

OMO – Smart Factory

Ovládání:

Program se spouští s argumentem čísla (1 až 6). Daná čísla odpovídají scénářům. Scénáře jsou popsány v Javadocu. Hodnoty jednotlivých scénářů (kapacitu skladů, množství materiálu, nakoupené stroje a najaté zaměstnance, typ vypsaných hlášení) je možno měnit přímo v nich.

Struktura:

Projekt Smart Factory je koncipován jakožto továrna vyrábějící auta. FactoryController je továrna a má produkční linku ProductionLine. Továrna má sklady aut, výrobních strojů a materiálu. Produkční linka ze skladu strojů vezme určité stroje, které zrovna potřebuje a pomocí nich auta vyrábí. Zároveň potřebným strojům přiřazuje pracovníky. Pracovníci se unavují a potřebují si odpočinout. Stroje se rozbíjejí a opraváři je musí opravit. Oprava trvá několik tiků. Auta jsou následně skladována ve skladu aut. Všechny sklady mají omezenou kapacitu. Manažer prochází celou továrnu a navštěvuje jednotlivé celky. Inspektor Prochází stroje od nejrozbítlejšího po nejméně rozbitý. Manažer a Inspektor vypisuje hlášení do loggeru.

Implementace:

Nejdříve je vytvořena továrna a produkční linka. Nakoupí se stroje, materiál a přijmou zaměstnanci. Následně se nastaví typ vyráběného auta a podle něj se nastaví produkční linka – zjistí se dostupnost strojů, seřadí se, přiřadí se pracovníci a sklad, ke kterému mají stroje přístup. Potom se zapne produkční linka, která postupně sestavuje auta pomocí strojů. Během toho se kontroluje, zda je dostatek materiálu pro další auto, zda se nějaký stroj nerozbil a zda se některý pracovník neunavil. Pokud se rozbil stroj, tak je pomocí opravářů opraven v několika tících. Pokud se unavil pracovník, tak si jde odpočinout a čeká se na něj, nebo se nasadí jiný pracovník. Programu je možné zadat počet tiků, po kterých má vytisknout hlášení událostí.

Hlášení (Reports):

Hlášení se vypisují na standardní výstup. Hlášení o spotřebě vytiskne spotřebu jednotlivých strojů a cenu spotřeby. Zároveň vytiskne celkovou spotřebu celé linky. Hlášení o konfiguraci vypíše nastavené celé továrny. Hlášení o výpadcích vytiskne počet všech výpadků, nejdelší výpadek, nejkratší výpadek a průměrnou dobu výpadku. Hlášení událostí vytiskne události podle typu události a zdroje.

Návrhové vzory:

Visitor – Manager, Inspector, OutageVisitor, ConsumptionMachineVisitor a AllEventVisitor jsou třídy využívající vzoru visitor.

Object Pool – Strojů, opravářů a pracovníků je omezený počet, jsou uchovávány a jsou používány znovu a znovu.

Strategy – Změna výrobní linky je koncipována pomocí Strategy patternu. Sekvence výrobní linky je uchovávána v LineSequence.

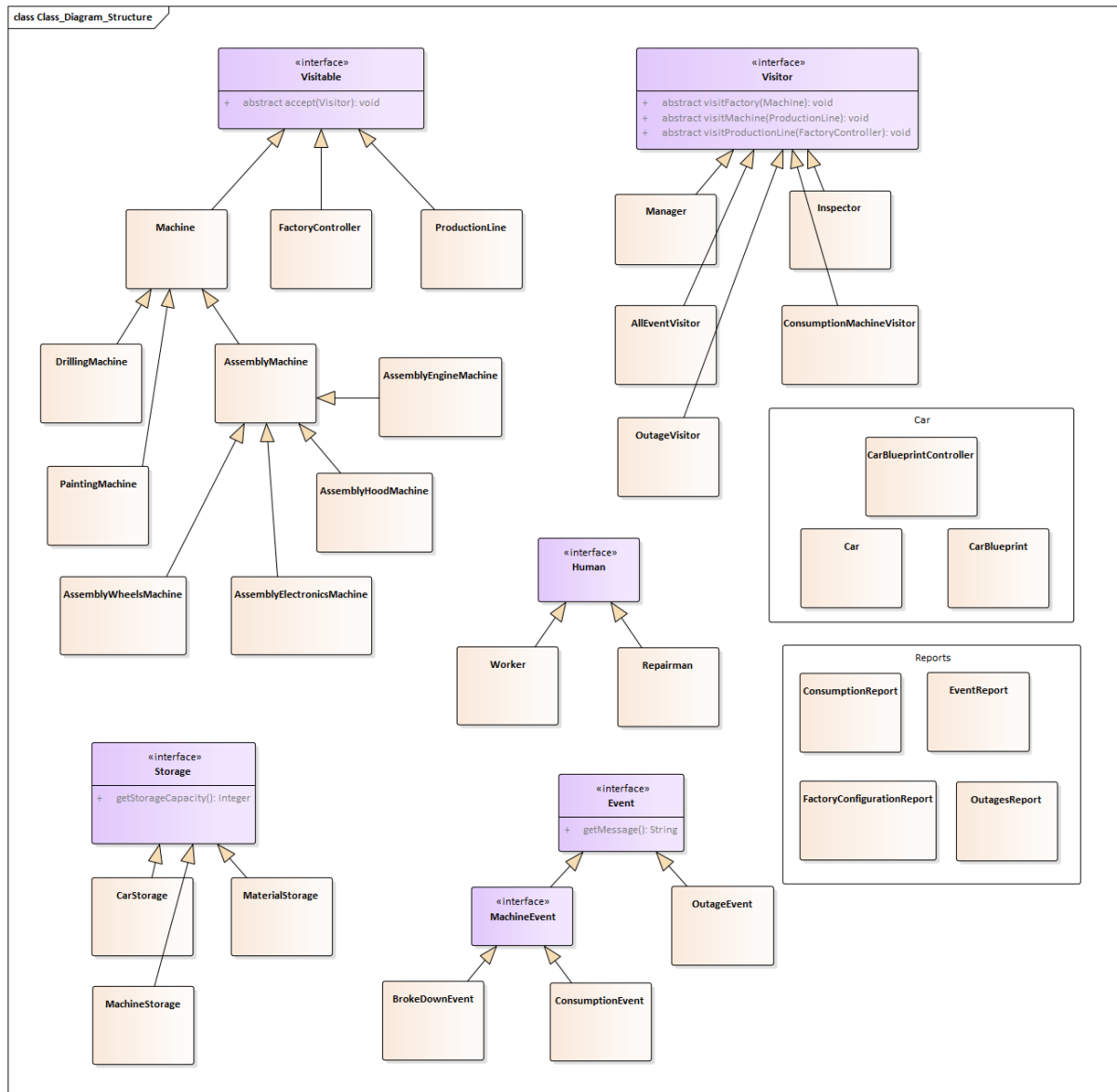
Builder – Stroje pracují v režimu Builder.

Decorator – U strojů (Machine) je použit Decorator – místo přímého použití processRequest a nutného zesložení je volána funkce „use“, která zároveň zaznamenává spotřebu a zaznamenává opotřebení.

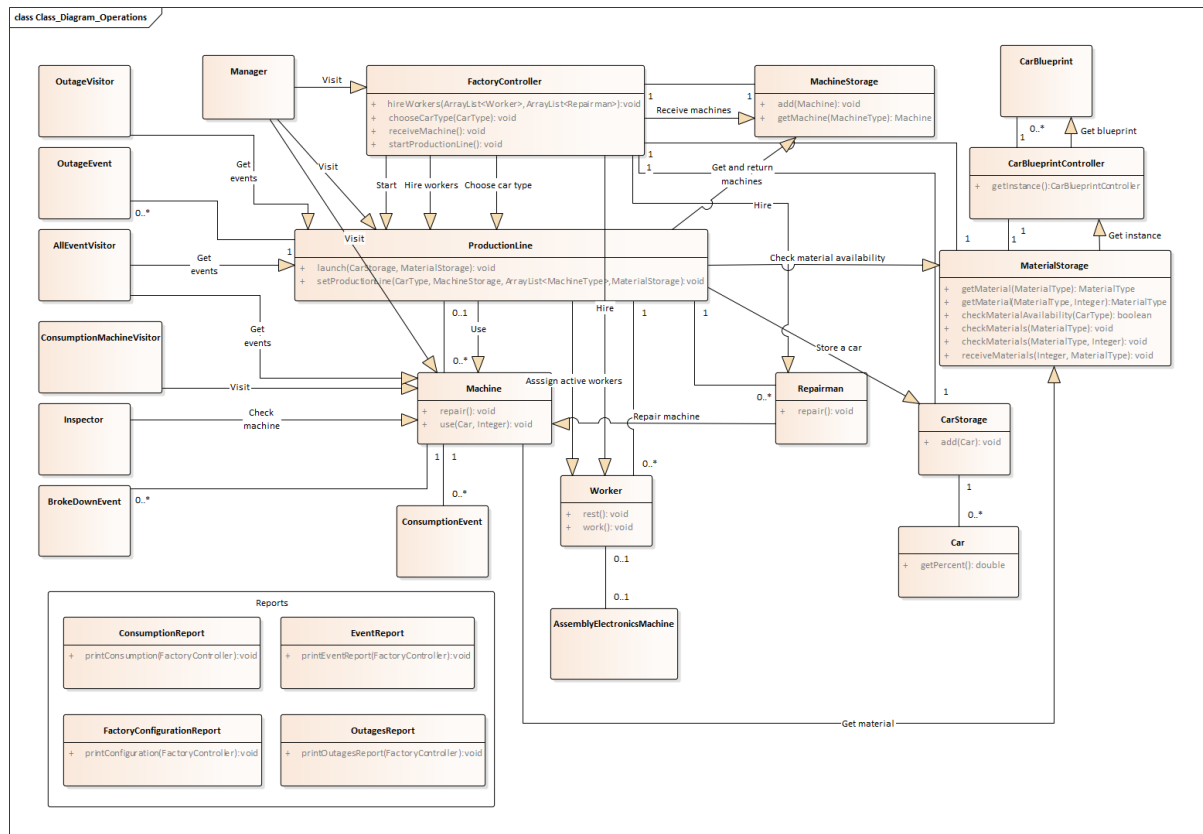
Singleton – Třída CarBlueprintController je postavena jako Singleton, protože stačí, aby existovala pouze jedna kopie této třídy poskytující plány na naše auta.

Diagramy (lepší kvalita v samostatné složce):

Struktura programu:



Hlavní operace v programu a asociativita mezi entitami:



Use case diagram:

