

Джурасик парк

Връзка към проекта: <https://github.com/MLambeva/Jurassic-park.git>

Изготвил: Моника Ламбева, ФН: 82017

1. Увод:

- 1.1** Идеята на проекта е да наподобим успешен развъдник за динозаври. В този развъдник има клетки с динозаври, склад за храна и персонал. В него може да се приема ново животно, да се строят нови клетки, да се хранят животните и да се зарежда склада с храна.
- 1.2** Целта на разработката е успешно да се пресъздаде домашен зоопарк за праисторически влечуги като този зоопарк може да достигне много големи размери. В него първоначално има клетки без никакви животни, а в склада има по 300кг храна от всеки вид – трева, месо и риба. В зоопарка могат да живеят няколко вида животни (плезиозавър, бронтозавър, тиранозавър и птицевзавър), които не съжителстват заедно, т.е. те трябва да са в отделни клетки. Те приемат различно количество храна: плезиозавър – 15кг риба, бронтозавър – 20кг трева, тиранозавър – 50кг месо, птицевзавър – 35кг месо. В отделни клетки са също и животни, които са от различна ера. В зоопарка е нужно да има достатъчно персонал и за тази цел съм предвидила за 1 животно да се грижат 2 човека, за 2 животни – 4 човека, за 3 животни – 5 човека, за 4 животни – 6 човека и така до 10 животни – 12 човека. Трябва да има храна за всички животни, затова, когато храната не е достатъчно, складът се зарежда с избрано от потребителя количество храна.
- 1.3** В документацията по-долу съм разяснила по-подробна информация относно:
- Как съм проектирала класовете си и какви член данни и методи съм използвала в тях.
 - Как съм реализирала методите в класовете.
 - Как мисля, че съм се справила и мнението ми за бъдещо усъвършенстване.

2. Проектиране:

2.1 Проектът се състои от:

- пет файла с разширение “.h”, в които са дефинирани отделните класове;
- шест файла с разширение “.cpp”, в пет от които са имплементирани конструктурите и функциите на отделните класове, а шестият е главната функция, в която се реализира проектът.

2.2 Основните класове в проекта са:

- **Клас Динозавър**, който се характеризира с:
- име (име, започващо с главна латинска буква, последвано от малки латински букви);
 - пол (женски, мъжки);
 - ера (триас, креда или юра);
 - разред (тревопасен, месояден, летящ или воден);
 - вид (плезиозавър, бронтозавър, тиранозавър и птицевзавър);
 - храна (трева, месо или риба);

В клас Динозавър са дефинирани конструктор по подразбиране, оператори за сравняване на 2 динозавъра, селектори, оператор за изход, функция за създаване на животно и две функции - едната за записване на информация за динозавър в бинарен файл, а другата – за четене на информация за динозавър от бинарен файл.

➤ **Клас Клетка**, който се характеризира с:

- размер (малък, среден или голям);
- климат (сухоземен, въздушен или воден);
- брой работници в клетката (за 1 животно се грижат 2 човека, за 2 животни – 4 човека, за 3 животни – 5 човека, за 4 животни – 6 човека и така до 10 животни – 12 човека).
- животни, които я обитават;
- ера на животните вътре;

В клас клетка са дефинирани конструктор по подразбиране, конструктор, приемащ два аргумента – размер и климат, селектори, мутатори, две функции - едната за строене на клетка за съществуващо животно, а другата – за строене на клетка по даден размер и климат, функция за премахване на животно от клетка и съответно премахване и на персонал, оператор за изход, две функции - едната за записване на информация за клетка в бинарен файл, а другата – за четене на информация за клетка от бинарен файл.

➤ **Клас Склад**, който се характеризира с:

- количество трева;
- количество месо;
- количество риба;

В клас склад са дефинирани конструктор по подразбиране, селектори, мутатори, функции за зареждане на склада, оператор за изход, две функции - едната за записване на информация за склада в бинарен файл, а другата – за четене на информация за склада от бинарен файл.

➤ **Клас Помощни функции**, в който са дефинирани статични функции за коректност, т.е. функции, които проверяват дали правилно са въведени данните от потребителя. Понеже са статични, те могат да се използват без създаване на обект от дадения клас.

➤ **Клас Джурасик парк**, който се характеризира с:

- множество от клетки
- склад с храна.

В него са дефинирани пет функции, които са в „private” частта, с помощта на които записваме информация за всички компоненти, нужни в парка, а именно – информация за клетките и наличните количества храна в склада и съответно другата е за извличане на същата информация от бинарния файл, чрез третата проверяваме дали съществува бинарен файл и дали в парка има някакви клетки и животни, чрез четвъртата при неначилене на такива, се записват във файла три клетки, а петата е за изпечатване на информация. В „public” частта има 2 конструкция и една функция, която е фундаментална за целия проект, защото чрез нея се добавят нови животни, създават се нови клетки, премахва се вече съществуващо животно, хранят се животните, зарежда се склада с храна, може да се види какви животни има в зоопарка и да се финализира Джурасик парка, т.е. да се запазят текущите промени.

class Dinosaur

```
std::string name;  
std::string gender;  
std::string era;  
std::string order;  
std::string species;  
std::string food;
```

class Warehouse

```
int quantityGrass = 300;  
int quantityMeat = 300;  
int quantityFish = 300;
```

class Cages

```
std::string size;  
std::string climate;  
int countEmp = 0;  
std::string era;  
std::vector<Dinosaur> animals;
```

class JurassicPark

```
vector<Cage> cages
```

```
Warehouse warehouse
```

3. Реализация

3.1 Реализиране на методите в класовете:

➤ В клас **Динозавър** са имплементирани:

- конструктор по подразбиране, чрез който се създава обект от класа, на който всичките член данни са празни.
- оператори за сравняване на 2 динозавъра, т.е. срявnavят член данните им. Използват се, когато се налага да бъде премахнато животно от парка и чрез тях се подсигурява, че ще премахнем животното, което потребителят желае.
- селектори, за да може да достъпваме елементи на класа извън него.
- оператор за изход – извежда информация за член данните на динозавъра.
- функция за създаване на животно по име, пол, ера и разред. Според ерата и разреда се определя вида на животното и храната, която яде, защото имаме 4 вида животни, от които Плезиозавърът е воден, Бронтозавърът е тревопасен, Тиранозавърът и Птицезавърът са месоядни.
- функция за записване на информация за динозавър в бинарен файл, която работи по следния начин:
 - 1) В новосъздадена променлива запазваме размера на стринга.
 - 2) Записваме във файла размера.
 - 3) Записваме информацията от самия стринг.
- функция за четене на информация за динозавър от бинарен файл, която работи по следния начин:
 - 1) Прочитаме дължината на стринга.
 - 2) Заделяме динамично памет, като имаме предвид да има място и за затапващия символ.
 - 3) Присвояваме стойността на динамично заделената променлива на стринга.
 - 4) Освобождаваме заделената памет.

➤ В клас **Клетка** са имплементирани:

- конструктор по подразбиране и конструктор, приемащ два аргумента – размер и климат, чрез който се създават в началото на програмата клетки без животни в тях.
- селектори, за да може да достъпваме елементи на класа извън него.
- мутатори, за да може да даваме стойности на елементи на класа извън него.
- функция за строене на клетка за съществуващо животно, която създава клетка според вида на животното, което ще е вътре като потребителят определя размера и климата, като се очаква потребителят да въведе климат, който отговаря на вида на животното, т.е. ако животното е или Бронтозавър, или Тиранозавър, то климатът трябва да е сухоземен, ако животното е Плезиозавър, то климатът трябва да е воден, ако животното е Птицезавър, то климатът трябва да е въздушен.
- функция за строене на клетка по даден размер и климат, която създава клетка, в която няма животни и персонал.
- функция за премахване на животно от клетка, в която премахването на животно е съпроводено с премахване на персонал като се има предвид, че за 1 животно се грижат 2 човека, за 2 животни – 4 човека, за 3 животни – 5 човека, за 4 животни – 6 човека и така до 10 животни – 12 човека.
- оператор за изход – извежда информация за член данните на клетката.
- функцията за записване на информация за клетка в бинарен файл работи по следния начин:
 - Когато записваме член данни, които са стрингове, то:
 - 1) В новосъздадена променлива запазваме размера на стринга.
 - 2) Записваме във файла размера.
 - 3) Записваме информацията от самия стринг.
 - Когато записваме броя на служителите, който е от тип „int”, то става директно чрез вградената функция „write”.
 - Когато записваме член-данната „animals”, която е вектор от животни, първо записваме броя животни, а след това чрез цикъл записваме и данните за самите животни чрез функцията „write” от „class Dinosaur” като си осигуряваме, че ако няма животни, то няма да се запише никаква информация за тях.
- функция за четене на информация за динозавър от бинарен файл, която работи по следния начин:
 - Когато четем член данните, които са стрингове, то:
 - 1) Прочитаме дължината на стринга.
 - 2) Заделяме динамично памет, като имаме предвид да има място и за затапващия символ.
 - 3) Присвояваме стойността на динамично заделената променлива на стринга.
 - 4) Освобождаваме заделената памет.
 - Когато четем броя на служителите, който е от тип “int”, то става директно чрез вградената функция „read”.
 - Когато четем член-данната „animals”, която е вектор от животни, първо прочитаме броя животни, а след това чрез цикъл четем и данните за самите животни чрез функцията „read” от „class Dinosaur” като добавяме прочетеното във вектора „animals”.

➤ В клас **Склад** са имплементирани:

- конструктор по подразбиране като първоначално в склада има 300 кг трева, 300 кг месо и 300 кг риба. Затова при имплементиране на конструктора по подразбиране, на трите член данни, които са от тип „int”, се задават стойности 300.
- гетъри, за да може да достъпваме елементи на класа извън него.
- сетъри, за да може да даваме стойности на елементи на класа извън него.
- функции за зареждане на склада с трева, месо, риба или с трите храни едновременно като количеството храна се посочва от потребителя.
- оператор за изход – извежда информация за наличното количество храна.
- функцията за записване на информация за склада в бинарен файл става директно чрез вградената функция „write”, понеже количествата са от тип „int”.
- функцията за четене на информация за склада от бинарен файл става директно чрез вградената функция „read”, понеже количествата са от тип „int”.

➤ В клас **Джурасик парк** са имплементирани:

В класа Джурасик парк освен член данни, в „private” частта има и пет функции. Те са в „private” частта, защото се използват само в този клас. Това са функцията за записване в бинарен файл, чрез която се записва информация за клетките и склада чрез дефинираната в другите класове функция „write” и функцията за четене от бинарен файл информация за наличните клетки и запаси в склада чрез дефинираната в другите класове функция „read”. Третата функция проверява дали съществува файл с информация за парка и ако съществува дали в парка има някакви клетки. При несъществуване на файл или при неналичие на клетки, се записват във файла три произволно генерирани клетки, за да може да се удовлетвори условието на заданието, т.е. първоначално в зоопарка да има клетки, но да няма животни вътре. Петата функция е за изпечатване на информация.

В „public” частта има 2 конструкция и една функция. Във функцията са използвани много помощни променливи, с което си подсигуриявам правилна работа на зоопарка. В нея има команди от 1 до 7, които са в два цикъла и функцията се изпълнява, докато потребителя не въведе команда 7, която означава да се финализира Джурасик парка и да се запазят промените във файла, за да може при следващо изпълнение на програмата, информацията да не се е загубила.

```
std::cout << "\nWhat do you want to do?\n"
<< "1)Accept new animal.\n"
<< "2)Build new cage.\n"
<< "3)Remove available animal.\n"
<< "4)Feed dinosaurs.\n"
<< "5)Load the warehouse with food.\n"
<< "6)See information about current Jurassic park.\n"
<< "7)Finalize your Jurassic Park!\n";
```

3.2 Примери:

- Пример 1:
 - Стартирам програмата без да има съществуващ бинарен файл. Той се създава и в него на случаен принцип се генерират коректни клетки без животни.

```
Cage:1-->Size: small; Climate: air; Employee:0
=====
Cage:2-->Size: large; Climate: dry; Employee:0
=====
Cage:3-->Size: small; Climate: dry; Employee:0
=====
```

- Пример 2:
 - Добавяне на животно като потребителят въвежда името, пола, ерата и разряда. Според това се създава динозавър и се показва какви свободни клетки има и потребителят трябва да избере къде да постави животното.

```
->1
Name:Dino
Gender(female/male):male
Era(Triassic/Cretaceous/Jura):Cretaceous
Order(herbivorous/carnivorous/flying/aqueous):herbivorous
species:Brontosaurus
food:grass
Cage 2 is empty!
Cage 3 is empty!
You can choose from 2 cages!
The available cages are:
Cage 2: Size: large; Climate: dry; Employee:0
=====
Cage 3: Size: small; Climate: dry; Employee:0
=====
Please choose a cage number where you want to put the dinosaur:
-> 3
```

- Пример 3:
 - Създаване на клетка според размер и климат.

```
->2
Size(small/medium/large):medium
Climate(dry/air/water):water
You build new cage medium size and water climate with 0 employee inside!
```

- Пример 4:
 - Добавяне на животно. Според това се създава динозавър и изписва, че няма свободна подходяща клетка като потребителят създава такава, която да е подходяща според климата и разряда.

```
->1
Name:Dico
Gender(female/male):male
Era(Triassic/Cretaceous/Jura):Triassic
Order(herbivorous/carnivorous/flying/aqueous):aqueous
species:Plesiosaur
food:fish
You should build new cage!
Size(small/medium/large):small
Climate(dry/air/water):water
You build new cage small size and water climate with 0 employee inside!
```

- Пример 5:
- Премахване на животно от парка.

```
->3
The available animals are 2
1)Dino male Cretaceous herbivorous Brontosaurus grass
2)Coco female Triassic aqueous Plesiosaur fish
Please, select which animal to be removed:
->1
An animal was removed successfully!
```

- Пример 6:
- Хранене на динозавър.

```
->4
You fed dinosaur: Coco female Triassic aqueous Plesiosaur fish with 15kg fish
```

- Пример 7:
- Зареждане на склада с храна.

```
->5
What quantity grass do you want to load?
->15
What quantity meat do you want to load?
->45
What quantity fish do you want to load?
->67
In the warehouse there are 315kg grass, 345kg meat, 352kg fish.
```

4. Заключение

Смятам, че успях да осъществя първоначалните си цели и идеи. В бъдеще, когато придобия повече познания, този проект може да се доусъвършенства, но с знанията ми до тук, съм способна само на това. ☺