Market Simulation

Damian Łukasiewicz

18 czerwca 2021

Spis treści

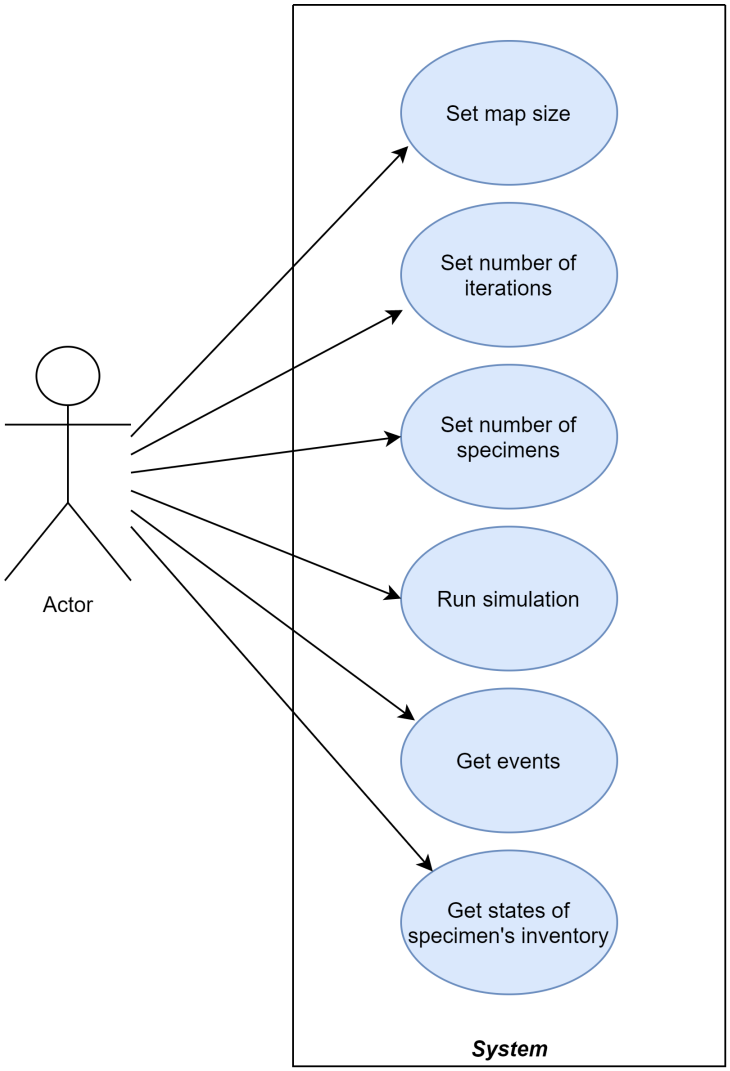
1 Opis symulacji

W symulacji istnieją 4 typy osobników - Sprzedawca, Kupujący, Złodziej oraz Strażnik. Na początku określana jest liczba każdego z typów osobników, liczba iteracji oraz wielkość planszy. Każda jednostka może zająć jedno pole. Gdy kupujący znajduje się na polu sąsiadującym ze stanowiskiem sprzedawcy to istnieje szansa na dokonanie zakupu określonych produktów. Istnieje także typ złodzieja, który będąc w pobliżu sprzedawcy kradnie mu określone produkty, ale trafiając na pole sąsiadujące ze strażnikiem musi oddać zasoby. Co stałą 10 epok sprzedawcy uzupełniają swoje zapasy wydając na to swój budżet uzyskany ze sprzedaży produktów. Z każdą epoką zapisywane są dane takie jak stan zasobów każdego z osobników oraz zdarzenia w każdej epoce.

2 Analiza czasownikowo-rzeczownikowa

W symulacji istnieją 4 typy osobników - Sprzedawca, Kupujący, Złodziej oraz Strażnik. Na początku określana jest liczba każdego z typów osobników, liczba iteracji oraz wielkość planszy. Każda jednostka może zająć jedno pole. Gdy kupujący znajduje się na polu sąsiadującym ze stanowiskiem sprzedawcy to istnieje szansa na dokonanie zakupu określonych produktów. Złodziej będąc w pobliżu sprzedawcy kradnie mu określone produkty, ale trafiając na pole sąsiadujące ze strażnikiem musi oddać zasoby. Co stałą 10 epok sprzedawcy uzupełniają swoje zapasy wydając na to swój budżet uzyskany ze sprzedaży produktów. Z każdą epoką zapisywane są dane takie jak stan zasobów każdego z osobników oraz zdarzenia w każdej epoce.

3 Diagram przypadków użycia



4 Karty CRC

4.1 Map

Classname: Map	
Superclass: none **Subclass(es)**: none	
Responsibilities: • generating a map • storing every positions of every Person	**Collaboration**: • Person

4.2 Person

Classname: Person	
Superclass: none **Subclass(es)**: Shopkeeper, Customer, Thief, Guard	
Responsibilities: • changing position • storing inventory	**Collaboration**: • Map

4.3 Customer

Classname: Customer	
Superclass: Person **Subclass(es)**: none	
Responsibilities: • buying items when near Shopkeeper	**Collaboration**: • Shopkeeper

4.4 Shopkeeper

Classname: Shopkeeper	
Superclass: Person **Subclass(es)**: none	
Responsibilities: • storing inventory • restocking inventory	**Collaboration**: • Customer • Thief

4.5 Thief

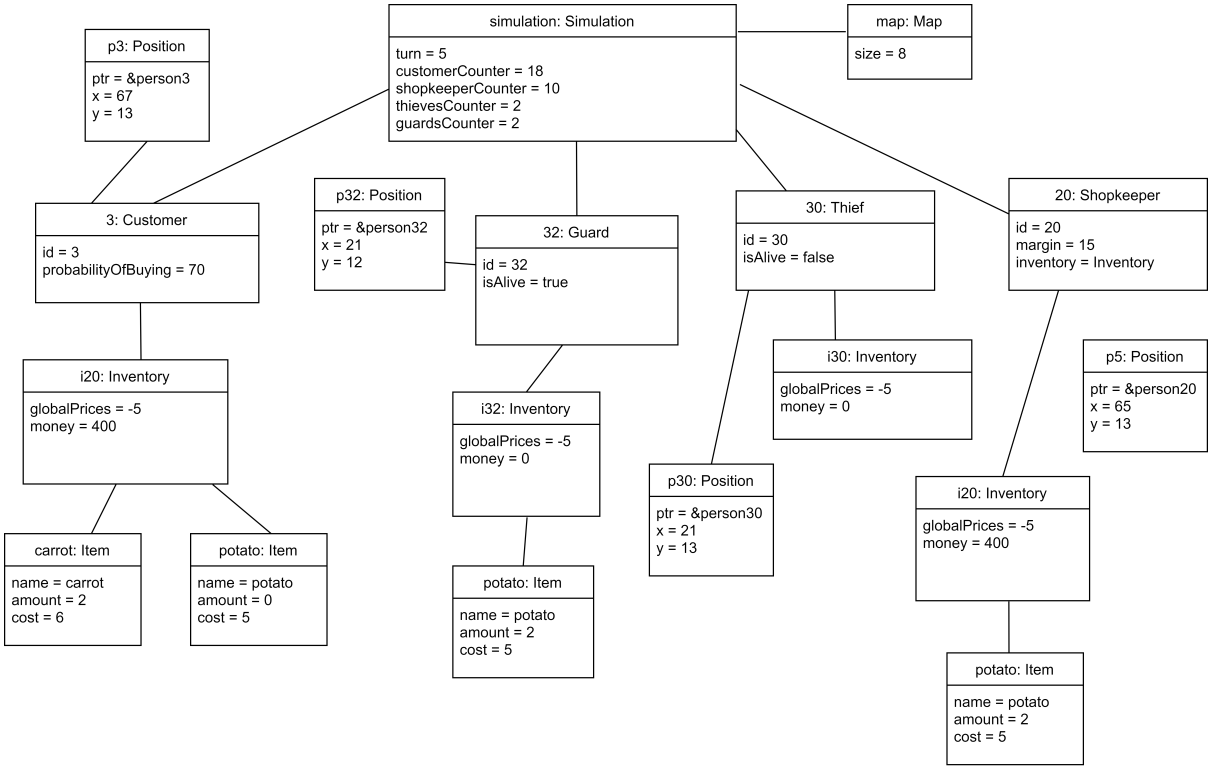
Classname: Thief	
Superclass: Person **Subclass(es)**: none	
Responsibilities: • stealing items when near Shopkeeper • returning items when caught by Guard	**Collaboration**: • Shopkeeper

4.6 Guard

Classname: Guard	
Superclass: Person **Subclass(es)**: none	
Responsibilities: • catches Thief when near him	**Collaboration**: • Thief

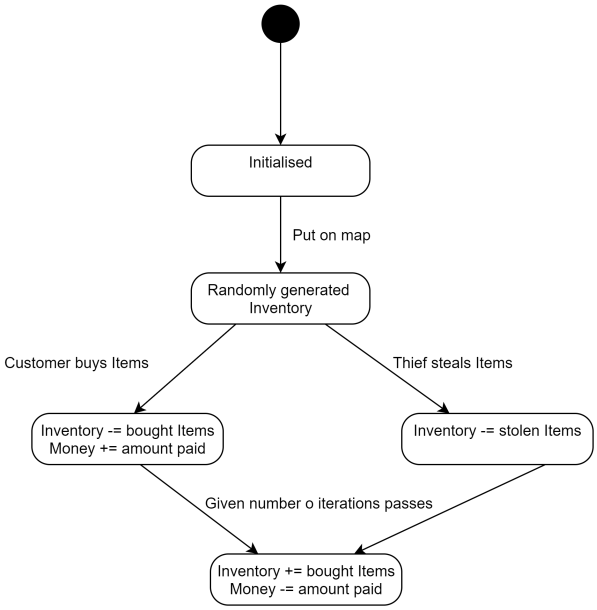
5 Diagram klas

6 Diagram obiektów

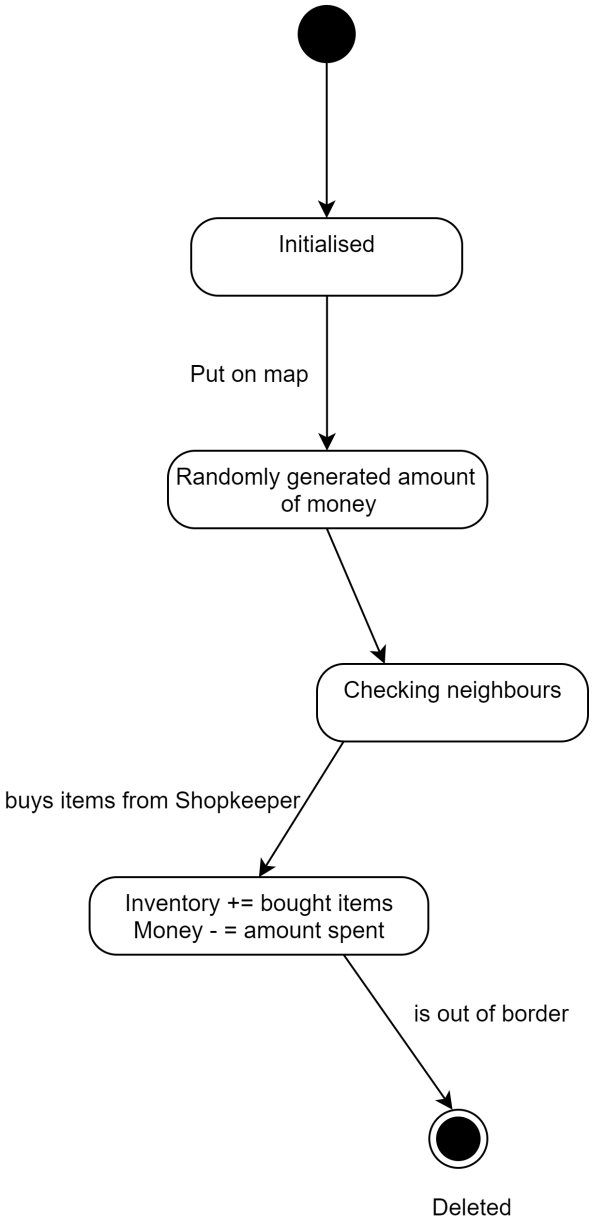


7 Diagramy maszyny stanów

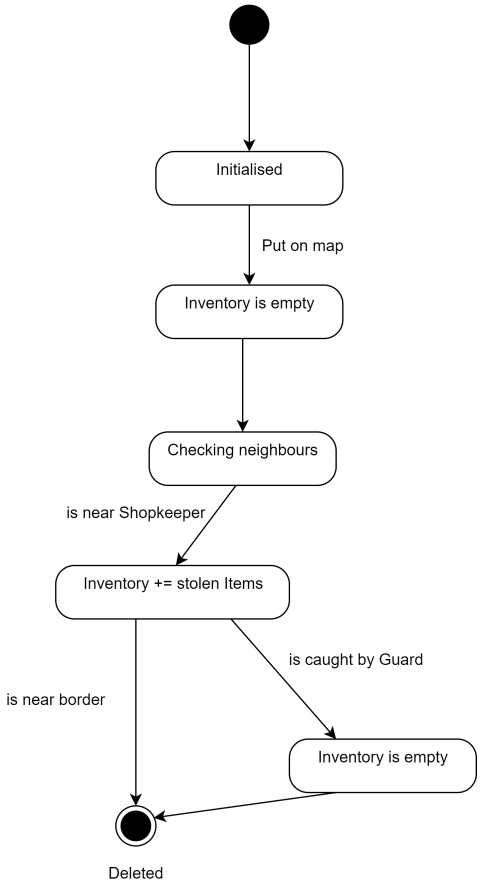
7.1 Sprzedawca



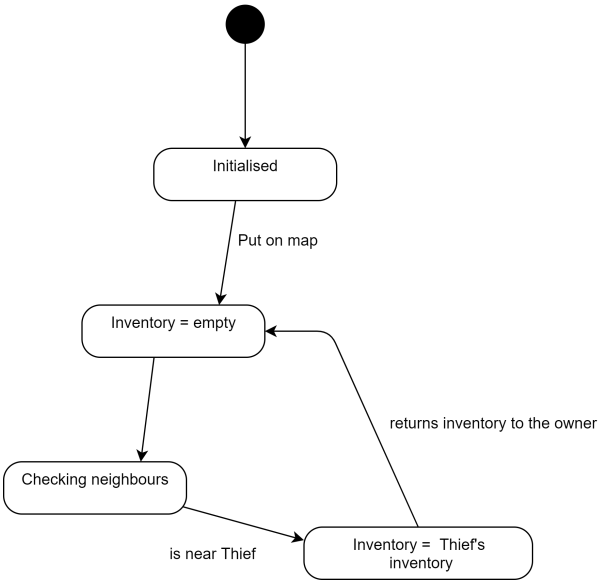
7.2 Kupujący


Initialised
Put on map
Randomly generated amount of money
Checking neighbours
buys items from Shopkeeper
Inventory += bought items
Money - = amount spent
is out of border
Deleted

7.3 Złodziej

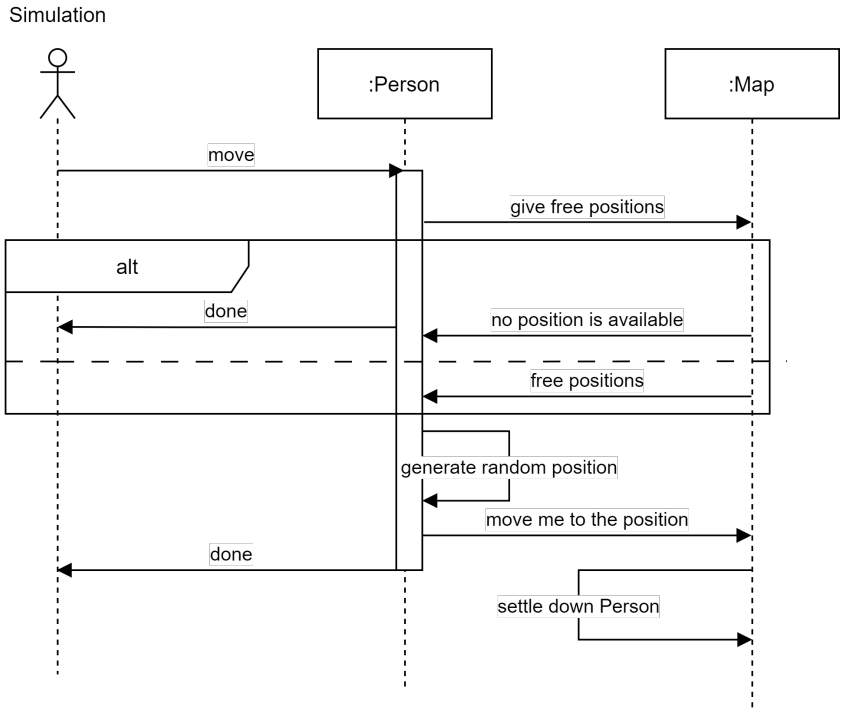


7.4 Strażnik

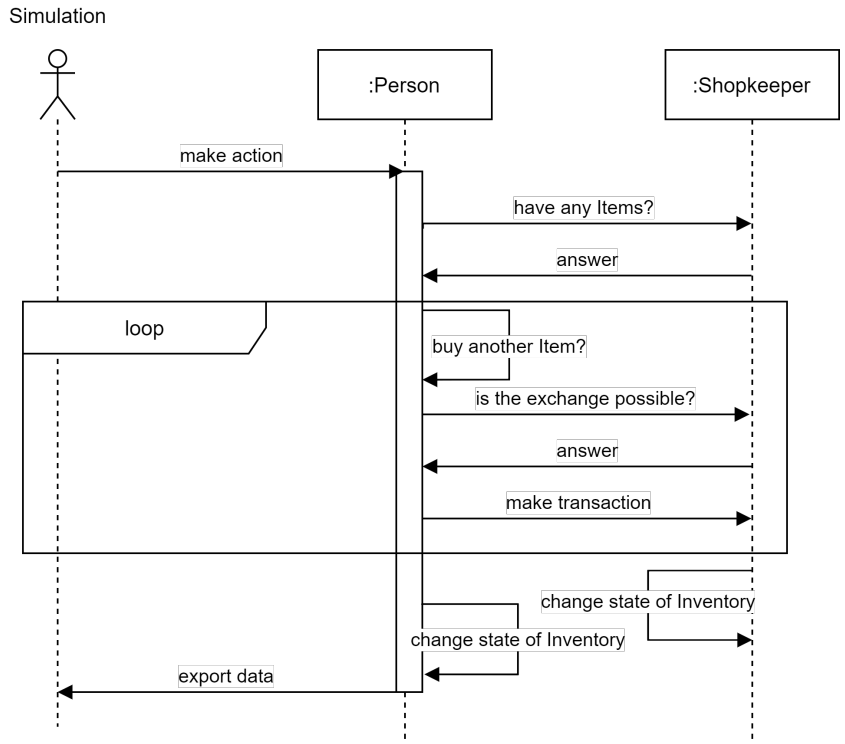


8 Diagramy sekwencji

8.1 Ruch na nową pozycję

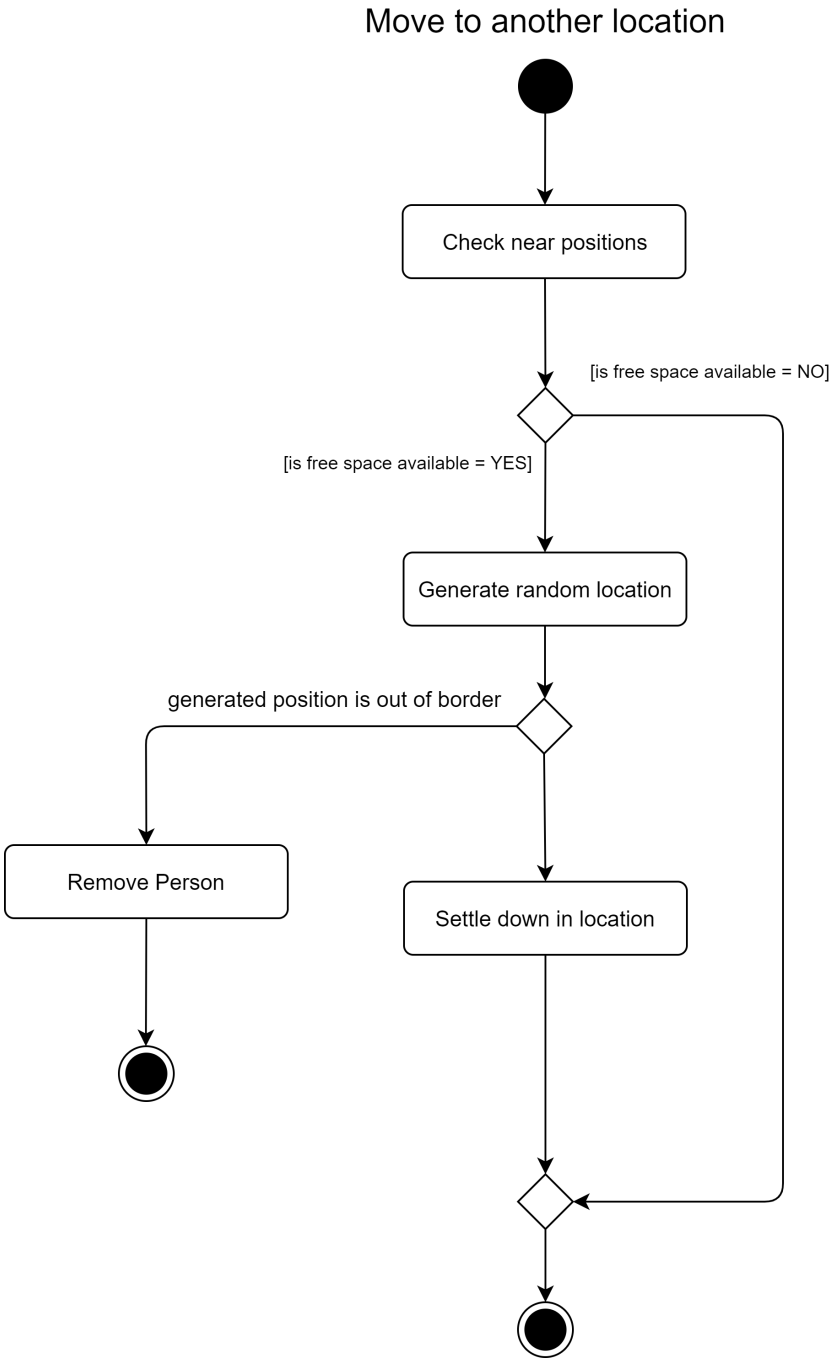


8.2 Kupno

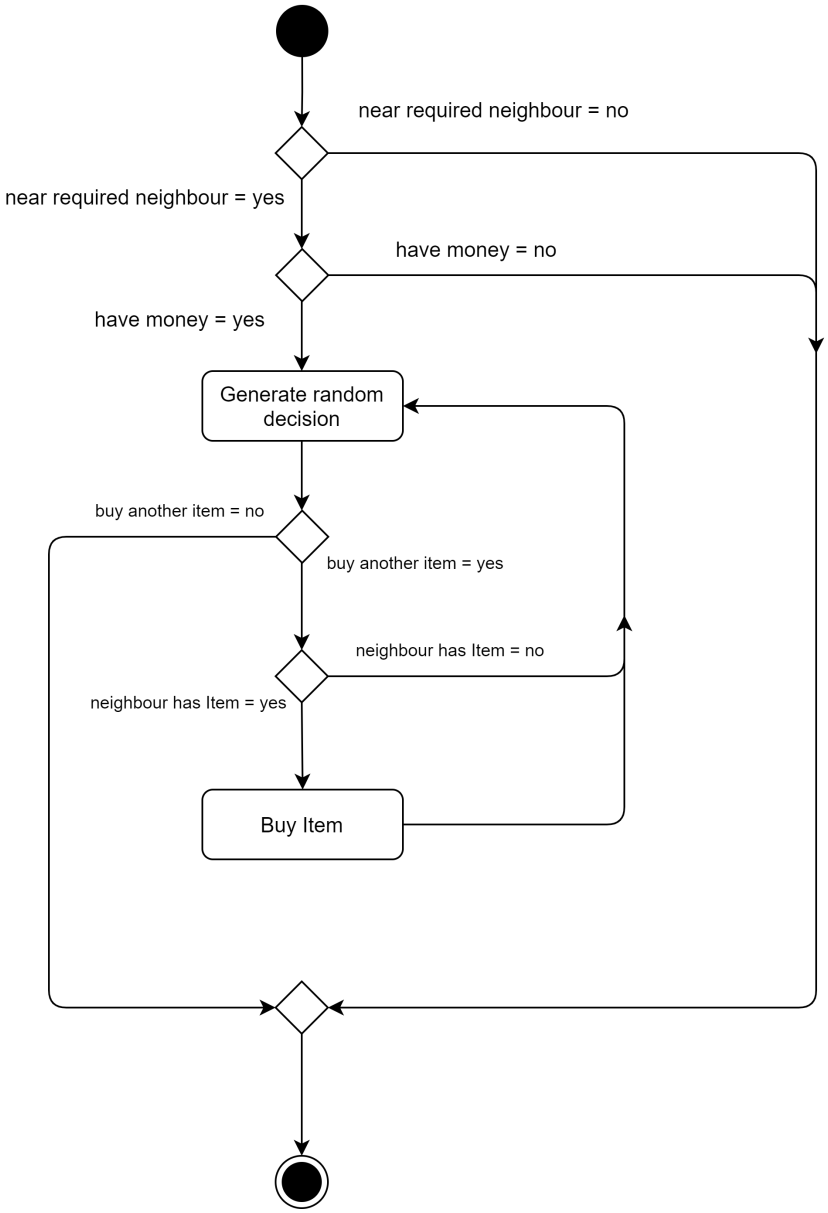


9 Diagramy aktywności

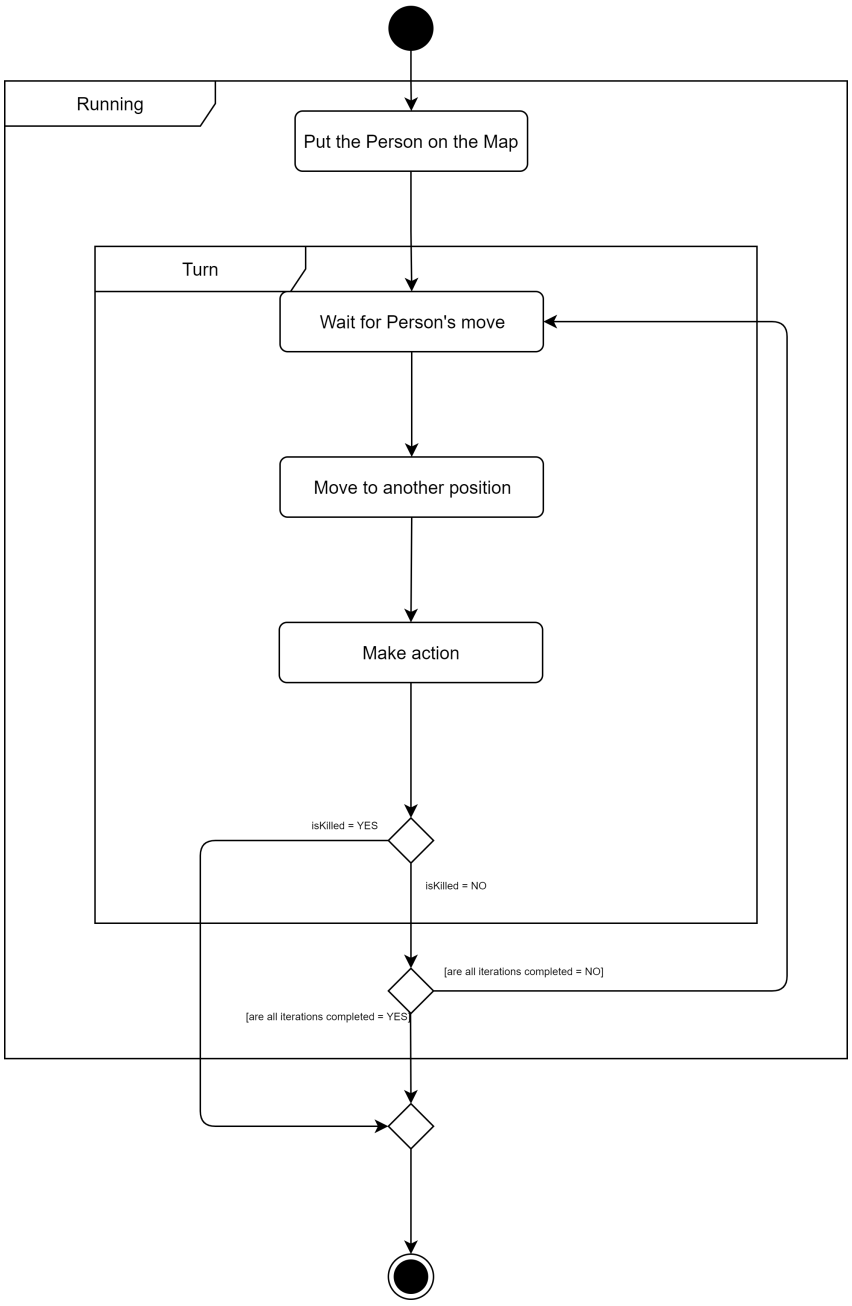
9.1 Ruch na nową pozycję



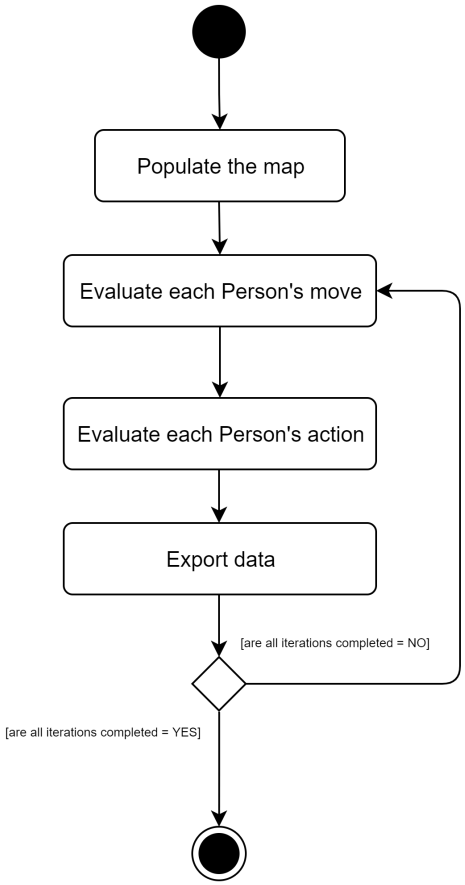
9.2 Kupno


near required neighbour = no
near required neighbour = yes
have money = no
have money = yes
Generate random decision
buy another item = no
buy another item = yes
neighbour has Item = no
neighbour has Item = yes
Buy Item

9.3 Osobnik


Running
Put the Person on the Map
Turn
Wait for Person's move
Move to another position
Make action
isKilled = YES
isKilled = NO
[are all iterations completed = NO]
[are all iterations completed = YES]

9.4 Symulacja



10 Doxygen

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

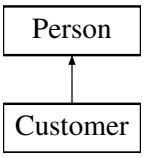
Class Documentation

3.1 Customer Class Reference

Represents customer.

`#include <Customer.h>`

Inheritance diagram for Customer:



Public Member Functions

- Customer (Position *position)
- void makeAction (Map *map)
- unsigned getProbability ()

Additional Inherited Members

3.1.1 Detailed Description

Represents customer.

Buys Items from Shopkeeper

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Customer()

```
Customer::Customer (
            Position * position )
```

Default constructor for Customer

Postcondition

Initializes random amount of money and random value of `probabilityOfBuying`

Parameters

position	Pointer to Position where Customer will be settled

3.1.3 Member Function Documentation

3.1.3.1 getProbability()

```
unsigned int Customer::getProbability ( )
```

Returns value of probabilityOfBuying

Postcondition

Does not change the object

Returns

Probability of buying

3.1.3.2 makeAction()

```
void Customer::makeAction (
            Map * map )  [virtual]
```

Buys Items when near him

Parameters

map	Pointer to map where other specimens are placed

Reimplemented from Person.

The documentation for this class was generated from the following files:

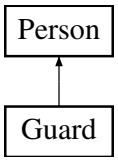
- C:/Users/ghj/Desktop/MarketSimulation/src/Customer.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Customer.cpp

3.2 Guard Class Reference

Represents guard.

`#include <Guard.h>`

Inheritance diagram for Guard:



Public Member Functions

- Guard (Position *position)
- void makeAction (Map *map)

Additional Inherited Members

3.2.1 Detailed Description

Represents guard.

Catches Thief when on nearby Position

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Guard()

```
Guard::Guard (
            Position * position )
```

Default constructor for Guard

Postcondition

Gets empty Inventory with 0 money

Parameters

position	Pointer to Position where Guard will be placed

3.2.3 Member Function Documentation

3.2.3.1 makeAction()

```
void Guard::makeAction (
            Map * map )  [virtual]
```

Catches Thief (see Thief::catchThief) and takes his Inventory

Parameters

map	Pointer to Map where other specimens are placed

Reimplemented from Person.

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Guard.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Guard.cpp

3.3 Inventory Class Reference

A class representing Person's inventory.

`#include <Inventory.h>`

Public Member Functions

- Inventory ()
- Inventory (unsigned money)
- void addMoney (int x)
- unsigned getMoney ()
- unsigned getAmountOfItems ()
- std::vector< Item > * getItems ()
- void addItem (Item item)

Static Public Member Functions

- static void setGlobalPrices (int newGlobalPrices)
- static int getGlobalPrices ()

Static Public Attributes

- static const std::vector< Item > defaultListOfItems

  Vector with types of all items available during simulation.

3.3.1 Detailed Description

A class representing Person's inventory.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 Inventory() [1/2]

```
Inventory::Inventory ( )
```

Default constructor

Postcondition

Inventory does not have any items, money = 0

3.3.2.2 Inventory() [2/2]

```
Inventory::Inventory (
            unsigned money )
```

Overloaded constructor

Postcondition

Inventory does not have any items

Parameters

money	Amount of money

3.3.3 Member Function Documentation

3.3.3.1 addItem()

```
void Inventory::addItem (
            Item item )
```

Adds one item of given type

Postcondition

If given type is already in inventory then it increments it's amount, if not adds new type of item with 1 amount

Parameters

item	Type of item to be added

3.3.3.2 addMoney()

```
void Inventory::addMoney (
            int x )
```

Adds money to inventory

Precondition

Amount to be added can be below 0

Postcondition

Money will be updated with added value. If (x <= money) money will be updated to 0

Parameters

x	Money to be added

3.3.3.3 getAmountOfItems()

```
unsigned int Inventory::getAmountOfItems ( )
```

Counts and returns amount of every type of items in inventory

Postcondition

Does not change the object

Returns

Amount of all items in inventory

3.3.3.4 getGlobalPrices()

```
int Inventory::getGlobalPrices ( )  [static]
```

Returns global prices

Postcondition

Does not change the object

Returns

Global prices

3.3.3.5 getItems()

```
std::vector< Item > * Inventory::getItems ( )
```

Returns pointer to vector with items

Postcondition

Does not change the object

Returns

Pointer to vector with items

3.3.3.6 getMoney()

```
unsigned int Inventory::getMoney ( )
```

Returns money in inventory

Postcondition

Does not change the object

Returns

Money in inventory

3.3.3.7 setGlobalPrices()

```
void Inventory::setGlobalPrices (
            int newGlobalPrices )  [static]
```

Sets new global prices

Postcondition

Global prices will be updated with new value

Parameters

newGlobalPrices	New global prices

3.3.4 Member Data Documentation

3.3.4.1 defaultListOfItems

```
const std::vector< Item > Inventory::defaultListOfItems  [static]
```

Initial value:

```
= {
        Item("carrot", 3),
        Item("apple", 5),
        Item("potato", 2),
        Item("chicken", 8),
        Item("milk", 6),
        Item("beef", 14)
        }
```

Vector with types of all items available during simulation.

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Inventory.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Inventory.cpp

3.4 Item Class Reference

A class representing type of item in Inventory.

```
#include <Inventory.h>
```

Public Member Functions

- Item (std::string name, unsigned price, unsigned amount)
- Item (std::string name, unsigned price)
- unsigned getPrice ()
- std::string getName ()
- unsigned getAmount ()
- void setAmount (unsigned newAmount)
- void addMargin (unsigned margin)
- void decrementAmount ()
- void incrementAmount ()

3.4.1 Detailed Description

A class representing type of item in Inventory.

This class represents a type of item not item itself so that it also consists of number of items. For example `name = "carrot"`, `price = 3`, `amount = 4`

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Item() [1/2]

```
Item::Item (
            std::string name,
            unsigned price,
            unsigned amount )
```

Default constructor

Parameters

Name	name of item
Price	price of item
Amount	number of items

3.4.2.2 Item() [2/2]

```
Item::Item (
            std::string name,
            unsigned price )
```

Overloaded constructor

Postcondition

Amount of items is 0

Parameters

name	Name of the Item
price	Price of an Item

3.4.3 Member Function Documentation

3.4.3.1 addMargin()

```
void Item::addMargin (
            unsigned margin )
```

Adds margin to price of an Item

Postcondition

Margin will be added to price

Parameters

margin	Margin added to price

3.4.3.2 decrementAmount()

```
void Item::decrementAmount ( )
```

Decrements amount of items

Postcondition

The amount of items will be decremented

3.4.3.3 getAmount()

```
unsigned int Item::getAmount ( )
```

Returns amount of items

Postcondition

Does not change the object

Returns

Number of items

3.4.3.4 getName()

```
std::string Item::getName ( )
```

Returns name of an item

Postcondition

Does not change the object

Returns

Name of Item

3.4.3.5 getPrice()

```
unsigned int Item::getPrice ( )
```

Returns price of an item

Postcondition

Does not change the object

Returns

Price of an item

3.4.3.6 incrementAmount()

```
void Item::incrementAmount ( )
```

Increments amount of items

Postcondition

The amount of items will be incremented

3.4.3.7 setAmount()

```
void Item::setAmount (
            unsigned newAmount )
```

Sets number of items

Postcondition

Amount of items will be updated with new value

Parameters

newAmount	New amount of items

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Inventory.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Inventory.cpp

3.5 Map Class Reference

Stores board with Positions.

`#include <Map.h>`

Public Member Functions

- Map (unsigned size)
- Position ∗ getPosition (unsigned x, unsigned y)
- Position ∗ getPosition (unsigned i)
- unsigned getSize ()

3.5.1 Detailed Description

Stores board with Positions.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 Map()

```
Map::Map (
            unsigned size )
```

Default constructor

Creates a square map with side of size and every position with nullptr

Parameters

size	Size of map

3.5.3 Member Function Documentation

3.5.3.1 getPosition() [1/2]

```
Position* Map::getPosition (
            unsigned i )
```

Returns Position with given index

Postcondition

Does not change the object

Parameters

i	Index of position in board

Returns

Pointer to Position

3.5.3.2 getPosition() [2/2]

```
Position* Map::getPosition (
            unsigned x,
            unsigned y )
```

Returns Position with given coordinates

Postcondition

Does not change the object

Parameters

x	X coordinate
y	Y coordinate

Returns

Pointer to Position

3.5.3.3 getSize()

```
unsigned int Map::getSize ( )
```

Returns size of the map

Postcondition

Does not change the object

Returns

Size of the map

The documentation for this class was generated from the following files:

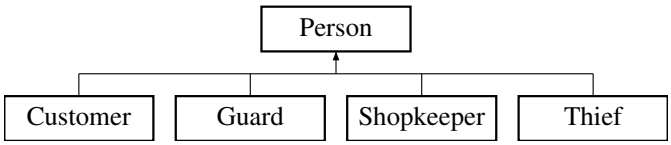
- C:/Users/ghj/Desktop/MarketSimulation/src/Map.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Map.cpp

3.6 Person Class Reference

Base class for species.

```
#include <Person.h>
```

Inheritance diagram for Person:



Public Member Functions

- Person (Position *position, unsigned money)
- Person (Position *position)
- virtual void move (Map *map)
- unsigned getID ()
- Position * getPosition ()
- Inventory * getInventory ()
- void kill ()
- bool isKilled ()
- virtual void makeAction (Map *map)
- virtual void sell (Person *buyer, unsigned probabilityOfBuying)
- virtual void steal (Person *thief, unsigned probability)
- virtual void catchThief (Person *guard)

Static Public Member Functions

- static unsigned getCounter ()

3.6.1 Detailed Description

Base class for species.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 Person() [1/2]

```
Person::Person (
            Position * position,
            unsigned money )
```

Default constructor for Person

Postcondition

Person's Inventory has no items

Parameters

position	Pointer to Position where Person is settled
money	Money in Person's Inventory

3.6.2.2 Person() [2/2]

```
Person::Person (
            Position * position )
```

Overloaded constructor for Person

Postcondition

Person's Inventory has no items and 0 money

Parameters

position	Pointer to Position where Person is settled

3.6.3 Member Function Documentation

3.6.3.1 catchThief()

```
void Person::catchThief (
            Person * guard )  [virtual]
```

Gets caught

For more information see Thief::catchThief

Parameters

guard	Pointer to Guard that is catching

Reimplemented in Thief.

3.6.3.2 getCounter()

```
unsigned int Person::getCounter ( )  [static]
```

Returns value of counter

Postcondition

Does not change object

Returns

Value of counter

3.6.3.3 getID()

```
unsigned int Person::getID ( )
```

Returns ID

Postcondition

Does not change object

Returns

ID

3.6.3.4 getInventory()

```
Inventory * Person::getInventory ( )
```

Returns pointer to Person's Inventory

Postcondition

Does not change object

Returns

Pointer to Person's Inventory

3.6.3.5 getPosition()

```
Position * Person::getPosition ( )
```

Returns pointer to Person's Position

Postcondition

Does not change object

Returns

Pointer to Person's Position

3.6.3.6 isKilled()

```
bool Person::isKilled ( )
```

Returns `true` if `isAlive == false`

Postcondition

Does not change object

Returns

Bool value

3.6.3.7 kill()

```
void Person::kill ( )
```

Sets `isAlive` to `false`

Postcondition

`isAlive` is set to `false`

3.6.3.8 makeAction()

```
void Person::makeAction (
            Map * map )  [virtual]
```

Makes actions according to it's type

For more information see:

- Shopkeeper::makeAction
- Customer::makeAction
- Thief::makeAction
- Guard::makeAction

Parameters

map	Pointer to Map with other specimens

Reimplemented in Thief, Shopkeeper, Guard, and Customer.

3.6.3.9 move()

```
void Person::move (
            Map * map )  [virtual]
```

Moves a Person to a random nearby Position if possible

Postcondition

If move is possible Person changes Position

Parameters

map	Pointer to map

Reimplemented in Shopkeeper.

3.6.3.10 sell()

```
void Person::sell (
            Person * buyer,
            unsigned probabilityOfBuying )  [virtual]
```

Sells Items

For more information see Shopkeeper::sell

Parameters

buyer	Pointer to Customer that is buying
probabilityOfBuying	Probability of buying an Item

Reimplemented in Shopkeeper.

3.6.3.11 steal()

```
void Person::steal (
            Person * thief,
            unsigned probability )  [virtual]
```

Gets robbed

For more information see Shopkeeper::steal

Parameters

thief	Pointer to Thief that is stealing
probability	Probability of stealing

Reimplemented in Shopkeeper.

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Person.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Person.cpp

3.7 Position Class Reference

A class representing position on Map.

```
#include <Map.h>
```

Public Member Functions

- Position (unsigned X, unsigned Y)
- Person * getPerson ()
- unsigned getX ()
- unsigned getY ()
- void setPointer (Person *newPtr)

3.7.1 Detailed Description

A class representing position on Map.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 Position()

```
Position::Position (
            unsigned X,
            unsigned Y )
```

Default constructor

Postcondition

Pointer is initialized to nullptr

Parameters

X	X coordinate
Y	Y coordinate

3.7.3 Member Function Documentation

3.7.3.1 getPerson()

```
Person * Position::getPerson ( )
```

Returns pointer to Person

Postcondition

Does not change the object

Returns

Pointer to Person

3.7.3.2 getX()

```
unsigned int Position::getX ( )
```

Returns x coordinate

Postcondition

Does not change the object

Returns

x coordinate

3.7.3.3 getY()

```
unsigned int Position::getY ( )
```

Returns y coordinate

Postcondition

Does not change the object

Returns

y coordinate

3.7.3.4 setPointer()

```
void Position::setPointer (
            Person * newPtr )
```

Sets new pointer

Postcondition

Pointer will be updated with new value

Parameters

newPtr	New value of pointer

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Map.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Map.cpp

3.8 Random Class Reference

Static class that generates random outcomes.

`#include <Random.h>`

Static Public Member Functions

- static int getRandInt (int min, int max)
- static bool getDecision (unsigned probability)
- static std::default_random_engine * getGenerator ()

3.8.1 Detailed Description

Static class that generates random outcomes.

3.8.2 Member Function Documentation

3.8.2.1 getDecision()

```
bool Random::getDecision (
            unsigned probability )  [static]
```

Returns random bool value

For example for probability of 70 there is 70% chance of returning `true`

Precondition

Probability <= 100. Otherwise returns false

Parameters

probability	Probability of returning true value

Returns

Random bool value

3.8.2.2 getGenerator()

```
std::default_random_engine * Random::getGenerator ( )  [static]
```

Returns pointer to random number generator

Returns

Pointer to random number generator

3.8.2.3 getRandInt()

```
int Random::getRandInt (
            int min,
            int max )  [static]
```

Returns random int in given range

Parameters

min	Minimal value
max	Maximal value

Returns

Random int value in range [min, max]

The documentation for this class was generated from the following files:

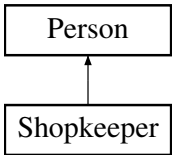
- C:/Users/ghj/Desktop/MarketSimulation/src/Random.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Random.cpp

3.9 Shopkeeper Class Reference

Represents shopkeeper.

```
#include <Shopkeeper.h>
```

Inheritance diagram for Shopkeeper:



Public Member Functions

- Shopkeeper (Position *position)

  Creates a Shopkeeper with random Inventory and 0 money.
- void makeAction (Map *map)
- void move (Map *map)
- void steal (Person *thief, unsigned probability)
- void sell (Person *buyer, unsigned probabilityOfBuying)

Additional Inherited Members

3.9.1 Detailed Description

Represents shopkeeper.

Doesn't move. When initialized gets Inventory with items from Inventory::defaultListOfItems with random amount of every Item. Can trade with Customer and be robbed by Thief

3.9.2 Constructor & Destructor Documentation

3.9.2.1 Shopkeeper()

```
Shopkeeper::Shopkeeper (
            Position * position )
```

Creates a Shopkeeper with random Inventory and 0 money.

Default constructor for Shopkeeper

Postcondition

Gets 0 money

Parameters

position	Position where Shopkeeper will be settled

3.9.3 Member Function Documentation

3.9.3.1 makeAction()

```
void Shopkeeper::makeAction (
            Map * map )  [virtual]
```

After 10 calls it will restock (Shopkeeper::restock)

Parameters

map	Map where other specimens are placed

Reimplemented from Person.

3.9.3.2 move()

```
void Shopkeeper::move (
            Map * map )  [virtual]
```

Stays in position

Parameters

map	Map where other specimens are placed

Reimplemented from Person.

3.9.3.3 sell()

```
void Shopkeeper::sell (
            Person * buyer,
            unsigned probabilityOfBuying )  [virtual]
```

Sells Items to buyer

Parameters

buyer	Pointer to Customer
probabilityOfBuying	Probability of buying (see Random::getDecision)

Reimplemented from Person.

3.9.3.4 steal()

```
void Shopkeeper::steal (
            Person * thief,
            unsigned probability )  [virtual]
```

Gets robbed by Thief

There is `probability` chance of stealing an Item, loops for every Item

Parameters

thief	Pointer to Thief
probability	Probability of stealing (see Random::getDecision)

Reimplemented from Person.

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Shopkeeper.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Shopkeeper.cpp

3.10 Simulation Class Reference

Runs simulation.

`#include <Simulation.h>`

Public Member Functions

- Simulation (unsigned size, unsigned n_customers, unsigned n_shopkeepers, unsigned n_thieves, unsigned n_guards)
- void runSimulation (unsigned n_iterations)
- void exportEvents (unsigned turn)
- void exportSpecies (unsigned turn)
- void printSpecies ()
- void printEvents (unsigned turn)

Static Public Member Functions

- static void addEvent (std::string type, Person ∗person, Person ∗otherPerson, unsigned numberOfItems, unsigned money)
- static void addEvent (std::string type, Person ∗person, unsigned numberOfItems, unsigned money)

3.10.1 Detailed Description

Runs simulation.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 Simulation()

```
Simulation::Simulation (
            unsigned size,
            unsigned n_customers,
            unsigned n_shopkeepers,
            unsigned n_thieves,
            unsigned n_guards )
```

Default constructor

Makes Map with given size and randomly populates it with specimens

Precondition

Population must be smaller than number of Positions (size∗size)

Parameters

size	Size of map
n_customers	Number of Customers
n_shopkeepers	Number of Shopkeepers
n_thieves	Number of Thieves
n_guards	Number of Guards

3.10.3 Member Function Documentation

3.10.3.1 addEvent() [1/2]

```
void Simulation::addEvent (
            std::string type,
            Person * person,
            Person * otherPerson,
            unsigned numberOfItems,
            unsigned money )  [static]
```

Adds data about an event to static buffer

Parameters

type	Type of event
person	Pointer to first Person
otherPerson	Pointer to other Person
numberOfItems	Number of exchanged Items
money	Amount of exchanged money

3.10.3.2 addEvent() [2/2]

```
void Simulation::addEvent (
            std::string type,
            Person * person,
            unsigned numberOfItems,
            unsigned money )  [static]
```

Adds data about an event to static buffer

Parameters

type	Type of event
person	Pointer to first Person
numberOfItems	Number of exchanged Items
money	Amount of exchanged money

3.10.3.3 exportEvents()

```
void Simulation::exportEvents (
            unsigned turn )
```

Exports data about events to Events.csv

Parameters

turn	

3.10.3.4 exportSpecies()

```
void Simulation::exportSpecies (
            unsigned turn )
```

Exports data about state of species for every turn to Species.txt

3.10.3.5 printEvents()

```
void Simulation::printEvents (
            unsigned turn )
```

Prints data about events

3.10.3.6 printSpecies()

```
void Simulation::printSpecies ( )
```

Prints data about state of species

3.10.3.7 runSimulation()

```
void Simulation::runSimulation (
            unsigned n_iterations )
```

Runs Simulation for given number of iterations

After each iterations saves exports data

Parameters

n_iterations	Number of iterations to run

The documentation for this class was generated from the following files:

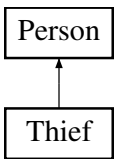
- C:/Users/ghj/Desktop/MarketSimulation/src/Simulation.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Simulation.cpp

3.11 Thief Class Reference

Represents shopkeeper.

`#include <Thief.h>`

Inheritance diagram for Thief:



Public Member Functions

- Thief (Position *position)

  Creates Thief with 0 money and empty Inventory.
- void makeAction (Map *map)
- void catchThief (Person *guard)

Additional Inherited Members

3.11.1 Detailed Description

Represents shopkeeper.

Steals Items from Shopkeeper when near him. Can be caught by Guard

3.11.2 Constructor & Destructor Documentation

3.11.2.1 Thief()

```
Thief::Thief (
            Position * position )
```

Creates Thief with 0 money and empty Inventory.

Default constructor

Postcondition

Gets 0 money and empty Inventory

Parameters

position	

3.11.3 Member Function Documentation

3.11.3.1 catchThief()

```
void Thief::catchThief (
            Person * guard )  [virtual]
```

Gets caught by Guard

Postcondition

Gets killed

Parameters

guard	Pointer to Guard that caught Thief

Reimplemented from Person.

3.11.3.2 makeAction()

```
void Thief::makeAction (
            Map * map )  [virtual]
```

Steals from Shopkeeper when near him

For more details see Shopkeeper::steal

Parameters

map	Map where other specimens are placed

Reimplemented from Person.

The documentation for this class was generated from the following files:

- C:/Users/ghj/Desktop/MarketSimulation/src/Thief.h
- C:/Users/ghj/Desktop/MarketSimulation/src/Thief.cpp

Index