FNAA

Konrad Maciejewski

Classes

```
class Employee {
    private:
        int id;
    public:
        float money;
        string name;
        int salary;
        Employee(int id, string name, int salary) { this->id = id; this->name = name; this->salary = salary; }
};

class Staff : public Employee {
    public:
        Staff(int id, string name, int salary) : Employee(id, name, salary) {}
};

class Motorist : public Employee {
    public:
        Motorist(int id, string name, int salary) : Employee(id, name, salary) {}
};
```

Classes

```
class Boss : public Employee {
    public:
        int drivers;
        Boss(int id, string name, int salary) : Employee(id, name, salary) {}
        void PaySalary(Staff &employee) {
            employee.money += employee.salary;
            money -= employee.salary;
            cout << "Paid " << employee.name << " " << employee.salary << "PLN" << endl;
        }
        void PaySalary(Motorist &employee) {
            employee.money += employee.salary;
            money -= employee.salary;
            cout << "Paid " << employee.name << " " << employee.salary << "PLN" << endl;
        }
        void getRicher() {
            money += salary;
            cout << "You got " << salary << "PLN" << endl;
        }
};
```

```
class Product {
    public:
        float price;
        string name;
        int quantity;
        Product(string name, float price, int quantity) { this->name = name; this->price = price; this->quantity = quantity;}
};
```

Classes

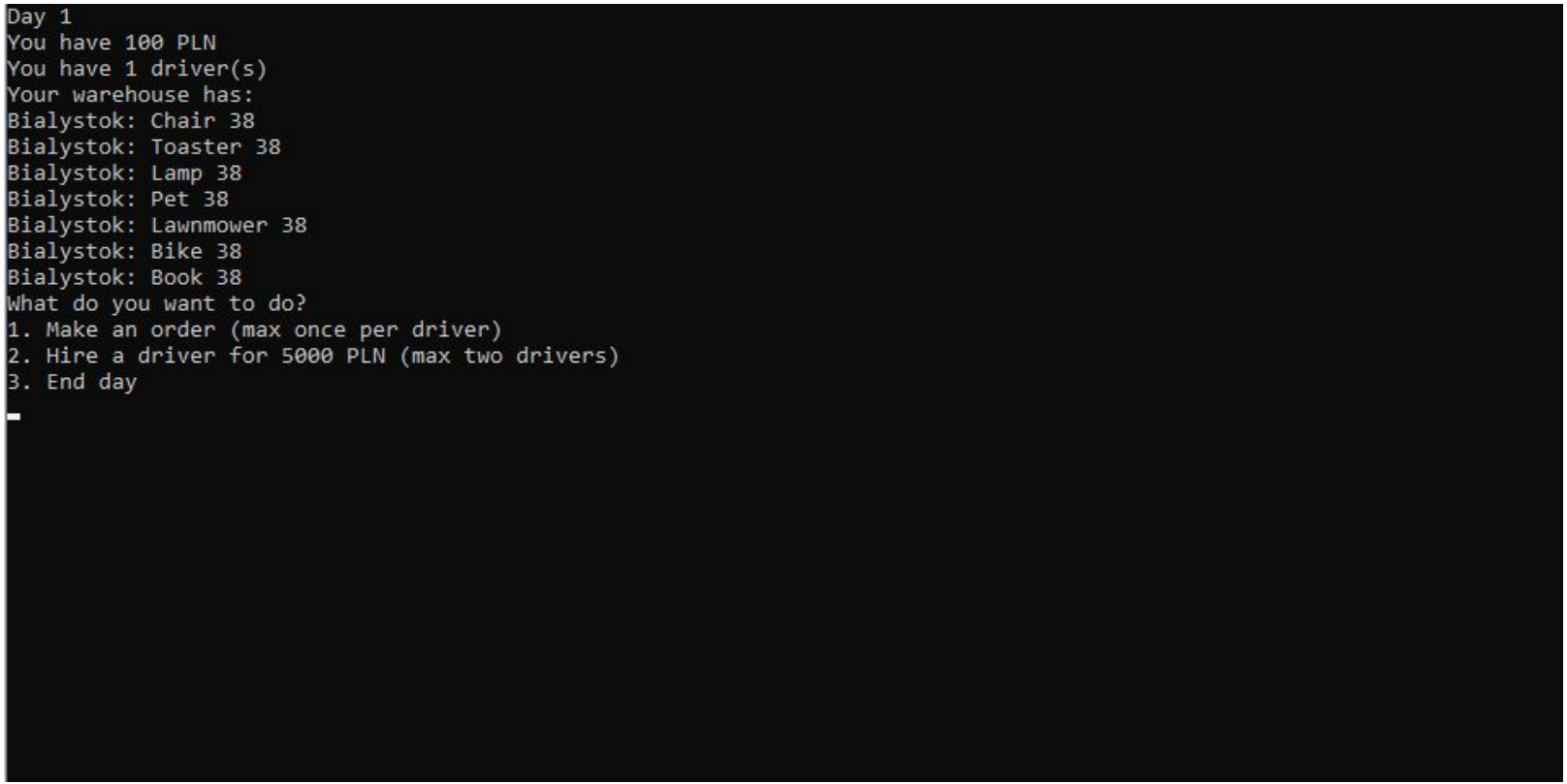
```
class Warehouse {
    private:
        vector<Product> products;
        vector<Employee> employees;
    public:
        string location;
        Warehouse(string location) { this->location = location; }
        void addEmployee(Employee employee);
        void addProduct(Product product);
        void shipProduct(string product, int n, bool gamePhase);
        void receiveProduct(string product, int n);
        int printMaxValue(string product);
        void printProducts();
};
```

Classes

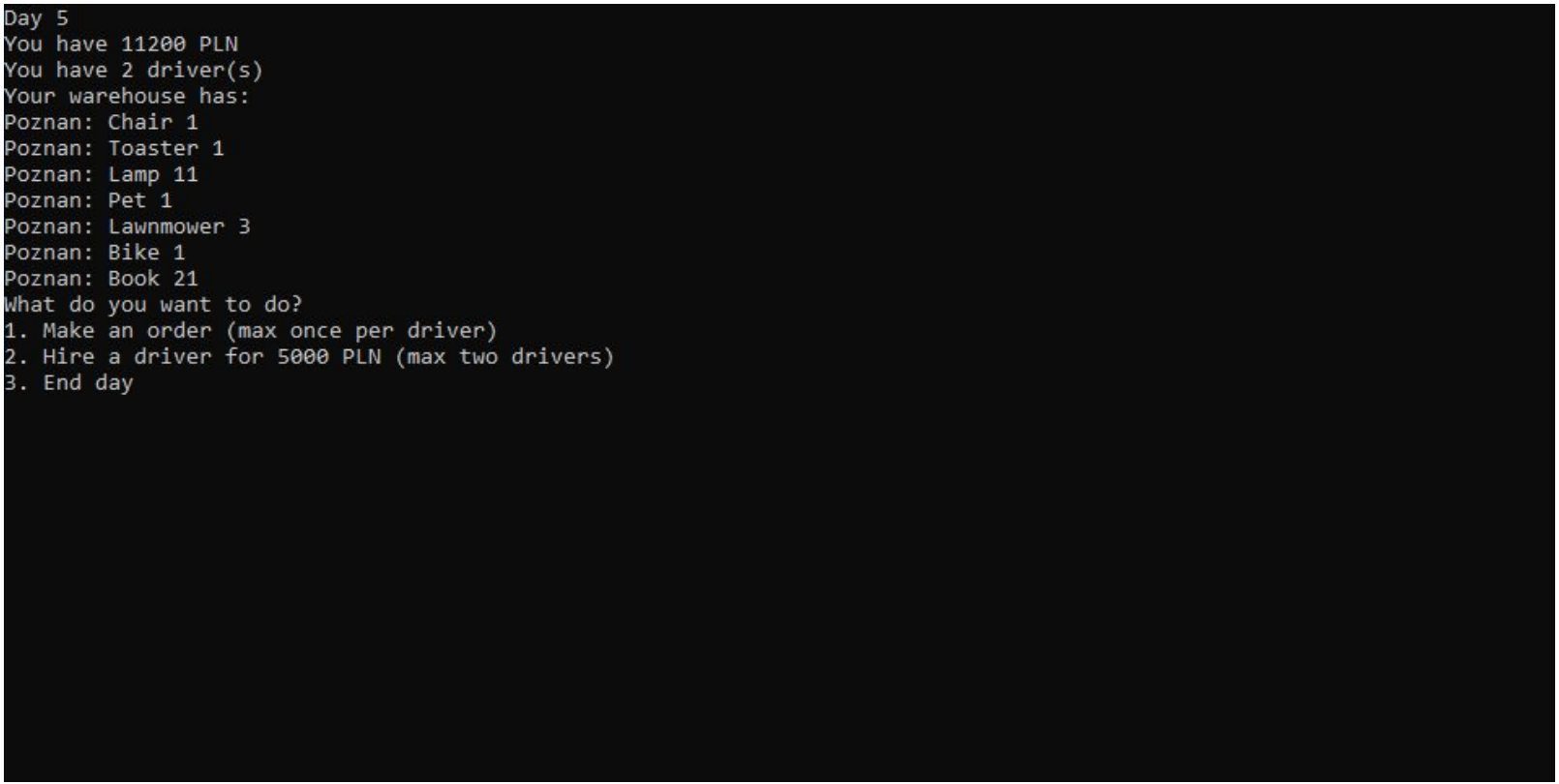
```
class Event {
    private:
        int mode;
        int probability;
    public:
        Event(int mode) { this->mode = mode; probability = mode*20; }
        void IRS(Boss &manager) {
            if (rand() % 100 + 1 <= probability) {
                cout << "Skarbowka came to your warehouse and took 5000PLN. Next time do your taxes" << endl;
                manager.money -= 5000;
            }
        }
        void fire(Boss &manager) {
            if (rand() % 100 + 1 <= probability) {
                cout << "There was a fire in your warehouse, you lost 1000PLN" << endl;
                manager.money -= 1000;
            }
        }
        void bialystokMoment(Boss &manager) {
            if (rand() % 100 + 1 <= probability) {
                cout << "You got robbed by bialystokian mafia, you lost 3000PLN" << endl;
                manager.money -= 3000;
            }
        }
};
```

Gameplay

Gameplay


Day 1
You have 100 PLN
You have 1 driver(s)
Your warehouse has:
Bialystok: Chair 38
Bialystok: Toaster 38
Bialystok: Lamp 38
Bialystok: Pet 38
Bialystok: Lawnmower 38
Bialystok: Bike 38
Bialystok: Book 38
What do you want to do?
1. Make an order (max once per driver)
2. Hire a driver for 5000 PLN (max two drivers)
3. End day

Gameplay


Day 5
You have 11200 PLN
You have 2 driver(s)
Your warehouse has:
Poznan: Chair 1
Poznan: Toaster 1
Poznan: Lamp 11
Poznan: Pet 1
Poznan: Lawnmower 3
Poznan: Bike 1
Poznan: Book 21
What do you want to do?
1. Make an order (max once per driver)
2. Hire a driver for 5000 PLN (max two drivers)
3. End day

Gameplay


You successfully retired!
You earned 12840 PLN
1600 PLN went to worker1
1600 PLN went to worker2
1600 PLN went to worker3
2400 PLN went to driver1
960 PLN went to driver2

Problems

```
void Sell(string product, int n, Warehouse &from, Boss &manager) {
    from.shipProduct(product, n, true);
    manager.money += n * 100;
}
```

```
void PaySalary(Motorist &employee) {
    employee.money += employee.salary;
    money -= employee.salary;
    cout << "Paid " << employee.name << " " << employee.salary << "PLN" << endl;
}
```

```
void Transport(string product, int n, Warehouse &from, Warehouse &to) {
    from.shipProduct(product, n, false);
    to.receiveProduct(product, n);
}
```

What have i learned?

```
void shipProduct(string product, int n, bool gamePhase) {
    for (int i = 0; i < products.size(); i++) {
        if (products[i].name == product) {
            if(gamePhase==false) {
                cout << "Shipped " << n << " " << product << " from " << location << endl;
            }
            if(gamePhase==true) {
                cout << "Sold " << n << " " << product << " for " << products[i].price*n << "PLN" << endl;
                for (int j = 0; j < employees.size(); j++) {
                    if (employees[j].name == "You") {
                        employees[j].money += products[i].price*n;
                    }
                }
            }
            products[i].quantity -= n;
        }
    }
}
```

```
void Transport(string product, int n, Warehouse &from, Warehouse &to) {
    from.shipProduct(product, n, false);
    to.receiveProduct(product, n);
}
```

What could be improved?

- less, more general classes
- randomness
- fix includes