**Behavioral Design Pattern Report**

Group 5

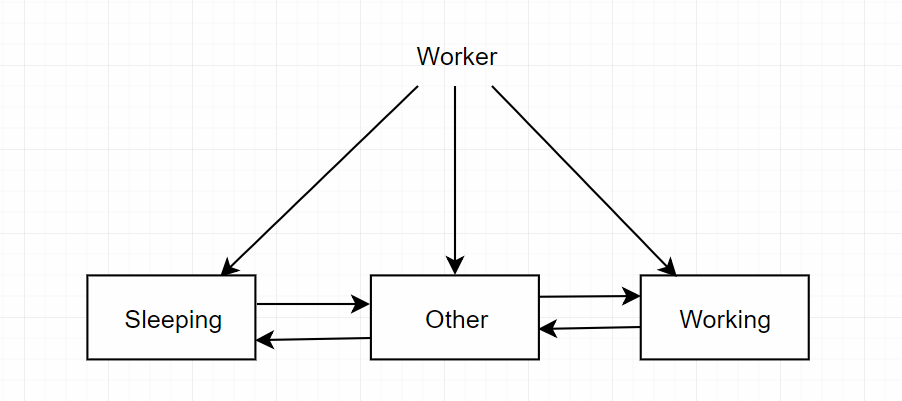
Bichi Zhang

04/23/2019

State

There will be three states for each worker: working state, sleeping state, other state. The time for each worker to change their states is usually fixed so we can use time to decide the worker’s current state and change to another state.

Structure:



Code:

File Name: WorkerState.h

Author: Bichi Zhang

Date: 04/28/2019

Complier Used: C++ by Visual Studio 2017

#pragma once

#include <iostream>

using namespace std;

class Worker;

class State

{

public:

virtual void doing(Worker\* w) {}

};

class WorkingState : public State

{

public:

void doing(Worker\* worker);

};

class SleepingState : public State

{

public:

void doing(Worker\* worker);

};

class OtherState : public State

{

public:

void doing(Worker\* worker);

};

File Name: Worker.h

Author: Bichi Zhang

Date: 04/28/2019

Complier Used: C++ by Visual Studio 2017

#pragma once

#include "WorkerState.h"

class Worker

{

public:

Worker()

{

state = new WorkingState();

}

void setState(State\* stat)

{

state = stat;

}

double getHour()

{

return hour;

}

void setHour(double hou)

{

hour = hou;

}

void requestDoing()

{

state->doing(this);

}

private:

State\* state;

double hour;

};

void WorkingState::doing(Worker\* worker)

{

if (worker->getHour() > 9 && worker->getHour() < 17)

cout << "WorkingState!" << endl;

else

{

worker->setState(new OtherState());

worker->requestDoing();

}

}

void SleepingState::doing(Worker\* worker)

{

if (worker->getHour() >= 24 || worker->getHour() < 8)

cout << "SleepingState!" << endl;

else

{

worker->setState(new OtherState());

worker->requestDoing();

}

}

void OtherState::doing(Worker\* worker)

{

if ((worker->getHour() >= 8 && worker->getHour() < 9)

|| (worker->getHour() >= 17 && worker->getHour() < 24))

cout << "OtherState!" << endl;

else if (worker->getHour() >= 24 || worker->getHour() < 8)

{

worker->setState(new SleepingState());

worker->requestDoing();

}

else

{

worker->setState(new WorkingState());

worker->requestDoing();

}

}

File Name: main.cpp

Author: Bichi Zhang

Date: 04/28/2019

Complier Used: C++ by Visual Studio 2017

#include "WorkerState.h"

#include "Worker.h"

int main()

{

Worker\* pWorker = new Worker();

pWorker->setHour(24);

cout << "time 24" << endl;

pWorker->requestDoing();

pWorker->setHour(11.5);

cout << "time 10.5" << endl;

pWorker->requestDoing();

pWorker->setHour(19);

cout << "time 19" << endl;

pWorker->requestDoing();

delete pWorker;

return 0;

}

Screenshot:

