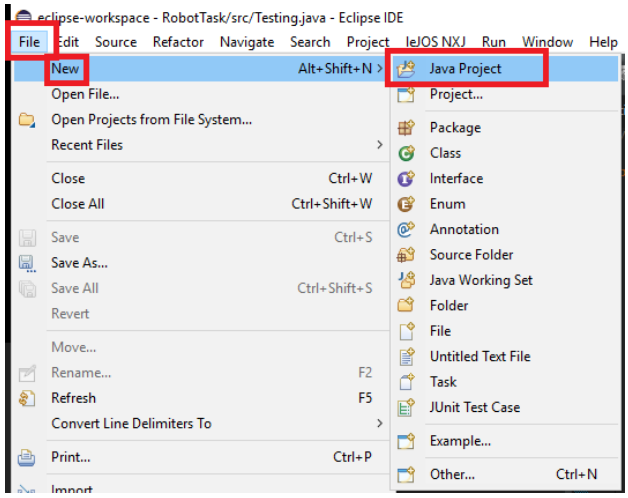
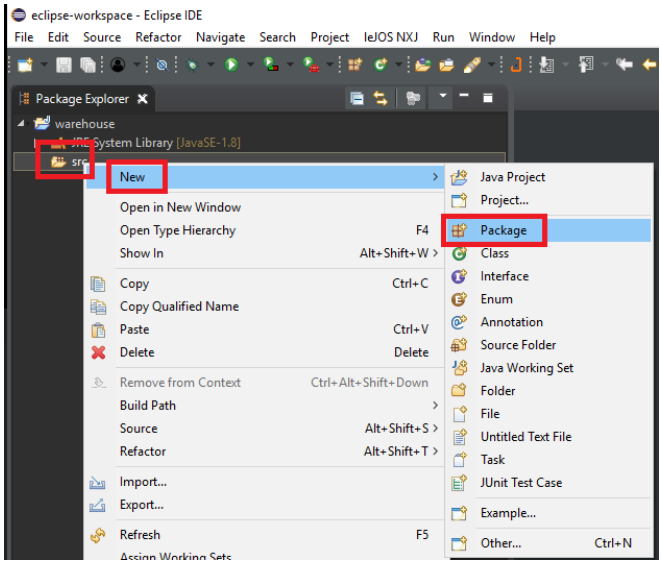
Summer School 2018 - Exercise 5
Automated high-bay warehouse

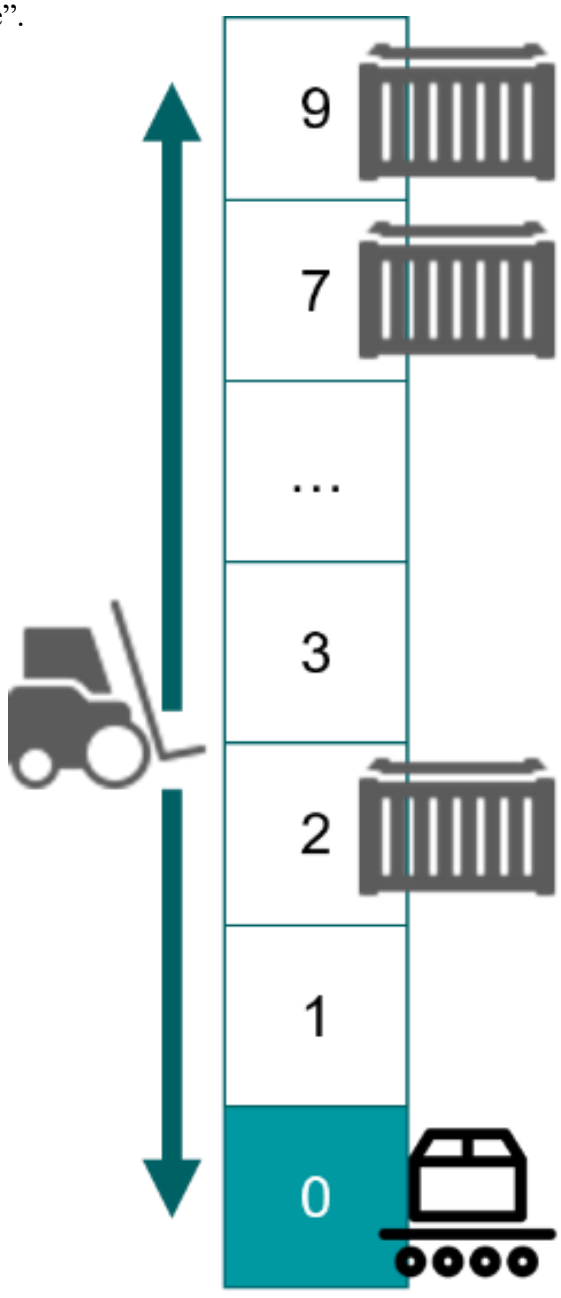
Task 0

- Create a new project in Eclipse and name it “warehouse”.



- Create a package “opms” and create all following classes inside this package.





IMA - RWTH Aachen University
Institute of Information Management in Mechanical Engineering IMA

Acting Head of Institute: Univ.-Prof. Dr.-Ing. Christian Hopmann
1st Deputy: apl.-Prof. Dr. habil. Ingrid Isenhardt
2nd Deputy: Dr. rer. nat. Frank Hees
Apl.-Prof. Dr. rer. Nat. Sabina Jeschke
Senior Advisor: Univ.-Prof. Dr.-Ing. em. Klaus Henning

Task 1: Pallet

- Use the package name "opms" for all classes.
- Create a class `Pallet`. Add a field `private int serialNumber`. Add a constructor `public Pallet(int serialNumber)` which saves the given serial number to the field `serialNumber`.
- Add a method `public int getSerialNumber()` which returns the `serialNumber` field.
- Add a method `public String toString()`, which returns the `serialNumber` as `String`.
  Hint:You can use `Integer.toString(serialNumber)` for that.

Task 2: Conveyor Belt

- Create a class `ConveyorBelt`.
- Add a field `private List<Pallet> pallets`.
  Initialize it via `new ArrayList<>().`
- Add a method `void loadPallet(Pallet pallet)` that adds the given `Pallet` to the List `pallets`.
  Hint: You can use a list's method `add(Pallet pallet)` for that.
- Add a method `List<Pallet> getPallets()` which returns the list called pallets.

Task 3: Forklift

- Create a class `Forklift`.
- Add a private field `Pallet[] palletSlots` and a private field `ConveyorBelt conveyorBelt`.
- Add a constructor `public Forklift(Pattet[] palletSlots, ConveyorBelt conveyorBelt)`, which saves both arguments to the private fields.

Task 4: Automated Warehouse

- Create a class `AutomatedWarehouse`.
- Add a method `public static void main(String[] args).`
- In this method, create an `Pallet[]` array of size 9 and name it `palletSlots`.
  Tip: You might want to use the command `new Pallet[int n]` with your desired size n.
- Create five `Pallet` objects via `new Pallet(int serialNumber)` with the serial numbers 541201, 541202, 663319, 663325 and 909042. Name them pallet1, pallet2, and so on.
- Add these `Pallet` objects to the above created array at slots 1, 2, 5, 6 and 9. Do not assign anything to the other slots, so they will automatically contain `null`.
- Also, create a `ConveyorBelt` instance named `conveyorBelt` via `new ConveyorBelt().`
- Finally, create a `Forklift` instance named `forklift` by passing the above-created variables `palletSlots` and `conveyorBelt` to the `Forklift` constructor.

Task 5: Bringing the Forklift to Life (1)

- You will now enable the `Forklift` to move and shift `Pallets`.
- Create a new Java class named `ForkliftOutOfBoundsException` that extends `Exception.`
- Navigate to the `Forklift` class and add a field `private int position`.
- Add a method `public void moveTo(int position) throws ForkliftOutOf-BoundsException`, which sets the private field `position` to the given value.
- Before setting the position, check if the given position is in the `Forklift's` boundaries (0 to 9). If not, throw a `new ForkliftOutOfBoundsException()`.
- We want to know if there is a `Pallet` at the forklift's current position.
  Add a method `public boolean seesPallet()` that returns `true`, if there is a `Pallet` in the `palletSlots` at the forklift's current `position`, and `false` otherwise.

Task 6: Bringing the Forklift to Life (2)

- We want to allow the forklift to lift pallets.
  Add a field Pallet `Pallet currentPallet` to the forklift.
- Add a method `public void liftPallet()` to the forklift, that
  - Assigns the `Pallet` from the `palletSlots[position]` to the private field `cur-rentPallet`
  - And removes the just picked up `Pallet` from the `palletSlots`.
    Hint: Assign `null` to the `palletSlots[position]` to remove the pallet.
- We want to allow the forklift to place pallets on the conveyor belt.
- Add a method public void `placeOnBelt()` that
  - Moves the forklift to the conveyor belt's position (0)
    Hint: Do not catch the exception here, but just pass it on by adding `throws ForkliftOutOfBoundsException` to this method's signature
  - Call `conveyor.loadPallet(this.currentPallet)` to load the current pallet on the belt.
  - Set the `currentPallet` to `null` to stop carrying it.

Final Task 7: Scheduling the Forklift

- Navigate to the main method in the `AutomatedWarehouse` class and implement the following schedule to load all pallets to the conveyor belt:
  1. Move the forklift to a pallet slot
  2. Check if there is a pallet at its current position using the `seesPallet()` method.
  3. If `true`: Call `liftPallet()` and then place it on the belt via `placeOnBelt().`
- Do these steps for all pallet slots.
  Hint: A loop `for(int slot = 1; i <= 9; i++)` might be useful.
- Hint: You need to catch any `ForkliftOutOfBoundsExceptions` here, so wrap your loop into a `try-catch`-block.
- As soon as all pallets have been loaded to the conveyor belt, your loop should stop and print a list of all loaded pallets to the console.
  Hint: Call `System.out.println(conveyorBelt.getPallets())`.
  Hint: The `conveyorBelt.getPallets()` automatically calls `pallet.toString()` for each pallet for you.