

Описание классов

Class Item

Item – данный класс используется для создания объектов класса Item (заявка).

Переменные класса Item:

- `name: String` – имя заявки;
- `description: String` – описание заявки;
- `create: long` – время создания заявки;
- `[]comm: Comment` – ссылочная переменная на объект класса Comment.

Методы класса Item:

- `setName(name: String): void` – задает имя заявки;
- `setDescription(description: String): void` – задает саму заявку;
- `setCreate(create: long): void` – задает время создания заявки;
- `getName(): String` – возвращает имя заявки;
- `getDescription(): String` – возвращает саму заявку;
- `getCreate(): long` – возвращает время создания заявки;
- `getComm(): Comment` – возвращает ссылочную переменную на массив объектов класса Comment (комментариев).

Class Bag

Bag – данный класс используется для создания объекта Bag (проблема, ремонт). Класс Bag наследует (extends) класс Item, поэтому имеет аналогичные поля (переменные) и свойства (методы) классу Item.

Class Task

Task – данный класс используется для создания объекта Task (заявка). Класс Task наследует (extends) класс Item, поэтому имеет аналогичные поля (переменные) и свойства (методы) классу Item.

***Примечание:** Названия классов Bag и Task будут использоваться в фильтрации: вывод объектов класса Bag или только – Task. Также, в дальнейшем, в классах наследниках (Bag, Task) будут реализованы свои переменные и методы.*

Class Comment

Comment – данный класс используется для создания объектов класса Comment (заявка). Комментарий (объект класса Comment) добавляются пользователем к любой заявке. К каждой заявке может быть добавлено не ограниченное количество комментариев.

Переменные класса Comment:

- `nameld: String` – имя пользователя который создает комментарий;
- `comment: String` – описание комментария;
- `createComment: long` – время создания комментария.

Методы класса Comment:

- `setNameld(nameld: String): void` – задает имя комментария;
- `setComment(comment: String): void` – задает сам комментарий;
- `setCreate(create: long): void` – задает время создания заявки;
- `getNameld(): String` – возвращает имя комментария;
- `getComment(): String` – возвращает сам комментарий;
- `getCreate(): long` – возвращает время создания комментария.

Class Tracker

Tracker – данный класс используется для работы с массивом объектов заявок. Работа производится через ссылочную переменную item.

Переменные класса Tracker:

- `[]item: Item` – объявление ссылочной переменной типа Item (ссылочная переменная на массив объектов заявок).

Методы класса Tracker:

- `addItem(): void` – производит добавление заявки (нового объекта) в общий массив заявок;

Примечание: В данном методе также производится проверка длины массива. Если массив заполнен, то производится вызов метода `riseLength` класса `AutoSizeArray`, который возвращает ссылку на новый массив заявок. Новый массив заявок имеет длину на единицу больше чем предыдущий и состоит из предыдущего массива плюс последняя ячейка «null». В последнюю ячейку нового массива заявок производится добавление новой заявки.

- `renameItem(position: int): void` – производит изменения значений переменных объекта `Item` (заявки) по указанной позиции заявки в массиве;
- `removeItem(position: int) void` – производит удаление объекта класса `Item` (заявки) из массива объектов по номеру массива объектов, а также массив комментариев созданных для данной заявки;

Примечание: Удаляемая ячейка массива инициализируется как «null». Далее «null» записывается в конец массива, а все элементы массива, которые были до «null», смещаются влево на один элемент. Далее ссылочная переменная передается в метод `cutLength` класса `AutoSizeArray`, где производится инициализация нового массива объектов класса `Item`, а копирование "старого" массива объектов в новый без «null». Метод `cutLength` возвращает ссылочную переменную на новый массив объектов класса `Item`.

- `showItem(): void` – выводит в консоль весь массив заявок (объектов), также выводит массив комментариев для заявки, если комментарии были добавлены;
- `showItemFilter(createStart: long, createEnd: long): void` – выводит список заявок в консоль с учетом фильтрации – заданный временной диапазон. Данный метод принимает на вход начало и конец временного диапазона поиска (`createStart: long, createEnd: long`);

Примечание: Для определения валидности и поиска указанного временного диапазона в методе `showItemFilter` используются методы `rangeItemValid`, `getItemFilter` класса `Filter`.

- `addComment(position: int): void` – производит добавление комментария в массив комментариев к заявке. Добавление производится по указанной позиции массива объектов (переменная `position: int`). Данный метод работает с ссылочной переменной `item`, а также использует метод `getComm` класса `Item` для доступа к полям и методам объекта класса `Comment`.

Примечание: При добавлении первого комментария производится инициализация массива объектов комментариев в длиной "1". при добавлении второго комментария происходит увеличение массива комментариев на единицу при использовании метода `riseLength` класса `AutoSizeArray`.

Class Filter

Filter – данный класс используется для определения валидности временного диапазона и самого диапазона по времени в массиве объектов класса `Item`.

Переменные класса `Filter`:

- `[]itemFilter: int` – ссылочная переменная на массив целочисленных значений (содержит два значения: позицию начала и конца диапазона объектов в массиве класса `Item`);
- `rangeItem: boolean` – переменная фиксирующая валидность заданного временного диапазона.

Примечание: Если `rangeItem = true`, то `[]itemFilter` инициализируется и содержит два значения: позицию начала и конца диапазона объектов в массиве класса `Item`.

Методы класса `Filter`:

- `rangeItemValid(createStart: long, createEnd: long, []item: Item): boolean` – производит проверку на валидность указанного временного диапазона по входящим переменным `createStart` и `createEnd` в массиве объектов `Item` (заявок). Также в методе инициализируется массив `[]itemFilter`;
- `getItemFilter(): int` – возвращает ссылочную переменную на массив `[]itemFilter`.

Class Search

Search – данный класс используется для поиска введенной подстроки в строке. В качестве строки используются переменные объектов массива класса `Item` – `name`, `desription`.

Переменные класса `Search`:

- `[]itemSearchr: Item` – ссылочная переменная на массив объектов класса `Item`, где были

- совпадения подстроки в строке;
- `[]Item Item` – ссылочная переменная на массив объектов заявок.

Методы класса Search:

- `keyItemValid(keyItem: String): boolean` – производит проверку на валидность введенного критерия поиска – переменная name или description класса Item;
- `textItemValid(testItem: String): boolean` – производит поиск подстроки в строке, а также формирует массив объектов (заявок) `[]ItemSearch` по результатам поиска и возвращает «true» если результат поиска положителен ил «false» - отрицателен;
- `getItemSearchr(): int` – возвращает ссылочную переменную на массив `[]ItemFiltr`.

Class AutoSizeArray

`AutoSizeArray` – данный класс используется для увеличения или уменьшения массива объектов (заявок) на «1» в процессе работы с `[]Item Item`.

Методы класса AutoSizeArray:

- `riseLength(item: Item): Item[]` – производит формирование нового массива объектов (заявок), который включает все объекты класса Item, а также один дополнительный элемент «null» для добавления нового объекта (заявки) и возвращает ссылочную переменную на "новый" массив объектов Item – производится переинициализация ссылочной переменной `[]Item` класса Item;
- `cutLength(item: Item): Item[]` – производит формирование нового массива объектов (заявок), который включает все объекты класса Item и не содержит последний элемент «null», метод также возвращает ссылочную переменную на "новый" массив объектов Item – производится переинициализация ссылочной переменной `[]Item` класса Item.