Program workflow

1. Program start - main()
   1. A singleton TuringMachine object is created
   2. *IOServiceFactory::createService()* reads the first two lines of the input and decides what *IOService* object to return. The developer have to derive from this class to add IO functionality from new type of stream
      1. *FileIOService* provides methods for input and output from a file
      2. *StandartIOService* provides methods for console input and output
   3. *IOService* object reads the machine information from the input and stores it in the *TuringMachine* object
   4. A design time defined object of type *MachineStrategy* executes an algorithm over the stored information in the *TuringMachine* object. The developer have to derive from *MachineStrategy* in order to create new algorithms
   5. *IOService* object prints the machine condition in the corresponding stream