Semestrální práce – KIV/PGS

## Struktura programu

### Main

Načte vstupní parametry z příkazové řádky a vypíše je. Vytvoří instance ostatních tříd potřebných k běhu programu. Spustí simulaci.

### Foreman

Načte ze vstupního souboru data, tedy bloky se zdroji z těžení a uloží si je do pole.

### Worker

Každý worker má své vlákno, které běží, dokud je co těžit. Bere si bloky od foremana a každý zdroj bloku těží tím, že je nad jeho vláknem volána metoda sleep() na nějaký čas.

Funkcí programu je simulace těžebního procesu. Těžbu provádí několik dělníků, kteří postupně nakládají vytěžené zdroje na náklaďáky, které je odváží pryč z dolu. Těžba zdrojů, jejich nakládání a odvoz jsou operace, které probíhají paralelně, proto musí být aplikace více vláknová. Vlastní vlákno potřebuje každý dělník a každý náklaďák. Vlákna dělníků poběží od začátku simulace, dokud bude co těžit. Vlákno náklaďáku poběží od chvíle, kdy bude náklaďák naplněn, až do chvíle, kdy dojede do cíle a náklad vyloží. Po spuštění program načte parametry z příkazové řádky a uloží si je do příslušných proměnných a spustí simulaci. Nejprve předák načte zdroje a uloží si je jako pole bloků, kde každý blok je reprezentován počtem zdrojů. Poté se najednou spustí vlákna všech dělníků. Funkce vláken je zajištěna tak, že třídy Lorry a Worker, reprezentující dělníky a náklaďáky implementují rozhraní Runnable. První kritickou sekcí programu je nakládání zdrojů na náklaďák, protože velmi často bude metodu pro naložení dělník ve chvíli, kdy ještě nakládá jiný. Proto je tato metoda synchronized, což znamená, že se metoda uzamkne pro dělníka, který momentálně nakládá, a pro dalšího dělníka se odemkne až poté co první dělník naloží náklad. V momentě, kdy se naplní náklaďák je třeba jeho vlákno pozastavit, dokud se nenaplní kapacita přívozu. Přívoz po naplnění přejede řeku a náklaďáky mohou pokračovat do cíle. To je druhou kritickou sekcí programu, aby všechny náklaďáky v přívozu počkali na jeho naplnění byla implementována bariéra. Ta počítá, kolik vláken ji volalo a každé vlákno pozastaví. Po naplnění kapacity přívozu voláním *notifyAll()*, všechny vlákna náklaďáků opět rozběhne. Poslední kritickou sekcí programu je ukončení simulace. Aby se nestalo, že se vypíší výsledky, poté co dělníci vytěží všechny zdroje, ale náklaďáky nejsou v cíli, je z metody *main()*, volána metoda *waitUntilOver()*. Tato metoda volá nad všemi vlákny dělníků a náklaďáků *join()*, aby simulace počkala než skončí. Pro jistotu je zde ještě cyklus který volá nad objektem *changedExtracted* příkaz *wait()*, dokud není roven počet vytěžených a počet nalezených zdrojů. Nad objektem *changedExtracted* je voláno notify(), pokaždé, když se navyšuje počet zdrojů dovezených do cíle. Nakonec programu se do konzole vypíše, kolik zdrojů vytěžil, který dělník a celkový počet vytěžených zdrojů. Události v programu se postupně vypisují do výstupního souboru.