**Практическая работа № 6**

**Тема: Структурное и сетевое планирование и управление**

**Цель: Научиться составлять сетевой график, рассчитывать критический путь и строить календарный график работы.**

Менеджер проекта, занимаясь его планированием, должен помнить о проектном треугольнике ограничений: «продолжительность» – «стоимость» – «содержание». Ресурсные и стоимостные ограничения предопределяют качество расписания проекта. Сетевое планирование, хотя и является достаточно рутинным инструментом календарной проработки, тем не менее, позволяет лучшим образом выполнить оптимизацию плана в отношении ресурсов и сроков. Сетевой график, построенный по методу «вершина – работа», предоставляет все возможности применения прикладных методов оптимизации.

**Сетевой график** – это ориентированный граф, в котором вершинами обозначены работы проекта, а дугами – временные взаимосвязи работ.

Сетевой график должен удовлетворять следующим **свойствам**.

1. Каждой работе соответствует одна и только одна вершина. Ни одна работа не может быть представлена на сетевом графике дважды. Однако любую работу можно разбить на несколько отдельных работ, каждой из которых будет соответствовать отдельная вершина графика.
2. Ни одна работа не может быть начата до того, как закончатся все непосредственно предшествующие ей работы. То есть если в некоторую вершину входят дуги, то работа может начаться только после окончания всех работ, из которых выходят эти дуги.
3. Ни одна работа, которая непосредственно следует за некоторой работой, не может начаться до момента ее окончания. Другими словами, если из работы выходит несколько дуг, то ни одна из работ, в которые входят эти дуги, не может начаться до окончания этой работы.
4. Начало и конец проекта обозначены работами с нулевой продолжительностью. Такие работы называются **вехами** и обозначают начало или конец наиболее важных этапов проекта.

**Задание № 1:** По предложенной таблице 1 построить **сетевой график** и определить **критический путь** проекта **«Разработка программного комплекса».**

**Для этого выполнить расчет раннего и позднего времени начала задач проекта и заполнить таблицу 2.**

**Таблица 1.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер работы** | **Название работы** | **Длительность** | **Предшественники** |
| 1 | Начало реализации проекта | 0 | - |
| 2 | Постановка задачи | 10 | 1 |
| 3 | Разработка интерфейса | 5 | 2 |
| 4 | Разработка модулей обработки данных | 7 | 3,5 |
| 5 | Разработка структуры базы данных | 6 | 2 |
| 6 | Заполнение базы данных | 8 | 5 |
| 7 | Отладка программного комплекса | 5 | 4,6 |
| 8 | Тестирование и исправление ошибок | 10 | 7 |
| 9 | Составление программной документации | 5 | 7 |
| 10 | Завершение проекта | 0 | 8,9 |

Сетевой график позволяет по заданным значениям длительностей работ найти критические работы проекта и его критический путь.

**Критической** называется такая работа, для которой задержка ее начала приведет к задержке срока окончания проекта в целом. Такие работы не имеют запаса времени. Некритические работы имеют некоторый запас времени, и в пределах этого запаса их начало может быть задержано.

**Критический путь** – это путь от начальной к конечной вершине сетевого графика, проходящий только через критические работы. Суммарная длительность работ критического пути определяет минимальное время реализации проекта.

Нахождение критического пути сводится к нахождению критических работ и выполняется в два этапа.

1. Вычисление **раннего времени начала** каждой работы проекта. Эта величина показывает время, раньше которого работа не может быть начата.
2. Вычисление **позднего времени начала** каждой работы проекта. Эта величина показывает время, позже которого работа не может быть начата без увеличения продолжительности всего проекта.

Критические работы имеют одинаковое значение раннего и позднего времени начала.

Обозначим t_i – время выполнения работы i, T_p(i) – раннее время начала работы i, T_П(i) – позднее время начала работы i. Тогда

T_p(i)=max(T_p(j)+t_j)\\
j\in G

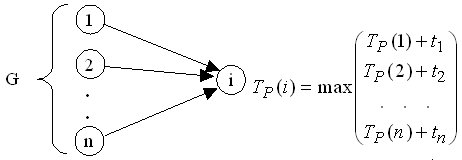

где G – множество работ, непосредственно предшествующих работе i. Раннее время начальной работы проекта принимается равным нулю.

Поскольку последняя работа проекта – это веха нулевой длительности, раннее время ее начала совпадает с длительностью всего проекта. Обозначим эту величину T. Теперь T принимается за позднее время начала последней работы, а для остальных работ позднее время начала вычисляется по формуле:

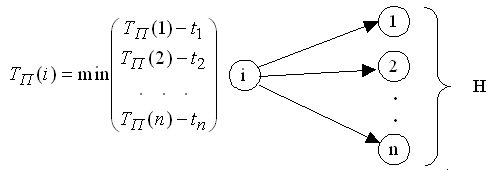
T_П(i)=min(T_П(j)-t_i)\\
j\in H


Здесь H – множество работ, непосредственно следующих за работой i.

Схематично вычисления раннего и позднего времени начала изображены, соответственно, на рис. 1 и рис.2.



**Рис. 1.**Схема вычисления раннего времени начала работы



**Рис. 2.**Схема вычисления позднего времени начала работы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 2. | | | | | |
| **Номер работы** | **Длительность** | **Трн** | **Тпн** | **Резерв (Тпн-Трн)** | **Критические работы** |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |

**Задание № 3. Построить календарный график проекта.**

Предположим, что в качестве ресурсов выступают только исполнители.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 2.3. | | | | | |
| **№ работы** | **Название работы** | **Длительность** | **Предшественники** | **Исполнитель** |
| 1 | Начало реализации проекта | 0 | - | - |
| 2 | Постановка задачи | 10 | 1 | Постановщик |
| 3 | Разработка интерфейса | 5 | 2 | Программист1 |
| 4 | Разработка модулей обработки данных | 7 | 3,5 | Программист1 |
| 5 | Разработка структуры базы данных | 6 | 2 | Программист2 |
| 6 | Заполнение базы данных | 8 | 5 | Программист2 |
| 7 | Отладка программного комплекса | 5 | 4,6 | Программист1  Программист2 |
| 8 | Тестирование и исправление ошибок | 10 | 7 | Программист1  Программист2  Постановщик |
| 9 | Составление программной документации | 5 | 7 | Постановщик |
| 10 | Завершение проекта | 0 | 8,9 | - |

Календарный график (диаграмма Ганта) строится в первой четверти координатной плоскости, по оси х откладывается время, по оси y работы. Ромбиками обозначаются вехи, сплошными линиями – продолжительность работ, сплошными линиями со стрелками – резерв времени работ, пунктирными линиями – связь между окончанием предшествующих и началом последующих работ.

**Задание № 4. Построить сетевой и календарный графики проекта «Модернизация компьютерного класса» основываясь на данных таблиц, полученных в работе № 5.**

**Задание № 5. Построить график загруженности ресурсов.**

На основании диаграммы Ганта может быть построен **график загруженности ресурсов**. Этот график показывает процент загрузки конкретного трудового ресурса в ходе выполнения проекта. По оси абсцисс откладывается временной интервал проекта, а по оси ординат – суммарный процент загруженности исполнителя по всем задачам проекта, которые он выполняет в текущий момент времени.

Обычно исполнитель целиком занят решением некоторой задачи и по ее завершении переходит к следующей. Это соответствует 100% загрузки.

Для построения использовать ось времени, полученную в календарном графике.

**Результат продемонстрировать преподавателю!**