произведенных единиц сляба за определенный период времени T. Рассмотреть краткосрочное планирование (T=1 неделя) и долгосрочное планирование (T=3 месяца).

На рис. 1 для каждой операции указано среднее время выполнения (вверху прямоугольника, в часах:минутах).

***00:05±00:02***

Подготовка конвертера к плавке

***00:05±00:02***

Подготовка конвертера к плавке

***K***

Подача чугуна

**7600 т**

***00:05±00:02***

Подготовка конвертера к плавке

***00:05±00:03***

Слив стали из конвертера в разливной ковш

**До 310 т**

***00:04±00:01***

Передвижен ие ковша со сталью

***01:11±00:10***

УНРС 5

***01:11±00:10***

УНРС 4

***01:11±00:10***

УНРС 3

***01:11±00:10***

УНРС 2

***01:11±00:10***

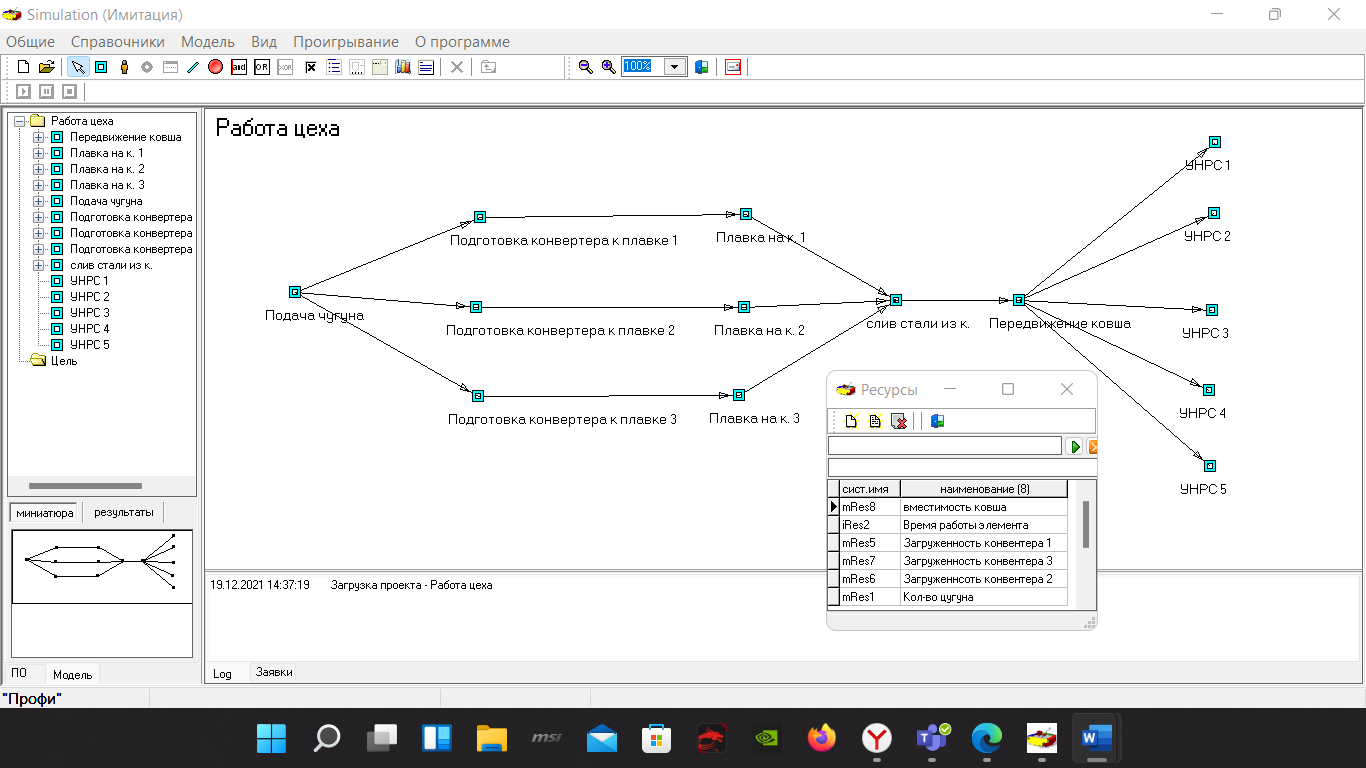
УНРС 1

|  |  |
| --- | --- |
| ***00:37±00:05***  Плавка на конвертере 1  **280 т** |  |
|  |
|  | |
| ***00:34±00:05***  Плавка на конвертере 2  **250 т** |  |
|  |
|  | |
| ***00:43±00:05***  Плавка на конвертере 3  **310 т** |  |
|  |

Рис. 1. Работа конвертерного цеха

**Ход работы**

Была идея воссоздать модель в среде BPsim.mas, но после двухчасового моделирования ничего особо и не получилось. Столкнулись с несколькими проблемами, в том числе и в отсутствии опыта. Поэтому было решение прийти к решению пошаговым разбором задачи и выявлением закономерностей с помощью логики.



Я, Александр Перехрест, написал программу, которая симулирует работу цеха и вычисляет время на получение энного количества слябов из 7600 т чугуна. Полное решение доступно по ссылке <https://github.com/SashaPerekhrest/labaCeh>. Я не стал разбивать программу на кодовые файлы, так что весь активный код лежит в файле Program.

Изображение выглядит как текст

Автоматически созданное описание

В начале программы мы видим объявление переменных, какие за что отвечают подписано.  
Так же в программе определенно 2 класса: PlavkaKon и UNRS. Первый является описанием элемента «конвертер», второй же описанием UNRS.

public class PlavkaKon  
{  
 public int countChug; *// количество чугуна* public int timeWork; *// время работы плавления* private Timer aTimer; *// таймер* public bool isReady; *// готовность к принятию новой партиии* public PlavkaKon(int timeWork, int countChug)  
 {  
 this.countChug = countChug;  
 this.timeWork = timeWork;  
 isReady = true;  
 }  
  
 public int Work()  
 {  
 SetTimer();  
 return countChug;  
 }  
  
 private void SetTimer()  
 {  
 aTimer = new Timer(timeWork);  
 aTimer.Elapsed += OnTimerEvent;  
 aTimer.AutoReset = false;  
 aTimer.Enabled = true;  
 }  
  
 private void OnTimerEvent(object sender, ElapsedEventArgs e)  
 {  
 isReady = true;  
 }  
}

В любой момент мы можем вызвать метод Work, который отработает нужное время и в конце вернет количества чугуна, которое было переплавлено.

public class UNRS  
{  
 public int timeWork; *// время работы* private Timer aTimer; *// таймер* public bool isReady; *// готовность принять новую партию* public UNRS(int timeWork)  
 {  
 this.timeWork = timeWork;  
 isReady = true;  
 }  
   
 public int Work()  
 {  
 SetTimer();  
 return 1;  
 }  
  
 private void SetTimer()  
 {  
 aTimer = new Timer(timeWork);  
 aTimer.Elapsed += OnTimerEvent;  
 aTimer.AutoReset = false;  
 aTimer.Enabled = true;  
 }  
  
 private void OnTimerEvent(object sender, ElapsedEventArgs e)  
 {  
 isReady = true;  
 }  
}

Здесь также мы можем вызвать в любой момент метод Work, который по истечении времени работы вернет нам один сляб.

Так же в начале программы мы выставляем значения нужным переменным.

Изображение выглядит как текст

Автоматически созданное описание

Количество чугуна берем 7600 (как в схеме), 0 слябов, ковш готов, но не отработал, УНРС готовы, рандомайзер для будущих вычислений.

Небольшое пояснение к следующему материалу. В схеме указано время в часах и минутах. Я перевел все время в минуту и , для быстроты работы программы, записал минуты как миллисекунды и дописал еще 0 для исключения погрешности. То есть, 1 час 11 минут превратился в 71 минуту, затем считаем что это 71 миллисекунда, добавляем 0 и записываем в программу 710.

Так же создаем массивы с конвекторами и УНРС.

Изображение выглядит как текст, стена, монитор, экран

Автоматически созданное описание

После чего запускаем таймер.

Изображение выглядит как текст

Автоматически созданное описание

Первым делом загружаем все конвекторы чугуном.

Изображение выглядит как текст

Автоматически созданное описание

После чего, пока весь чугун не кончится, повторяем действия:

Изображение выглядит как текст

Автоматически созданное описание

Проходимся по каждому конвектору и смотрим, готов ли он к загрузке и свободно ли место в ковше. Если да, то загружаем партию в ковш и новую партию в конвектор. Перед тем, как загрузить новую партию в ковш, проверяем, отработал ли он. Если нет, значит он пустой (почему так дальше пояснение) и можно загрузить новую партию. После чего еще раз проверяем, отработал ли ковш. Если да, загружаем в УРНС.

Обработчик работы ковша.

Изображение выглядит как текст

Автоматически созданное описание

По истечению времени мы говорим, что ковш выполнил работу.

Обработчик работы всех УРНС

Изображение выглядит как текст

Автоматически созданное описание

Проверяем все УРНС. Если находим 1 отработавшую, заполняем ее, переводим ковш из состояния «отработал» в состояние «готов принять новую партию», запускаем работу текущего УРНС и по исходу работы получаем 1 сляб.

После выполнения всех вышеперечисленных действий, считаем общее время работы и выводим общее время и количество слябов в консоль.

Изображение выглядит как текст

Автоматически созданное описание