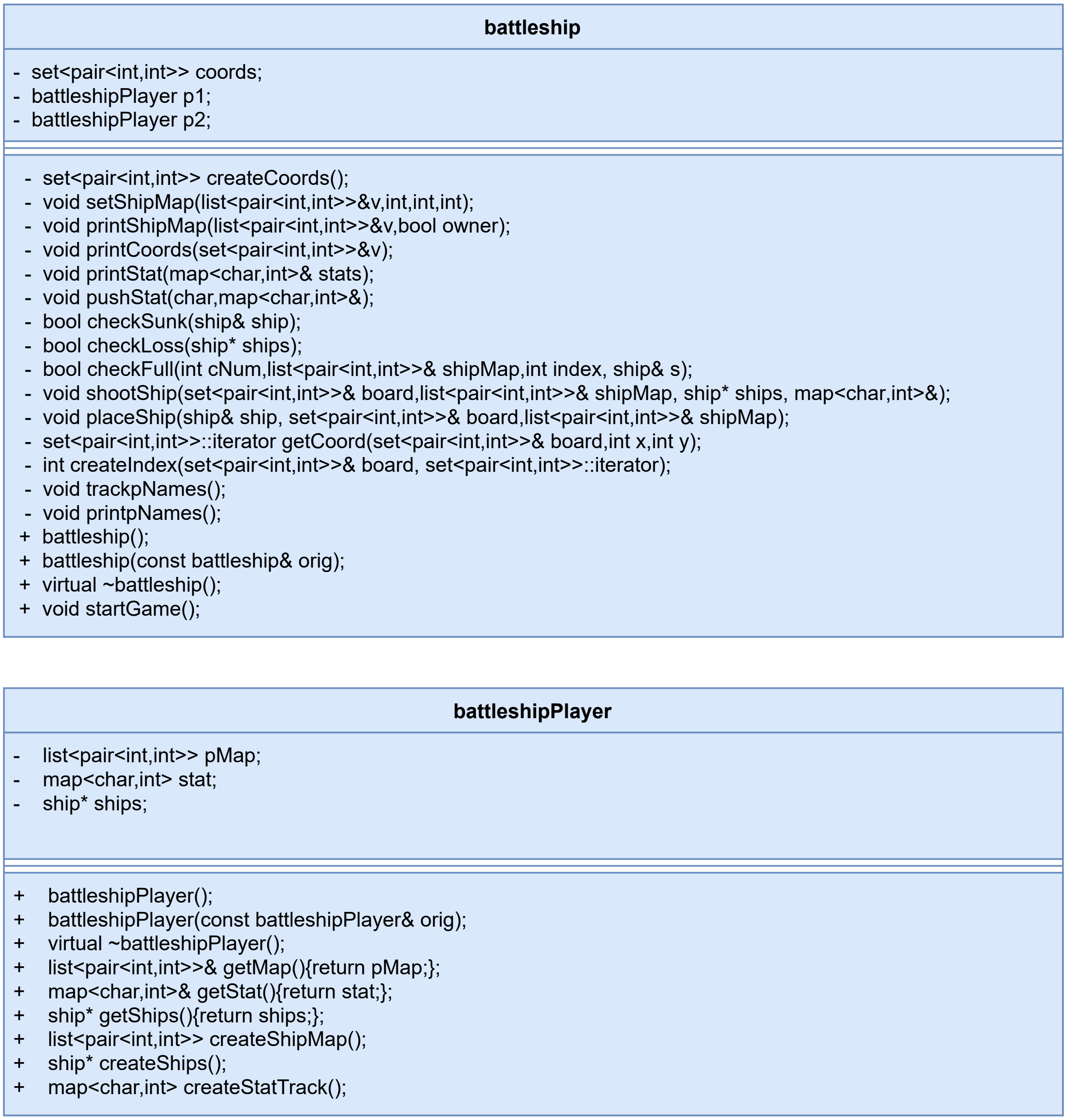

battleship

```
-  set<pair<int,int>> coords;
-  battleshipPlayer p1;
-  battleshipPlayer p2;
```

```
  -  set<pair<int,int>> createCoords();
  -  void setShipMap(list<pair<int,int>>&v,int,int,int);
  -  void printShipMap(list<pair<int,int>>&v,bool owner);
  -  void printCoords(set<pair<int,int>>&v);
  -  void printStat(map<char,int>& stats);
  -  void pushStat(char,map<char,int>&);
  -  bool checkSunk(ship& ship);
  -  bool checkLoss(ship* ships);
  -  bool checkFull(int cNum,list<pair<int,int>>& shipMap,int index, ship& s);
  -  void shootShip(set<pair<int,int>>& board,list<pair<int,int>>& shipMap, ship* ships, map<char,int>&);
  -  void placeShip(ship& ship, set<pair<int,int>>& board,list<pair<int,int>>& shipMap);
  -  set<pair<int,int>>::iterator getCoord(set<pair<int,int>>& board,int x,int y);
  -  int createIndex(set<pair<int,int>>& board, set<pair<int,int>>::iterator);
  -  void trackpNames();
  -  void printpNames();
  +  battleship();
  +  battleship(const battleship& orig);
  +  virtual ~battleship();
  +  void startGame();
```

battleshipPlayer

```
-    list<pair<int,int>> pMap;
-    map<char,int> stat;
-    ship* ships;
```

```
+    battleshipPlayer();
+    battleshipPlayer(const battleshipPlayer& orig);
+    virtual ~battleshipPlayer();
+    list<pair<int,int>>& getMap(){return pMap;};
+    map<char,int>& getStat(){return stat;};
+    ship* getShips(){return ships;};
+    list<pair<int,int>> createShipMap();
+    ship* createShips();
+    map<char,int> createStatTrack();
```