

Джурасик парк

Изготвено от Георги Газепов

Проектът представлява модел на зоопарк за динозаври. Паркът има клетки, в които живеят динозаврите, служители, които се грижат за динозаврите и склад с храна за животните. Реализацията позволява добавяне и премахване на динозаври, създаване на нова клетка и пълнене на склада с достатъчно храна за всички динозаври. Всяка клетка в парка е с някакъв капацитет в зависимост от това какъв е размерът ѝ. Всеки служител отговаря за една клетка. Динозаврите се хранят след всяко действие и се известява, ако няма достатъчно храна, за да бъдат нахранени. В една клетка живеят единствено динозаври от една ера и един разред. Цялата нужна информация за парка се пази в текстов файл.

1. Архитектура

1.1. Enums

Файлът Enums съдържа всички необходими за проекта Enum класове, а именно:

- **Sex** – Enum клас, който се използва за определяне на полът на динозаврите.
- **Era** – Enum клас, който се използва за определяне на ерата на динозаврите.
- **Category** – Enum клас, който се използва за определяне на разреда на динозаврите.
- **Food** – Enum клас, който се използва за определяне на типът храна, с които се хранят динозаврите.
- **Size** – Enum клас, който се използва за определяне на размера на клетките.

- **Climate** – Enum клас, който се използва за определяне на климатът на клетките.

1.2. FoodStorage

Структура, която представлява слад за храна. Използва се в класът DinosaurPark. Основните му характеристики са:

➤ **Public:**

- **meat** - променлива от тип `int`, в която се пази количеството месо.
- **fish** – променлива от тип `int`, в която се пази количеството риба.
- **grass** - променлива от тип `int`, в която се пази количеството трева.
- **FoodStorage()** – конструктор по подразбиране на класа. Задава стойности на **meat**, **fish** и **grass** съответно 0, 0 и 0.
- **FoodStorage(int meat, int fish, int grass)** – Конструктор с параметри. Задава стойности на **meat**, **fish** и **grass** спрямо подадените параметри.
- **FoodStorage& operator=(const FoodStorage& f)** – Предефиниция на `operator=`.
- **FoodStorage& operator-(const FoodStorage& f1, const FoodStorage& f2)** – Предефиниция на `operator-`.
- **operator<(const FoodStorage& f1, const FoodStorage& f2)** – Предефиниция на `operator<`.
- **std::ostream& operator<<(std::ostream& os, const FoodStorage& foodStorage)** - Предефиниция на `operator<<`

за изходен поток от данни. Подава всички данни във форматирания вид на потокът `os`.

- **`std::istream& operator>>(std::istream& is, FoodStorage& foodStorage)`** – Предефиниция на `operator>>` за входен поток от данни. Задава стойности на данните на `dinosaur` спрямо информацията в потока `is`.

1.3. Dinosaur

Класът `Dinosaur` съдържа всички данни за един динозавър, сетъри, гетъри и голямата четворка за класа, както и предефиниции на някои оператори. Основните му характеристики са:

➤ **Private:**

- **`name`** – променлива от тип `char*` в която се пази името на динозавъра
- **`sex`** – променлива от тип `Sex`, в която се пази полът на динозавъра.
- **`era`** – променлива от тип `Era`, в която се пази ерата на динозавъра.
- **`species`** – променлива от тип `char*`, в която се пази видът на динозавъра.
- **`category`** – променлива от тип `Category`, в която се пази разредът на динозавъра.
- **`food`** – променлива от тип `Food`, в която се пази типът храна, с който динозавърът се храни.
- **`void SetName(const char* name)`** – сетър за променливата `name`.
- **`void SetSex(const Sex sex)`** – сетър за променливата `sex`.

- ***void SetEra(const Era era)*** – сетър за променливата era.
- ***void SetSpecies(const char* species)*** – сетър за променливата species
- ***void SetCategory(const Category category)*** – сетър за променливата category.
- ***void SetFood()*** – сетър за променливата food. Стойността, която задава се определя пряко от стойността на променливата category.
- ***void CopyFrom(const Dinosaur& other)*** - копира всички данни на даден обект other. Използва се в копиконструктора и операторът за присвояване на класа.
- ***void Free()*** – освобождава заетата динамична памет от обекта, а именно паметта заета от променливите name и species. Използва се в оператора за присвояване и в дескструктора на класа.

➤ **Public:**

- ***Dinosaur()*** - конструктор по подразбиране на класа. Задава стойности на *name*, *sex*, *era*, *species*, *category* и *food* със стойности съответно nullptr, InvalidSex(-1), InvalidEra(-1), nullptr, InvalidCategory(-1) и InvalidFood(-1)
- ***Dinosaur(const char* name, const Sex sex, const Era era, const char* species, const Category category)*** – конструктор с параметри. Задава стойности на *name*, *sex*, *era*, *species*, *category* и *food* спрямо подадените параметри.

- ***Dinosaur(const Dinosaur& other)*** – Копиконструктора на класа. Създава обект с всички характеристики на ***other***.
- ***Dinosaur& operator=(const Dinosaur& other)*** – операторът за присвояване на класа. Присвоява всички данни на обекта ***other*** към вече съществуващ обект.
- ***~Dinosaur()*** – Деструктор на класа . Извиква ***Free()***, с помощта на която освобождава заетата от обекта памет.
- ***std::ostream& operator<<(std::ostream& os, const Dinosaur& dinosaur)*** – Предефиниция на ***operator<<*** за изходен поток от данни. Подава всички данни във форматиран вид на потокът ***os***.
- ***std::istream& operator>>(std::istream& is, Dinosaur& dinosaur)*** – Предефиниция на ***operator>>*** за входен поток от данни. Задава стойности на данните на ***dinosaur*** спрямо информацията в потока ***is***.
- ***bool operator==(const Dinosaur& d1, const Dinosaur& d2)*** – Предефиниция на ***operator==*** на класа. Проверява дали стойностите на всички данни на два обекта съвпадат.
- ***const char* GetName() const*** – Константен гетър за името на динозавъра. Връща стойността на променливата ***name***.
- ***const Sex GetSex() const*** – Константен гетър за полът на динозавъра. Връща стойността на променливата ***sex***.
- ***const Era GetEra() const*** – Константен гетър за ерата на динозавъра. Връща стойността на променливата ***era***.

- ***const char* GetSpecies() const*** - Константен гетър за видът на динозавъра. Връща стойността на променливата ***species***.
- ***const Category GetCategory() const*** – Константен гетър за разредът на динозавъра. Връща стойността на променливата ***category***.
- ***const Food GetFood() const*** – Константен гетър за храната с която динозавърът се храни. Връща стойността на променливата ***food***.

1.4. Cage

Класът Cage съдържа всички данни на една клетка в парка, сетъри, гетъри ама

➤ Private:

- ***dinosaurs*** - поинтер към от тип ***Dinosaur***. Представява масив, който съдържа динозаврите, които са в клетката.
- ***size*** – променлива от тип ***Size***, в която се пази размерът на клетката.
- ***climate*** – променлива от тип ***Climate***, в която се пази климатът на клетката.
- ***eraOfDinosaurs*** – променлива от тип ***Era***, в която се пази каква е ерата на динозаврите, които живеят в клетката.
- ***capacity*** – променлива от тип ***unsigned int***, в която се пази броят на динозаврите, които клетката може да побере.
- ***numbeOfDinosaurs*** – променлива от тип ***unsigned int***, в която се пази текущият брой на динозаврите в клетката.
- ***void SetSize(const Size size)*** – Сетър за променливата ***size***.

- ***void SetClimate(const Climate climate)*** – Сетър за променливата ***climate***.
- ***void SetEraOfDinosaurs(const Era eraOfDinosaurs)*** – Сетър за променливата ***eraOfDinosaurs***.
- ***void SetCapacity()*** – Сетър за променливата ***capacity***. Работата и се определя пряко от стойността на ***size***
- ***bool IsDinosaurEligible(const Dinosaur& dinosaur)*** – Която проверява дали подаденият като параметър динозавър е с подходящите характеристики, за да може да живее в клетката. Използва се в функцията ***AddDinosaur***.
- ***void CopyFrom(const Cage& other)*** – Копира всички данни на даден обект ***other***. Използва се в копиконструктора и операторът за присвояване на класа.
- ***void Free()*** – освобождава заетата динамична памет от обекта, а именно паметта заета от променливата ***dinosaur***. Използва се в оператора за присвояване и в дескструктора на класа.

➤ **Public:**

- ***Cage()*** – Конструктора по подразбиране на класа. Задава стойности на ***dinosaurs***, ***size***, ***climate***, ***eraOfDinosaurs***, ***capacity*** и ***numberOfDinosaurs*** със стойности съответно ***nullptr***, ***InvalidSize(-1)***, ***InvalidClimate(-1)***, ***InvalidEra(-1)***, ***0***, ***0***.
- ***Cage(const Size size, const Climate climate)*** – Конструктора на класа. Задава стойности на ***dinosaurs***, ***size***, ***climate***, ***eraOfDinosaurs***, ***capacity*** и ***numberOfDinosaurs*** спрямо подадените параметри.

- ***Cage(const Cage& other)*** – Копиконструктор на класа. Създава нов обект с характеристиките на other.
- ***Cage& operator=(const Cage& other)*** – оператора за присвояване на класа. Копира всички характеристики на other към вече съществуващ обект.
- ***~Cage()*** – Дескрукторът на класа. Освобождава заетата памет.
- ***std::ostream& operator<<(std::ostream& os, const Cage& cage)*** – Предефиниция на operator<< за изходен поток от данни. Подава всички данни във форматиран вид на поток за изходни данни.
- ***std::istream& operator>>(std::istream& is, Cage& cage)*** - Предефиниция на operator>> за входен поток от данни. Задава стойности на данните на cage спрямо информацията в потока is.
- ***bool IsEmpty()const*** – Проверява дали клетката е празна.
- ***bool IsFull()const*** – Проверява дали клетката е пълна.
- ***short AddDinosaur(const char* name, const Sex sex, const Era era, const char* species, const Category category)*** – Създава обект от тип ***Dinosaur*** с подадените като аргументи характеристики и вика функцията ***AddDinosaur(const Dinosaur& dinosaur)***, като връща резултатът от нейното изпълнение.
- ***short AddDinosaur(const Dinosaur& dinosaur)*** – Добавя динозавърът, който е подаден като параметър към клетката. Връща -1, ако вече в клетката има динозавър със

същите характеристики, 0, ако няма клетка, която да отговаря на нужният климат и/или разред на динозавъра и 1, ако динозавърът е добавен успешно. При успешно добавяне увеличава *numberOfDinosaurs* с 1.

- ***bool RemoveDinosaur(const char* name, const Sex sex, const Era era, const char* species, const Category category)*** – Създава обект от тип *Dinosaur* с подадените като аргументи характеристики и вика функцията ***bool RemoveDinosaur(const Dinosaur& dinosaur)***, като връща резултатът от нейното изпълнение.
- ***bool RemoveDinosaur(const Dinosaur& dinosaur)*** – Премахва динозавър от клетката, ако подаденият като аргумент такъв съществува. Връща false, ако клетката е празна или не съществува динозавър, който напълно да отговаря на попаденият като аргумент. Ако има такъв динозавър, намалява стойността на *numberOfDinosaurs* с 1, шифтва в масива всеки динозавър след премахнатият с една позиция наляво и връща true.
- ***const Dinosaur* GetDinosaurs()const*** – Константен гетър за динозаврите в клетката. Връща адреса на указателя *dinosaurs*.
- ***const Size GetSize()const*** – Константен гетър за размерът на клетката. Връща стойността на променливата *size*.
- ***const Climate GetClimate()const*** – Константен гетър за климатър на клетката. Връща стойността на променливата *climate*.
- ***const Era GetEraOfDinosaurs()const*** – Константен гетър за ерата на динозаврите в клетката. Връща стойността на променливата *eraOfDinosaurs*.

- ***const unsigned GetCapacity()const*** – Константен гетър за капацитета на клетката. Връща стойността на променливата ***capacity***.
- ***const unsigned GetNumberOfDinosaurs()const*** – Константен гетър за текущият брой на динозаврите в клетката. Връща стойността на променливата ***numberOfDinosaurs***.

1.5. DinosaurPark

Класът представлява реализация на паркът за динозаври. В него има клетки, в които има динозаври. Също така има склад с храна, с която се хранят динозаврите и служители, които се грижат за динозаврите в клетките.

➤ Private:

- ***cages*** – Поинтер към обект от тип ***Cage***. Представлява масив, който съдържа клетките в парка.
- ***numberOfCages*** – Променлива от тип unsigned int, в която се пази броят на клетките в парка.
- ***maxNuberOfCages*** – Променлива от тип unsigned int, в която се пази максималният брой на клетките, които може да има в парка, без да е необходимо той да се оразмерява.
- ***staffCount*** – Променлива от тип unsigned int, в която се пази броят на служителите в парка.
- ***foodAvailable*** – Променлива от тип FoodStorage, в която се пазят наличните месо, трева и риба в склада.
- ***foodRequired*** – Пронелива от тип FoodStorage, в която се пазят необходимите месо, трева и риба за храненето на динозаврите.

- ***void Resize()*** – Преоразмерява масивът *cages*, като го прави два пъти по-голям и променя стойността на *maxNuberOfCages* като я умножава по две.
- ***void CopyFrom(const DinosaurPark& other)*** - Копира всички данни на даден обект *other*. Използва се в копиконструктора и операторът за присвояване на класа.
- ***void Free()*** – освобождава заетата динамична памет от обекта, а именно паметта заета от променливата ***dinosaur***. Използва се в оператора за присвояване и в дескструктора на класа.

➤ **Public:**

- ***DinosaurPark()*** – Конструкторът по подразбиране на класа. Задава стойности на ***numberOfCages***, ***maxNumberOfCages*** и ***staffCount*** съответно 3, 4 и ***numberOfCages***. Заделя памет за и създава 3 клетки, по една от всеки вид климат. Задава стойности на ***meat***, ***fish*** и ***grass*** на променливите ***foodAvailable*** и ***foodRequired*** съответно 10, 10, 10 и 0, 0, 0.
- ***DinosaurPark(const DinosaurPark& other)*** - Копиконструктор на класа. Създава нов обект с характеристиките на *other*.
- ***DinosaurPark& operator=(const DinosaurPark& other)*** - оператора за присвояване на класа. Копира всички характеристики на *other* към вече съществуващ обект.
- ***~DinosaurPark()*** – Дескрукторът на класа. Освобождава заетата памет.
- ***std::ostream& operator<<(std::ostream& os, const DinosaurPark& park)*** - Предефиниция на *operator<<* за

изходен поток от данни. Подава всички данни във форматирания вид на поток за изходни данни.

- ***std::istream& operator>>(std::istream& is, DinosaurPark& park)*** – Предефиниция на ***operator>>*** за входен поток от данни. Задава стойности на данните на ***cage*** спрямо информацията в потока ***is***.
- ***void CreateCage(const Size size, const Climate climate)*** – Създава обект от тип ***Cage*** с подадените аргументи за характеристики и извиква функцията ***CreateCage(const Cage& cage)***.
- ***void CreateCage(const Cage& cage)*** – Създава нова клетка в парка с характеристиките на ***cage***, като извиква функцията ***Resize()***, ако ***numberOfCages*** е равно на ***maxNumberOfCages***. Увеличава стойностите на ***numberOfCages*** и ***stuffCount*** с единица.
- ***short AddDinosaur(const char* name, const Sex sex, const Era era, const char* species, const Category category)*** - Създава обект от тип ***Dinosaur*** с подадените като аргументи характеристики и вика функцията ***AddDinosaur(const Dinosaur& dinosaur)***, като връща резултатът от нейното изпълнение.
- ***short AddDinosaur(const Dinosaur& dinosaur)*** – За всяка клетка извиква ***AddDinosaur***, проверява дали добавянето е било успешно или не и ако резултатът от изпълнението на извиканата функция е -1 или 1 го връща. Съответно връща -1, ако вече съществува динозавър, чиито характеристики напълно да отговарят на попаденият като аргумент, 0, ако няма клетка, която да отговаря на климатът и разредът на ***dinosaur*** и 1, ако добавянето е успешно. При успешно извикване задава ерата на клетката спрямо ерата на

dinosaur, ако вече не е зададена и увеличава *meat*, *fish* или *grass* на *foodRequired* с единица в зависимост от това с какво се храни новият динозавър.

- ***bool RemoveDinosaur(const char* name, const Sex sex, const Era era, const char* species, const Category category)*** – Създава обект от тип *Dinosaur* с подадените като аргументи характеристики и вика функцията ***RemoveDinosaur(const Dinosaur& dinosaur)***, като връща резултата от нейното изпълнение.
- ***bool RemoveDinosaur(const Dinosaur& dinosaur)*** – За всяка клетка извиква функцията ***RemoveDinosaur(const Dinosaur& dinosaur)*** и ако в някоя от клетките съществува динозавър, чийто характеристики да отговарят напълно на попаденият като аргумент динозавър връща *true*. В противен случай връща *false*.
- ***void RefillStorage()*** – Пълни склада с храна с 4 пъти нужната за храненето на динозаврите.
- ***void FeedTheDinosaurs()*** – Храни динозаврите и вади нужното за храненето им от наличното количество храна в склада.
- ***const Cage* GetCages()const*** – Константен гетър за клетките. Връща адреса на указателя *cages*.
- ***const unsigned GetNumberOfCages()const*** – Константен гетър за текущият брой на клетките в парка. Връща стойността на *numberOfCages*.
- ***const unsigned GetMaxNumberOfCages()const*** – Константен гетър за броя на клетките, който паркът побира. Връща стойността на *maxNumberOfCages*.

- ***const unsigned GetStaffCount()const*** – Константен гетър за броят на служителите в парка. Връща стойността на ***staffCount***.
- ***const FoodStorage GetFoodRequired()const*** – Константен гетър за необходимата храна за храненето на всички динозаври. Връща стойността на ***foodRequired***.
- ***const FoodStorage GetFoodAvailable()const*** – Константен гетър за наличната храна в склада. Връща ***foodAvailable***.

1.6. JurassicParkUI

Класът представлява конзолният интерфейс на проекта. В него се съдържа цялата необходима логика при работата на потребителя. Има private функции, за по-лесното му реализиране. Зарежда информацията от файл при начало започване на работа с конзолата и записва промените в същия файл при приключване. Основните му характеристики са:

➤ Private:

- park – обект от тип ***DinosaurPark***.
- ***const char* EnterName()const*** – Константна функция, която се грижи за въвеждането на името на динозавъра при командите Add dinosaur и Remove dinosaur.
- ***const Sex EnterSex()const*** – Константна функция, която се грижи за въвеждането на полът на динозавъра при командите Add dinosaur и Remove dinosaur. Използва се във функциите ***AddDinosaur()*** и ***RemoveDinosaur()***.
- ***const Era EnterEra()const*** – Константна функция, която се грижи за въвеждането на ерата на динозавъра при

командите Add dinosaur и Remove dinosaur. Използва се във функциите **AddDinosaur()** и **RemoveDinosaur()**.

- **const char* EnterSpecies()const** – Константна функция, която се грижи за въвеждането на видът на динозавъра при командите Add dinosaur и Remove dinosaur. Използва се във функциите **AddDinosaur()** и **RemoveDinosaur()**.
- **const Category EnterCategory()const** – Константна функция, която се грижи за въвеждането на разредът на динозавъра при командите Add dinosaur и Remove dinosaur. Използва се във функциите **AddDinosaur()** и **RemoveDinosaur()**.
- **const Size EnterSize()const** – Константна функция, която се грижи за въвеждането на размерът на клетката при командата Create cage. Използва се във функцията **CreateCage()**.
- **const Climate EnterClimate()const** – Константна функция, която се грижи за въвеждането на климатът на клетката при командата Create cage. Използва се във функцията **CreateCage()**.
- **const ClimateCategoryToClimate(const Category category) const** – При подаден разред на динозавър връща какъв е подходящият климат за него.
- **const bool AreDinosaursHungry()const** – Връща дали динозаврите са гладни, т.е. в склада няма достатъчно храна за тях.
- **void AddDinosaur()** – Съдържа логиката, необходима за добавяне на динозавър при командата Add dinosaur.

- ***void RemoveDinosaur()*** – Съдържа логиката, необходима за премахване на динозавър при командата Remove dinosaur.
- ***void CreateCage()*** - Съдържа логиката, необходима за създаване на нова клетка при командата Create cage.
- ***void RefillTheStorage()*** – Пълни склада с храна.
- ***void LoadFromFile(const char* filename)*** – Зарежда данните от текстов файл с име filename към park. Хвърля изключение, ако файлът е празен или не съществува.
- ***void SaveToFile(const char* filename)const*** – Записва данните на park към текстов файл с име filename и създава такъв, ако не съществува.

➤ **Public:**

- ***void Run()*** – Функция, в която се съдържа цялата логиката за работа с конзолният интерфейс, както и самият конзолен интерфейс. За реализирането помагат всички функции от private частта.

2. Схема на проекта

